

**IVC Course Code : 319**

# **COMPUTER SCIENCE & ENGINEERING**

**First Year**

**Intermediate Vocational Course**

**Paper II : Programming In C<sup>TM</sup>**

**NAME:** \_\_\_\_\_

**ROLL NO.** \_\_\_\_\_

# **COMPUTER SCIENCE & ENGINEERING**

## **Paper - II**

### **PROGRAMMING IN 'C'**

#### **INDEX**

|                 |   |              |
|-----------------|---|--------------|
| <b>Unit-I</b>   | <b>Introduction to Problem Solving Techniques</b> | <b>3-12</b>  |
| <b>Unit-II</b>  | <b>Features of 'C'</b>                            | <b>13-45</b> |
| <b>Unit-III</b> | <b>Arrays in 'C'</b>                              | <b>46-57</b> |
| <b>Unit-IV</b>  | <b>Functions</b>                                  | <b>58-69</b> |
| <b>Unit-V</b>   | <b>Structures in 'C'</b>                          | <b>70-80</b> |
|                 | <b>QUESTION BANK</b>                              | <b>81-88</b> |

## UNIT-I

### Introduction to Problem Solving Techniques

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Procedure అనగా నేమి? What is procedure?

జ. **Procedure:** ఒక స్థిరమైన, దశల వారీ కార్యకలాపాల శ్రేణి అనేది ఒక విధిని సరిగ్గా నిర్వహించడానికి అదే క్రమంలో అనుసరించాల్సిన చర్య (ఖచ్చితమైన ప్రారంభ మరియు ముగింపు పాయింట్లతో).

#### 2. Algorithm అనగా నేమి? Define an algorithm.

జ. **Algorithm:** అల్గోరిథమ్ అనేది ఇచ్చిన సమస్యను పరిష్కరించడానికి ఒక క్రమపద్ధతిలో ఉపయోగించుకోనే విధానాన్ని " అల్గోరిథమ్ " అంటారు.

#### 3. ఇవ్వబడిన రెండు పూర్ణాంకాల సంఖ్యలలో పెద్దదాన్ని ముద్రించడానికి అల్గోరిథమ్ ను వ్రాయండి.

**Write an algorithm to print the biggest of given two integer numbers.**

జ. **Algorithm to print biggest of two integer numbers :**

పద్ధతి 1: ప్రారంభించండి

పద్ధతి 2: టైప్ పూర్ణాంకం యొక్క రెండు వేరియబుల్ a మరియు b లను ప్రకటించండి.

పద్ధతి 3: a మరియు b కోసం విలువలను చదవండి

పద్ధతి 4:  $a > b$  ఉందో లేదో తనిఖీ చేయండి

ఎ) అయితే (if)  $a > b$

బి) ప్రింట్ a పెద్దది

సి) ఎల్స్ (else)

డి) ప్రింట్ b పెద్దది.

పద్ధతి 5: ఆపు.

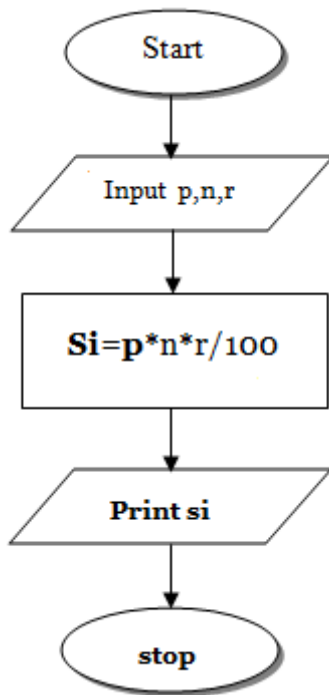
#### 4. Flow-chart ని నిర్వచింపుము Define flow-chart.

జ. **Flow-chart:** ఫ్లో-చార్ట్ అనేది ఇచ్చిన సమస్యను పరిష్కరించడానికి బొమ్మల రూపములో ప్రోగ్రామ్ ను ఉపయోగించుకోనే విధానాన్ని ఫ్లో-చార్ట్ అంటారు.

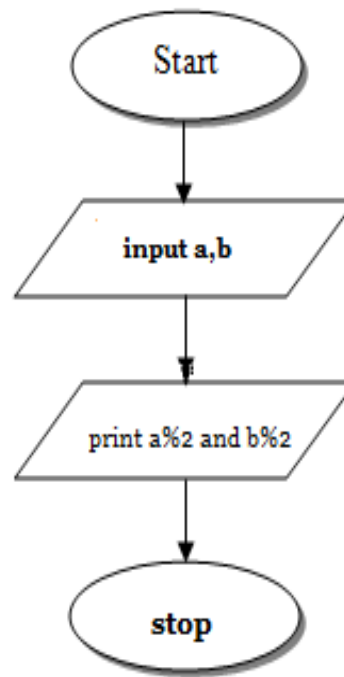
#### 5. Flow-chart చిహ్నాలు ఏమిటి? What are the flow-chart symbols ?

జ. **Symbols used in flow-chart:** బాణాలు, ఎలిప్స్, రాంబస్, దీర్ఘ చతురస్రాలు వజ్రాలు, సమాంతర రేఖలు, వృత్తాలు, సిలిండర్ , ఆఫ్-పేజీ కనెక్టర్లు మొదలైనవి.

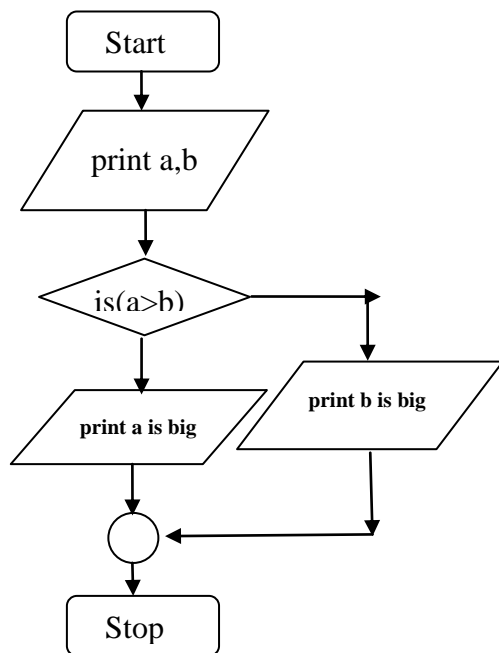
6. Draw a flow-chart to calculate the simple interest using  $(SI=P*N*R/100)$ .



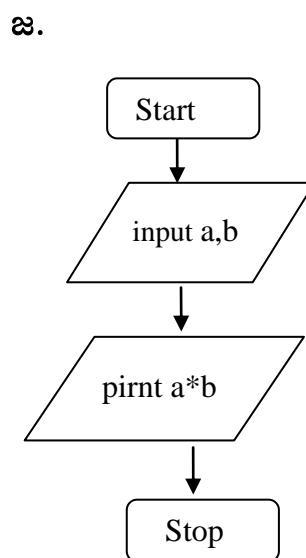
7. Draw a flow-chart to find the remainder values of given two numbers.



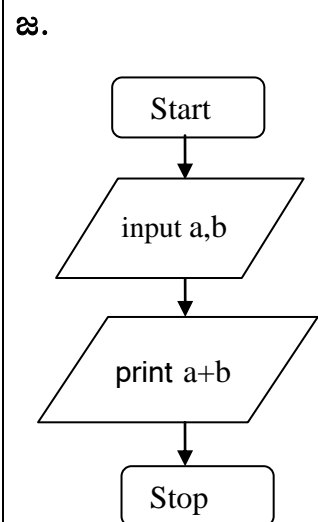
8. Draw a flow-chart to find the biggest value of given two integer numbers.



9. Draw a flow-chart to calculate product of given two numbers.



11. Draw flowchart to print sum of two given numbers.



## 10. Pseude code ఆనగా నేమి? What is pseudo code ?

**జ. Pseudo code :** ఇంగ్లీష్ వంటి భాషను ఉపయోగించి సమస్య సాధనకు కావలసిన ఆదేశాలను ఒక నిర్దిష్ట క్రమంలో రాస్తే దానిని సుడోకోడ్ అని అంటారు. ఇటువంటి బొమ్మలు లేదా సింబల్స్ ఉండవు సుడోకోడ్ రాసేటప్పుడు ఒక కంప్లీట్ లాంగ్వేజ్ యొక్క స్టేట్ మెంట్ ను పాటించవలసిన అవసరం లేదు. కొన్ని పదాలు మరియు ఫేసెస్ లను రాస్తారు.

**ఉపయోగాలు:**

- చాలా తేలికగా రాయవచ్చు
- చాలా తేలికగా అర్థం చేసుకోవచ్చు
- చాలా తేలికగా మార్పులు చేతులు చేయవచ్చు

**ధీర్ఘసమాధాన ప్రశ్నలు మరియు జవాబులు**

### 1. Flow-chart ఉపయోగాలను తెల్పండి. What are the advantages of flow-chart?

**జ. Flow-chart ఉపయోగాలు**

1. **కమ్యూనికేషన్:** సంబంధిత లేదా ప్రమేయం ఉన్న వారందరికీ సిస్టమ్ యొక్క లాజిక్ ను తెలియజేయడానికి ఫ్లోచార్ట్ లు మంచి మార్గం.
2. **ప్రభావవంతమైన విశ్లేషణ:** ఫ్లోచార్ట్ సహాయంతో, సమస్యను మరింత ప్రభావవంతమైన మార్గంలో విశ్లేషించవచ్చు, తద్వారా ఖర్చు మరియు సమయం వృధా తగ్గుతుంది.
3. **సరైన డాక్యుమెంటేషన్:** ప్రోగ్రామ్ ఫ్లోచార్ట్ లు మంచి ప్రోగ్రామ్ డాక్యుమెంటేషన్ గా పనిచేస్తాయి, ఇది వివిధ ప్రయోజనాల కోసం అవసరమవుతుంది, ఇది విషయాలను మరింత సమర్థవంతంగా చేస్తుంది.
4. **సమర్థవంతమైన కోడింగ్:** సిస్టమ్స్ విశ్లేషణ మరియు ప్రోగ్రామ్ అభివృద్ధి దశలో ఫ్లోచార్ట్ లు గైడ్ లేదా బ్లూప్రింట్ గా పనిచేస్తాయి.
5. **సరైన డీబగ్గింగ్:** సమర్థవంతమైన ఫ్లోచార్ట్ డీబగ్గింగ్ ప్రక్రియలో సహాయపడుతుంది.
6. **ప్రోగ్రామ్ నిర్వహణ:** ఫ్లోచార్ట్ సహాయంతో ఆపరేటింగ్ ప్రోగ్రామ్ నిర్వహణ సులభం అవుతుంది. ప్రోగ్రామర్ ఆ భాగంలో మరింత సమర్థవంతంగా ప్రయత్నాలు చేయడానికి ఇది సహాయపడుతుంది.

2. Alogritham మరియు Flow-chart మధ్య గల బేదాలు వ్రాయుము.

What are the differences between algorithm and flow-chart ?

| S.No. | Alogritham  | Flow-chart   |
|-------|---|--|
| 1     | అల్గారిథమ్ అనేది సమస్యను పరిష్కరించడానికి దశల వారీ విధానం | ఫ్లోచార్ట్ అనేది డేటా ప్రవాహాన్ని చూపించడానికి వివిధ ఆకృతులచే సృష్టించబడిన రేఖాచిత్రం. |
| 2     | అల్గారిథమ్ అర్థం చేసుకోవడానికి సంక్లిష్టమైనది             | ఫ్లోచార్ట్ అర్థం చేసుకోవడం సులభం   |
| 3     | అల్గారిథమ్ లో సాదా వచనం ఉపయోగించబడుతుంది                  | ఫ్లోచార్ట్ లో, చిహ్నాలు / ఆకారాలు ఉపయోగించబడతాయి                                       |
| 4     | అల్గారిథమ్ డీబగ్ చేయడం సులభం                              | ఫ్లోచార్ట్ డీబగ్ చేయడం కష్టం   |
| 5     | అల్గారిథమ్ నిర్మించడం కష్టం                               | ఫ్లోచార్ట్ నిర్మించడం సులభం  |
| 6     | అల్గారిథమ్ ఏ నియమాలను అనుసరించదు                          | ఫ్లోచార్ట్ నిర్మించాల్సిన నియమాలను అనుసరిస్తుంది                                       |
| 7     | ప్రోగ్రామ్ కోసం సూడో (నకిలీ) కోడ్                         | ఫ్లోచార్ట్ అనేది ఆతర్కం యొక్క గ్రాఫికల్ ప్రాతినిధ్యం                                   |

3. ఇచ్చిన మూడు పూర్ణాంకాల సంఖ్యలలో అతి చిన్న విలువను కనుగొనడానికి అల్గారిథమ్ ను వ్రాయండి.

Write an algorithm to find smallest value in given three integer numbers జ.

An algorithm to find smallest value in given three integer numbers:

- పద్ధతి 1: ప్రారంభించండి
- పద్ధతి 2 : a, b, c అనే మూడు పూర్ణాంకాల వేరియబుల్లను ప్రకటించండి.
- పద్ధతి 3 : a, b, c కోసం విలువలను చదవండి.
- పద్ధతి 4 : 'a' 'b' కంటే ఎక్కువ మరియు 'b' 'c' కంటే ఎక్కువ అయితే  
ఎ) ప్రింట్ c చిన్నది.
- పద్ధతి 5 : 'a', 'b' కంటే ఎక్కువ మరియు 'c' 'b' కంటే ఎక్కువ అయితే  
ఎ) ప్రింట్ b చిన్నది
- పద్ధతి 6 : 'b', 'a' కంటే ఎక్కువ మరియు 'c' 'a' కంటే ఎక్కువ అయితే  
ఎ) ప్రింట్ a చిన్నది.
- పద్ధతి 7: ముగింపు.

4. ఇచ్చిన సంఖ్య ఫ్రైమ్ లో ఉందో లేదో కనుగొనడానికి అల్గారిథమ్ ను వ్రాయండి.

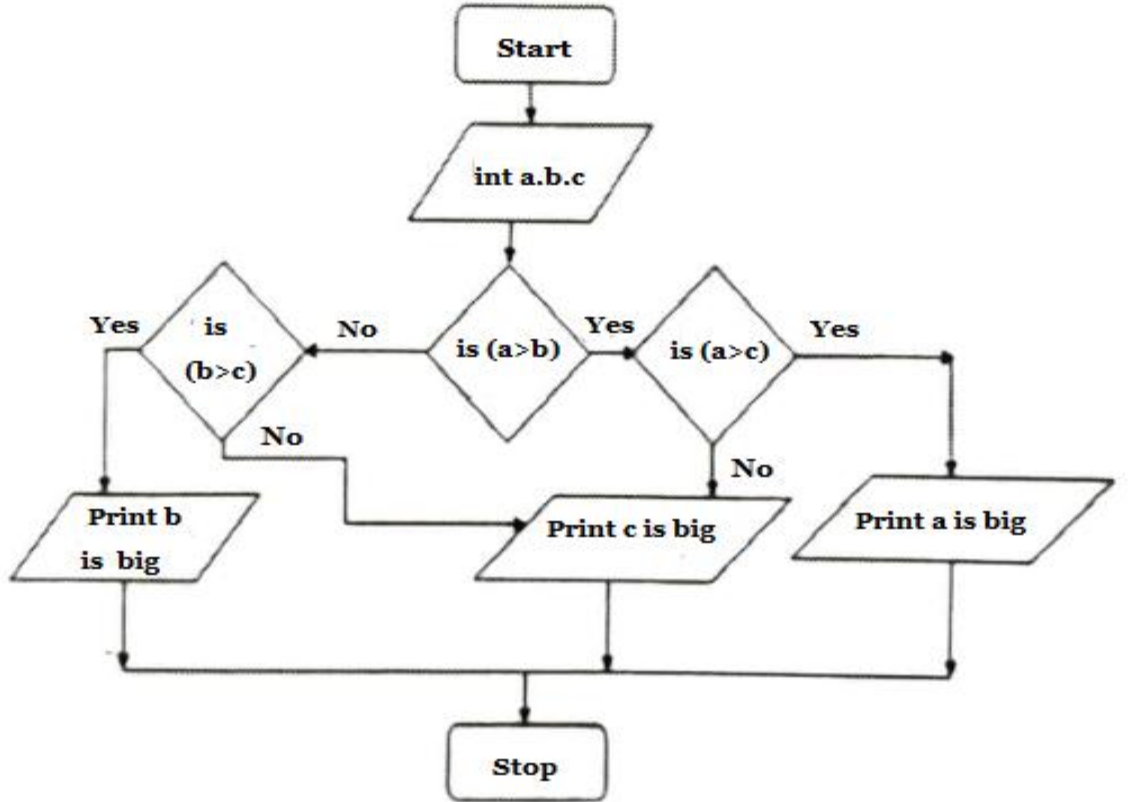
Write an algorithm to find whether the given number is prime or not .

జ. An algorithm to find whether the given number is prime or not.

- పద్ధతి 1: ప్రారంభించండి
- పద్ధతి 2: మూడు పూర్ణాంకాల వేరియబుల్స్  $n, i, p$  డిక్లైర్ చేయండి.
- పద్ధతి 3:  $n$  కోసం విలువను చదవండి.
- పద్ధతి 4:  $p$  కి 1 విలువను కేటాయించండి.
- పద్ధతి 5: లూప్  $i$  2 నుండి  $n$  ఉంటే ( $n \% i == 0$ )  
a) 0 నుండి  $p$  కి కేటాయించండి.
- పద్ధతి 6: ఎ)  $p, 1$  అయితే ప్రింట్  $n$  ప్రధానం.  
బి) లేకపోతే ప్రింట్  $n$  ప్రధానం కాదు.
- పద్ధతి 7: ఆపు.

6. మూడు సంఖ్యలలో పెద్ద సంఖ్యను కనుగొనుటకు flowchart ని గీయుము .

Draw a flow - chart to find the biggest value of a given three numbers.



5. ఇచ్చిన 4 అంకెల సంఖ్య రివర్స్ విలువను కనుగొనడానికి అల్గారిథమ్‌ను వ్రాయండి.

Write an algorithm to find the reverse value of given 4 digit number.

జ. An Algorithm to find the reverse value of given 4 digit number:

పద్ధతి 1: ప్రారంభించండి

పద్ధతి 2: d మరియు r అనే రెండు వేరియబుల్‌లను ప్రకటించండి మరియు  
వేరియబుల్  $r = 0$  ని ప్రారంభించండి.

పద్ధతి 3: అంకెలను చదవండి.

పద్ధతి 4: d అనేది 0 కంటే ఎక్కువగా ఉందో లేదో తనిఖీ చేసి, ఆపై 5వ పద్ధతి కు వెళ్లండి,  
పద్ధతి 9కి వెళ్లండి

పద్ధతి 5: rని 10తో గుణించి, ఫలితాన్ని rకి కేటాయించండి ( $r = r * 10$ )

పద్ధతి 6:  $r+d \% 10$ ని అమలు చేయండి ఫలితాన్ని rకి జోడించండి ( $r=r+d \% 10$ )

పద్ధతి 7:  $d/10$ ని అమలు చేయండి మరియు ఫలితాన్ని dకి కేటాయించండి ( $d=d/10$ )

పద్ధతి 8: పద్ధతి 4కి వెళ్లండి

పద్ధతి 9: ప్రింట్ r

పద్ధతి 10: ఆపు.

13. ఇచ్చిన సంఖ్య పాలిండ్రోమ్ కాదా అని తనిఖీ చేయడానికి అల్గారిథమ్‌ను వ్రాయండి.

Write an algorithm to check whether the given number palindrome or not.

జ. An Algorithm to check whether the given number palindrome or not

పద్ధతి 1: ప్రారంభించండి

పద్ధతి 2: అవసరమైన వేరియబుల్‌స్‌ని డిక్లార్ చేయండి మరియు ప్రారంభించండి

పద్ధతి 3: పాలిండ్రోమ్ ప్రాపర్టీ కోసం తనిఖీ చేయాల్సిన నెంబర్ ను నమోదు చేయండి

పద్ధతి 4: తాత్కాలిక వేరియబుల్‌తో నెంబర్ ను కేటాయించండి

పద్ధతి 5: లూప్ ఉపయోగించి సంఖ్యను రివర్స్ చేయండి

పద్ధతి 6: రివర్స్డ్ (reversed) నెంబర్ ను నిల్వ చేయండి

పద్ధతి 7: తాత్కాలిక వేరియబుల్‌లో నిల్వ చేయబడిన రివర్స్డ్ (reversed) నెంబర్  
మరియు ఒరిజినల్ నెంబర్ ను సరిపోల్చండి

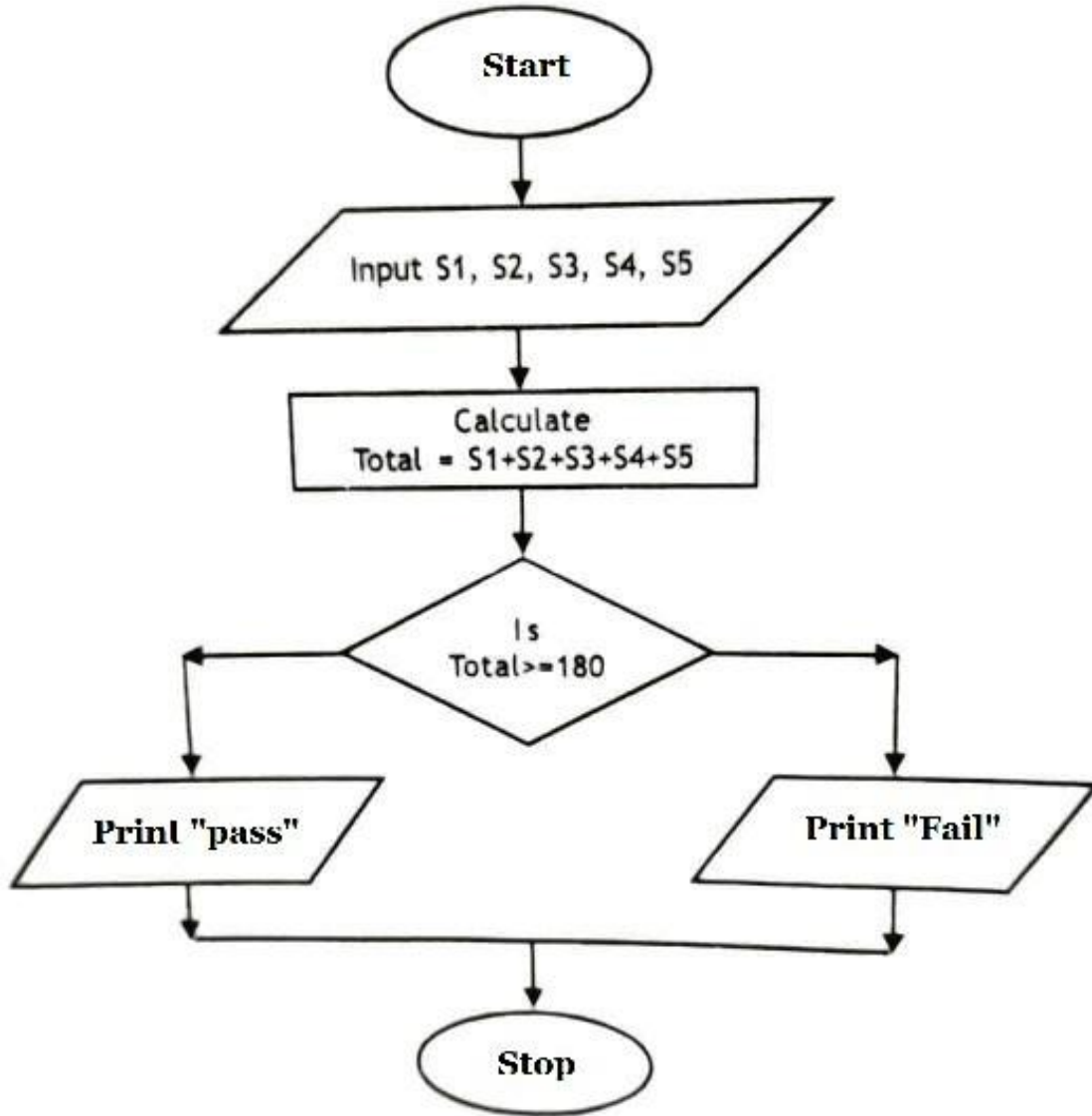
పద్ధతి 8: సంఖ్య మరియు దాని రివర్స్డ్ ఒకేలా ఉన్నట్లు గుర్తించినట్లయితే, ఆ సంఖ్యను  
పాలిండ్రోమ్ సంఖ్యగా ముద్రించండి

పద్ధతి 9: లేకపోతే, సంఖ్య పాలిండ్రోమ్ కాదు. పద్ధతి 10: ఆపు.



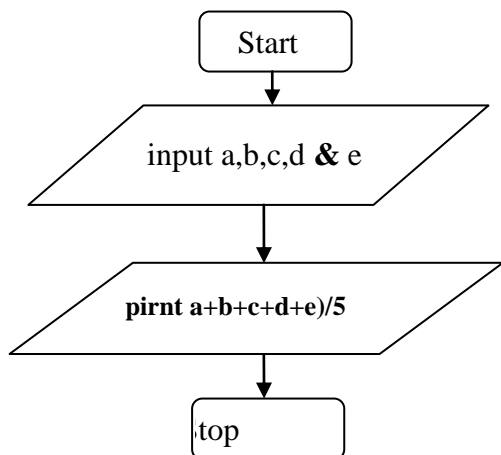
### More practice on related flow-charts

1. సెల్సియస్లో ఉష్ణోగ్రతను ఫారెన్హీట్గా మార్చడం.
  2. సంఖ్య యొక్క కారకం.
  3. ఫైబొనాక్సీ సీక్వెన్స్ (Fibonacci sequence).
  4. ఒక త్రిభుజం వైశాల్యం , దాని వైపులా ఇవ్వబడింది .
  5. రెండు సంఖ్యల GCD, HCFని కనుగొనడం.
  6. సిరీస్  $S = \frac{1}{2}! + 2/3! + \dots + n$ . యొక్క N నిబంధనలను కలుపుతోంది
7. Draw a flow-chart to find pass or failed of a student of a given 5 subject marks each subject maximum marks are 50, if total marks  $\geq 180$  results is pass or fail.



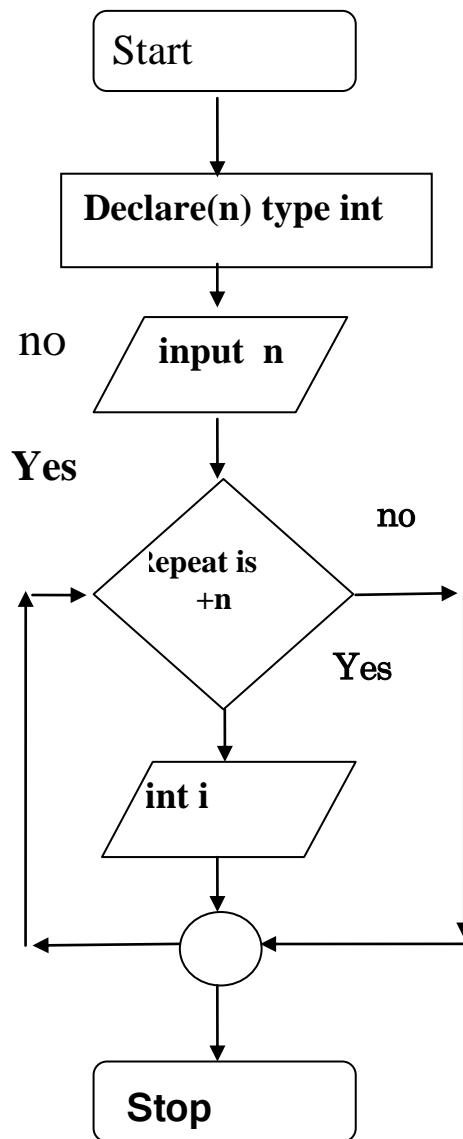
8. Draw a flow-chart to find the average of 5 given numbers.

∞ .



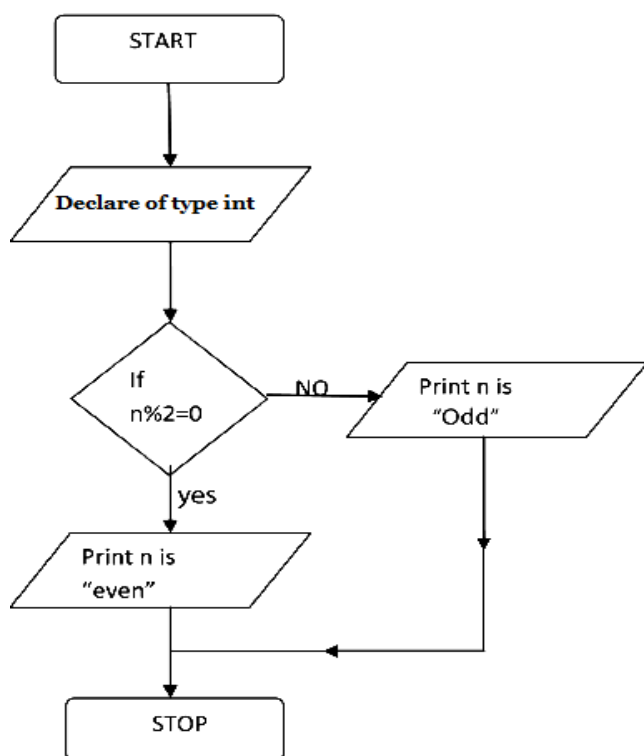
9. Draw a flow-chart to find the sum of 1 to n natural numbers.

∞.

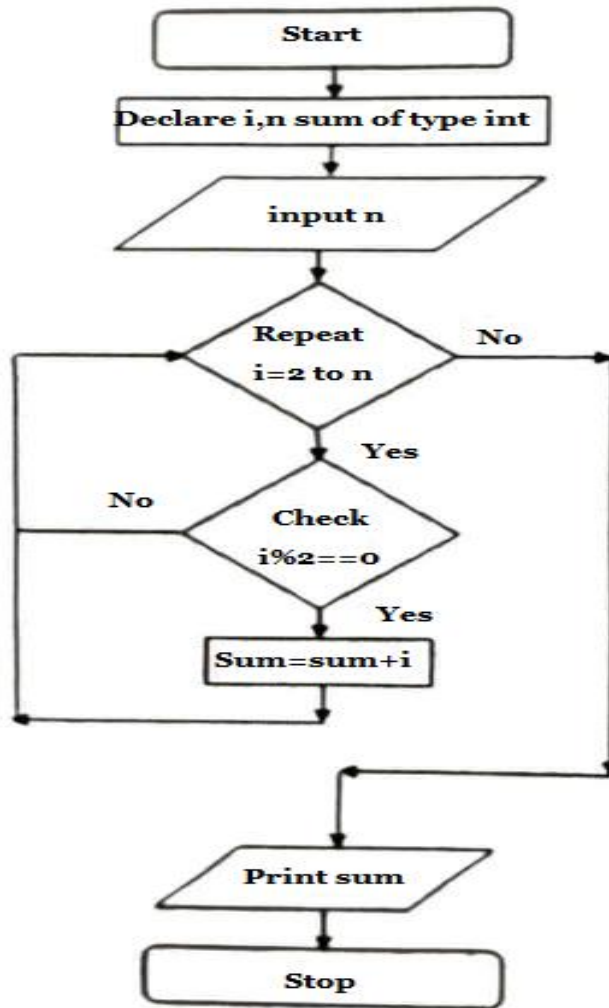


12. Draw a flow-chart to find whether the given number is even or odd.

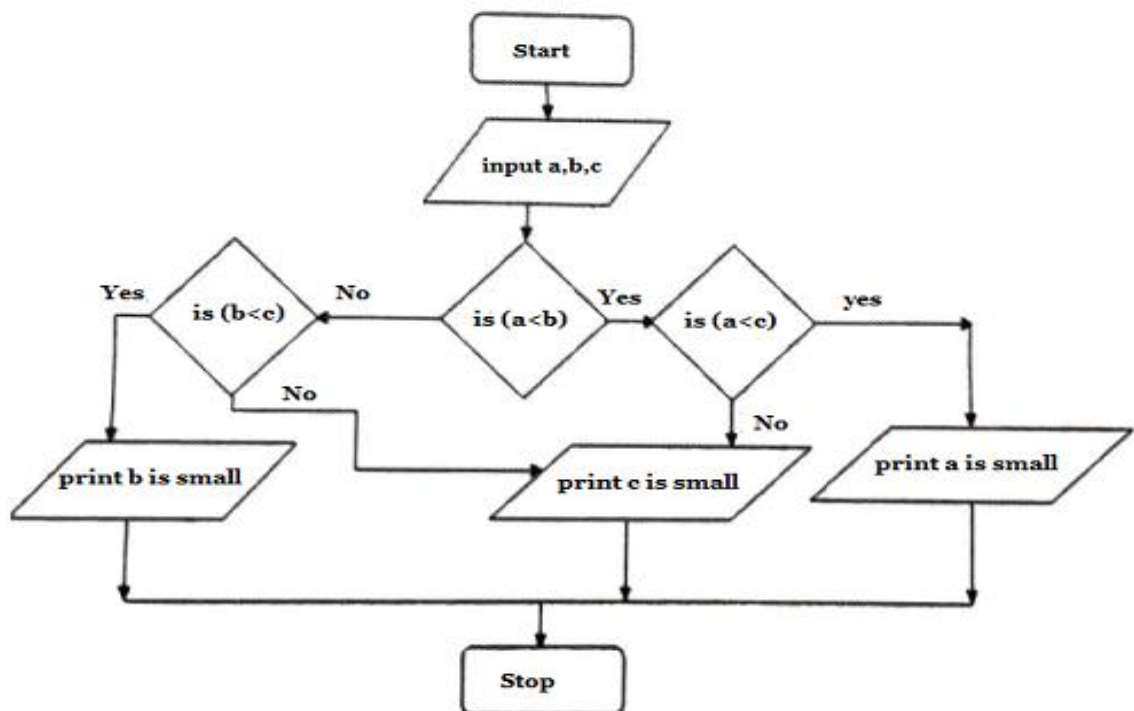
∞.



10. Draw a flow chart to find sum of 1 to n even numbers.



11. Draw a flow-chart to find the small value of given three numbers.


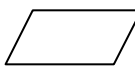
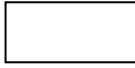

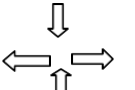
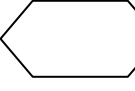



14. Write an algorithm to calculate the simple interest using  $(PNR / 100)$ .

జ. An Algorithm to calculate Simple interest :

- పద్ధతి 1: ప్రారంభించండి
- పద్ధతి 2: టైప్ పూర్ణాంకం యొక్క రెండు వేరియబుల్స్ p మరియు n డిక్లైర్ చేయండి.
- పద్ధతి 3: ఫ్లోటింగ్ పాయింట్ యొక్క రెండు వేరియబుల్స్ r మరియు si డిక్లైర్ చేయండి.
- పద్ధతి 4: p, n మరియు r విలువలను చదవండి .
- పద్ధతి 5: p, n మరియు r అనే మూడు వేరియబుల్స్ ని గుణించి 100తో భాగించండి మరియు ఫలిత విలువను si వేరియబుల్ కి కేటాయించండి.
- పద్ధతి 6: si విలువను ముద్రించండి.
- పద్ధతి 7: ఆపు.

15. What are the flow-chart symbols ?

| No | చిహ్నం  | చిహ్నం పేరు      | వినియోగం              | ఫ్లో చార్ట్ లో చిహ్నాన్ని ఉపయోగించడం                            |
|----|---|------------------|-----------------------|---|
| 1  |   | ఓవల్             | ప్రారంభం/ఆపు          | ఇది ప్రారంభించడానికి మరియు ఆపడానికి ఉపయోగించబడుతుంది            |
| 2  |  | సమాంతర చతుర్భుజం | ఇన్ పుట్ / అవుట్ పుట్ | ఇది ఇన్ పుట్ లేదా అవుట్ పుట్ డేటా కోసం ఉపయోగించబడుతుంది         |
| 3  |  | దీర్ఘచతురస్రం    | ప్రక్రియ              | ఇది డేటాను ప్రాసెస్ చేయడానికి ఉపయోగించబడుతుంది                  |
| 4  |  | డైమండ్           | నిర్ణయం తీసుకోవడం     | ఇది షరతు ఇవ్వడానికి ఉపయోగించబడుతుంది                            |
| 5  |  | బాణాలు           | ప్రవాహం               | ఇచ్చిన డేటా యొక్క ప్రవాహాన్ని అందించడానికి ఇది ఉపయోగించబడుతుంది |
| 6  |  | షడ్భుజి          | తయారీ                 | ఇది "ఫర్, అయితే, డూ-వైల్" వంటి లూప్ ల కోసం ఉపయోగించబడుతుంది     |
| 7  |  | వృత్తం           | కనెక్టర్              | ఇది లైన్లను కనెక్ట్ చేయడానికి ఉపయోగించబడుతుంది                  |

## UNIT-II

### Features of 'C'

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

1. C language ని ఎవరు అభివృద్ధి చేసారు? Who developed C Language?

జ. C' భాషను 1970లో బెల్ లాబొరేటరీలలో "డెన్నిస్ రిట్చీ" కనుగొన్నారు.

2. C program యొక్క ఆకారమును వ్రాయుము. Write the structure of C program.

జ. The structure of C program is

|                            |
|----------------------------|
| Documentation Section      |
| Link Section               |
| Global Declaration Section |
| Definition Section         |
| main () {<br>-----<br>}    |
| Sub Program 1              |
| Sub program 2              |
| Sub Program n              |

3. క్యారెక్టర్ సెట్ అంటే ఏమిటి? 'C' భాషలో ఉపయోగించిన అక్షరాన్ని వ్రాయండి.

What is character set? Write the character which used in 'C' language.

జ. **Character set:** ప్రోగ్రామ్ అమలు చేయబడిన కంప్యూటర్లను బట్టి పదాలు, సంఖ్యలు మరియు వ్యక్తీకరణలను రూపొందించడానికి క్యారెక్టర్ సెట్ను ఉపయోగించవచ్చు.

C లోని అక్షరాలు క్రింది రకాలుగా వర్గీకరించబడ్డాయి

1. A నుండి Z (లేదా) a నుండి z వరకు అక్షరాలు
2. అంకెలు (సంఖ్యలు 0 నుండి 9 వరకు)
3. ప్రత్యేక అక్షరాలు, ; :> <!= # \* మొదలైనవి.
4. తెల్లని ఖాళీలు (White spaces).

4. Constant అనగా నేమి? What is a Constant ?

జ. **Constants:** ప్రోగ్రామ్ అమలు సమయంలో మారని వాటిని స్థిరాంకాలు అంటారు.

## 5. వివిధ రకాల Constant లను వ్రాయండి. What are different types of constants?

జ. Constants: ప్రోగ్రామ్ అమలు సమయంలో మారని వాటిని స్థిరాంకాలు అంటారు.

స్థిరాంకాల రకాలు:

- సంఖ్యా స్థిరాంకాలు (Numeric Constants)
- అక్షర స్థిరాంకాలు (Character Constants)
- స్ట్రింగ్ స్థిరాంకాలు (String Constants)

## 6. Numeric constant అనగా నేమి? What is a Numeric constant?

జ. Numeric Constants: సంఖ్యా స్థిరాంకాలు, పేరు సూచించినట్లుగా, సంఖ్యలు, ఐచ్ఛిక చిహ్నం మరియు ఐచ్ఛిక వ్యవధిని కలిగి ఉంటాయి. అవి ఇంకా రెండు రకాలుగా

- విభజించబడ్డాయి:
- (ఎ) పూర్ణాంక స్థిరాంకాలు
  - (బి) వాస్తవ స్థిరాంకాలు

## 7. String constant అనగా నేమి? What is a String constant?

జ. String Constant: కొన్ని క్యారెక్టర్ ల సముదాయాన్ని స్ట్రింగ్ (String) అంటారు. స్ట్రింగ్ అంటే నెంబర్లు మరియు అక్షరాల కలయిక స్ట్రింగులను డబుల్ కోటేషన్ ల మధ్యలో వ్రాయాలి. స్ట్రింగ్ లో maximum 255 క్యారెక్టర్ లను ఇవ్వవచ్చును. స్ట్రింగ్ స్థిరాంకాల యొక్క ఉదాహరణలు క్రింది విధముగా ఉంటాయి.

- "My name is Raju"
- "Lecturer"
- "Salary is 65000.00"

## 8. Variable అనగా నేమి? What is a Variable?

జ. Variable: C లోని వేరియబుల్స్, డేటాను నిల్వ చేసే మెమరీ లొకేషన్ పేరు వేరియబుల్. మనం వేరియబుల్ యొక్క విలువను మార్చవచ్చును మరియు మనం దానిని అనేక సార్లు కూడా తిరిగి ఉపయోగించుకోవచ్చు దానినే వేరియబుల్ అంటారు.

ఉదాహరణ: Avg (Average) Ht (height)  
Tot((Total)  
Tot \_ stu(Total students)

## 9. String variable అనగా నేమి? What Is a String variable?

జ. String Variables: స్ట్రింగ్ వేరియబుల్ లో లెటర్స్ నెంబర్స్ మరియు ఇతర క్యారెక్టర్లు ఉంటాయి వీటిని స్ట్రింగ్ వేరియబుల్ అని అంటారు.

## 10. Single string constant ఆనగా నేమి?

What is a single string constant?

- జ. **Single string constant:** ఒకే అక్షరం స్ట్రింగ్ స్థిరాంకం సమానమైన పూర్ణాంకం విలువను కలిగి ఉండదు, ఇది రెండు బైట్లను ఆక్రమిస్తుంది, ఒకటి A యొక్క ASCII కోడ్ మరియు మరొకటి NULL అక్షరం కోసం విలువ 0, ఇది అన్ని స్ట్రింగ్లను ముగించడానికి ఉపయోగించబడుతుంది.

## 11. 'C' భాష యొక్క ప్రాథమిక డేటా రకాలు ఏమిటి?

What are the fundamental data types of 'C' language?

- జ. **Basic Data Types:** సి భాషలో నాలుగు ప్రాథమిక డేటా రకాలు ఉన్నాయి. అవి: Integer, character, floating point and double.

|                  | Data Type | Occupation of memory | Range                   |
|------------------|-----------|----------------------|-------------------------|
| Integer          | int       | 2 Bytes              | -32,768 to 32,767       |
| Character        | char      | 1 Byte               | - 128 to 127            |
| Floating Point   | float     | 4 Bytes              | -3.4 e-38 to 3.4°e-38   |
| Double Precision | double    | 8 Bytes              | -1.7 e-308 to 1.7 e-308 |

## 12. Arithmetical ఆపరేటర్లు ఏమిటి?

What are the arithmetical operators?

- జ. **Arithmetic Operators:** ఇవి రెండు ఒపెరాండల్లో అంకగణిత కార్యకలాపాలను నిర్వహించడానికి ఉపయోగించబడతాయి. a = 5 మరియు b = 3.

| Operator | Meaning                    | Example | Result |
|----------|----------------------------|---------|--------|
| +        | Addition or Unary Plus     | a+b     | 8      |
| -        | Subtraction or Unary Minus | a-b     | 2      |
| *        | Multiplication             | a*b     | 15     |
| /        | Division                   | a/b     | 1      |
| %        | Modulo Division            | a % b   | 2      |

13. Bitwise ఆపరేటర్లు అంటే ఏమిటి What are the bitwise operators?

జ. **Bitwise Operators:** బిట్ వైస్ మరియు బిట్ షిఫ్ట్ ఆపరేటర్లు బైనరీ ఫార్మాట్ ప్రకారం బిట్ స్థాయిలో వేరియబుల్స్ కంటెంట్లను మార్చడానికి ఉపయోగించబడతాయి.

| Symbol | Name of the Operator     | Example   |
|--------|--------------------------|-----------|
| -      | Unary bitwise complement | -op2      |
| &      | Bitwise AND              | op1 & op2 |
|        | Bitwise inclusive OR     | op1   op2 |
| ^      | Bitwise exclusive OR     | op1 ^ op2 |

14. షరతులతో కూడిన ఆపరేటర్లు ఏమిటి? What are the Conditional operators?

జ. **Conditional Operator:** ఈ ఆపరేటర్ మూడు వ్యక్తికరణలపై పరీక్షించబడుతుంది. షరతు మొదటి షరతు నిజమైతే, రెండవ వ్యక్తికరణ అమలు చేయబడుతుంది లేకపోతే మూడవ వ్యక్తికరణ అమలు చేయబడుతుంది.

Operator:                   ?:

Syntax:                   (exp 1? exp 2: exp 3)

15. ఇన్పుట్ మరియు అవుట్పుట్ స్టేట్మెంట్లు అంటే ఏమిటి

What are input and output statements?

జ.

|             | Input Statements | Output Statements |
|-------------|------------------|-------------------|
| Formatted   | scanf()          | printf()          |
| Unformatted | getchar().gets() | putchar() puts()  |

16. printf () మరియు scanf()స్టేట్మెంట్ల సింటాక్స్(syntax) అంటే ఏమిటి.

What is syntax of printf () and scanf()statements.

జ. **Syntax of printf:** printf("control string", arg1, arg2, ....arg n);

ఇక్కడ కంట్రోల్ స్ట్రీంగ్ అనేది ఫార్మాట్ చేయబడిన సమాచారాన్ని కలిగి ఉన్న స్ట్రీంగ్ మరియు arg1, arg2,...arg n అనేది అవుట్పుట్ డేటా అంశాలను సూచించే ఆర్గ్యుమెంట్లు.

**Syntax of scanf:** scanf ("control strings", &arg1,&arg2, ....&arg n);

నియంత్రణ స్ట్రీంగ్ నిర్దిష్ట అవసరమైన ఫార్మాటింగ్ సమాచారాన్ని కలిగి ఉన్న స్ట్రీంగ్ను సూచిస్తుంది మరియు arg1, arg2..arg n అనేది వ్యక్తిగత ఇన్పుట్ డేటా అంశాలను సూచించే ఆర్గ్యుమెంట్లు.



17. **getchar()** అనగా నేమి? What is a **getchar()** function in C?

జ. **Getchar()**:ని లైబ్రరీ ఫంక్షన్ **getchar()**లో ఉపయోగించి ఒకే అక్షరాన్ని కంప్యూటర్లోకి నమోదు చేయవచ్చు. ఇది ప్రామాణిక ఇన్పుట్ పరికరం నుండి ఒకే అక్షరాన్ని అందిస్తుంది. ఫంక్షన్కు ఎటువంటి వాదనలు అవసరం లేదు.

**Syntax:** < character variable >= getchar();

18. సాధారణ 'if' స్టేట్మెంట్ గురించి వ్రాయండి. Write about Simple 'if' statement.

జ. **Simple 'if' statement:** షరతులతో కూడిన వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది మరియు వ్యక్తీకరణ నిజమైతే, ప్రకటనలు అమలు చేయబడతాయి. వ్యక్తీకరణ తప్పు అయితే, స్టేట్మెంట్లు దాటవేయబడతాయి మరియు తదుపరి స్టేట్మెంట్లతో అమలు కొనసాగుతుంది.

**Syntax:** if conditional expression >

statements;

19. **if-else** స్టేట్మెంట్ యొక్క సింటాక్స్ ను వ్రాయండి.

Write the syntax of **if-else** statement.

జ. **if-else statement:** షరతులతో కూడిన వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది మరియు వ్యక్తీకరణ నిజమైతే స్టేట్మెంట్లు-1 అమలు చేయబడుతుంది. వ్యక్తీకరణ తప్పు అయితే స్టేట్మెంట్లు 2 అమలు చేయబడతాయి.

**Syntax:** If conditional expression >

statements-1;

else

statements-2;

20. How many types of loops are available in C ?

జ. C'లో మూడు రకాలు లూప్లు ఉన్నాయి. అవి

- while loop
- do-while loop
- for loop

21. **Break** statement అనగా నేమి? What is a **break** statement ?

జ. **Break:** బ్రేక్ అనేది లూప్ను ముగించడానికి లేదా బ్లాక్ నుండి నిష్క్రమించడానికి ఉపయోగించే కీలక పదం. నియంత్రణ లూప్ లేదా బ్లాక్ తర్వాత తదుపరి స్టేట్మెంట్కు వెళుతుంది.

**Syntax:** Break;

**22. Continue statement అనగా నేమి? What is a continue statement ?**

జ. **Continue:** కొనసాగించు అనేది లూప్ యొక్క తదుపరి పునరావృతాన్ని కలిగి ఉండటానికి ఉపయోగించే కీవర్డ్. **Syntax:** Continue;

**23. switch() స్టేట్ మెంట్ యొక్క syntax వ్రాయండి.**

**Write the syntax of switch ( ) statement.**

జ. **Switch statement:** అనేక ప్రత్యామ్నాయాలలో స్టేట్ మెంట్ ను ఎంచుకోవడానికి స్విచ్ స్టేట్ మెంట్ ఉపయోగించబడుతుంది.

**Syntax:**            switch (variable)  
                         {  
                         Case:    labell:    statements1;  
                         Case:    label2:    statements2;  
                         -----  
                         default:            statements n;    }

**24. Goto statement గురించి వ్రాయండి. Write about 'goto' statement.**

జ. **Goto statement:** Goto స్టేట్ మెంట్ ద్వారా ప్రోగ్రాం యొక్క ఎగ్జిక్యూట్ ను మార్చుకొని వచ్చును ఒక ప్రదేశం నుంచి వేరొక ప్రదేశానికి మార్చవచ్చు. దీనిని లేబుల్ అనునది కంట్రోల్ ను transfer చేసే లొకేషన్ ను తెలుపుతుంది.

**Syntax:**            goto <label>:  
                         -----

                         label: statements;

లేబుల్ అనేది లక్ష్యం స్టేట్ మెంట్ ను లేబుల్ చేయడానికి ఉపయోగించే ఐడెంటిఫైయర్, దీని నియంత్రణ లక్ష్య ప్రకటనకు బదిలీ చేయబడుతుంది.

**25. Numeric variable అనగా నేమి? What is a Numeric variable?**

జ. **Numeric variable:** సంఖ్యను కలిగి ఉన్న వేరియబుల్, సంఖ్యా వేరియబుల్ గా చెప్పబడుతుంది, ఇది పూర్ణాంకం, ఫ్లోట్ మరియు డబుల్ డేటా రకంలో ఏదైనా ఒకటిగా చెప్పవచ్చు.

**26. Logical operators వ్రాయండి. What are the Logical operators?**

జ. **Logical Operators:** లాజికల్ ఆపరేటర్లు అవి: &&(and), || (or) and ! (not).

## 27. Puchar() అనగా నేమి? What is a puchar()?

జ. **Puchar():** పుట్ క్యారెక్టర్ ఫంక్షన్ ద్వారా ఒక క్యారెక్టర్ ను display చేయవచ్చును స్టాండర్డ్ అవుట్ పుట్ డివైస్ మోనిటర్ ద్వారా క్యూప్టెన్ display చేయవచ్చును తప్పనిసరిగా ఆర్గ్యుమెంట్ అవసరం ఉంటుంది **Syntax:** puchar (< character variable>);

## 28. C లో gets () function అనగా నేమి? What is a gets () function in C?

జ. C లైబ్రరీలో ఉన్న getch() ఫంక్షన్ ద్వారా ఒక క్యారెక్టర్ ను కీబోర్డ్ ద్వారా ఇన్ పుట్ గా ఇవ్వవచ్చును మీ ఫంక్షన్ కు ఎటువంటి ఆర్గ్యుమెంట్ అవసరం లేదు.

**Syntax:** gets (<string\_ type variable>);

## 29. C లో puts () function అనగా నేమి? What Is a puts () function in C?

జ. **Puts ( ):** పుట్ క్యారెక్టర్ ఫంక్షన్ ద్వారా ఒక క్యారెక్టర్ ను Display చేయవచ్చును. స్టాండర్డ్ అవుట్ పుట్ డివైస్ మోనిటర్ ద్వారా క్యారెక్టర్ ను ఇస్తే చేయవచ్చును తప్పనిసరిగా ఆర్గ్యుమెంట్ అవసరం ఉంటుంది. **Syntax:** puts (s);

## 30. Nested loop అనగా నేమి? What is Nested loop?

జ. **Nested loop:** లూప్ లోని లూప్ రాసినట్లయితే దానిని nested loop అని అంటారు. loopలో(while,do..while,for)లోపలి మరియు బయటి లూప్లు ఒకే రకంగా ఉండవచ్చు లేదా ఒకే రకంగా ఉండకపోవచ్చు.

ధీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

## 1. C లో స్థిరంగా(Constant) ఉండేవి ఏమిటి? ఉదాహరణలతో వివరించండి.

**What are constant in C? Explain with examples.**

జ. ప్రోగ్రామ్ అమలు సమయంలో మారని వాటిని స్థిరాంకాలు అంటారు. స్థిరాంకాలను ఇలా వర్గీకరించవచ్చు:

- సంఖ్యా స్థిరాంకాలు (Numeric Constants)
- అక్షర స్థిరాంకాలు (Character Constants)
- స్ట్రింగ్ స్థిరాంకాలు (String Constants)

1. **సంఖ్యా స్థిరాంకాలు:** సంఖ్యా స్థిరాంకాలు, పేరు సూచించినట్లుగా, సంఖ్యలు, ఐచ్ఛిక చిహ్నం మరియు ఐచ్ఛిక వ్యవధిని కలిగి ఉంటాయి. అవి ఇంకా రెండు రకాలుగా విభజించబడ్డాయి:

- (ఎ) పూర్ణాంక స్థిరాంకాలు (Integer Constants)
- (బి) వాస్తవ స్థిరాంకాలు (Real Constants)

ఎ. పూర్ణాంక స్థిరాంకాలు: పూర్ణసంఖ్య పూర్ణాంక స్థిరాంకం. పూర్ణ సంఖ్య స్థిరాంకాలు దశాంశ బిందువును కలిగి ఉండవు. ఇవి వాటికి చెందిన సంఖ్యా వ్యవస్థలను బట్టి మూడు రకాలుగా విభజించబడ్డాయి. అవి:

- i. దశాంశ పూర్ణాంక స్థిరాంకాలు (Decimal integer constants)
- ii. అష్ట పూర్ణాంకాల స్థిరాంకాలు (Octal integer constants)
- iii. హెక్సాడెసిమల్ పూర్ణాంక స్థిరాంకాలు (Hexadecimal integer constants)

బి. వాస్తవ స్థిరాంకాలు: ఫ్లోటింగ్ పాయింట్ స్థిరాంకాలు అని కూడా పిలువబడే నిజమైన స్థిరాంకాలు రెండు రూపాల్లో వ్రాయబడతాయి:

- i. భిన్న రూపం (Fractional form)
- ii. ఘాతాంక రూపం (Exponential form)

2. అక్షర స్థిరాంకాలు: ఒకే కోట్స్ (') చేర్చబడిన ఏదైనా అక్షరాన్ని అక్షర స్థిరాంకం అంటారు. అక్షర స్థిరాంకం, ఒకే వర్ణమాల, ఒకే అంకె లేదా ఒకే కోట్లలో ఉంచబడిన ఒకే ప్రత్యేక అక్షరం కావచ్చు. గరిష్టంగా 1 అక్షరం నిడివిని కలిగి ఉంటుంది.

ఇవి కొన్ని ఉదాహరణలు 'C', 'c', '.', '\*'

3. స్ట్రింగ్ స్థిరాంకాలు: స్ట్రింగ్ స్థిరాంకం అనేది గరిష్ట పొడవు 255 అక్షరాలతో డబుల్ కోట్లతో జతచేయబడిన ఆల్ఫాన్యూమరిక్ అక్షరాల క్రమం. చెల్లుబాటు అయ్యే స్ట్రింగ్ స్థిరాంకాల యొక్క ఉదాహరణలు క్రిందివి:

- "నా పేరు కృష్ణ"
- "గుమాస్తా"
- "జీతం 18000.00"

చెల్లని స్ట్రింగ్ స్థిరాంకాల యొక్క ఉదాహరణలు క్రిందివి:

- "నా పేరు కృష్ణ"
- అక్షరాలు డబుల్ కోటేషన్ మార్కులలో చేర్చబడలేదు.
- "నా పేరు కృష్ణ - ముగింపు డబుల్ కోటేషన్ మార్క్ లేదు.
- 'నా పేరు కృష్ణ' - పాత్రలు డబుల్ కోటేషన్ మార్కులతో జతచేయబడవు.

## 2. Variable అనగా నేమి? వివిధ రకాలను ఉదాహరణ లతో తెల్పండి.

What is variable? Explain the types of variable in C with example.

**జ.Variables:** ఇవి ఆబ్జెక్ట్ ల పేర్లు, ప్రోగ్రామ్ అమలు సమయంలో వాటి విలువలను మార్చవచ్చును.

C లో అనేక రకాల వేరియబుల్స్ ఉన్నాయి:

- ఎ ) స్థానిక వేరియబుల్ (Local variable)
- బి) గ్లోబల్ వేరియబుల్ (Global variable)
- సి) స్టాటిక్ వేరియబుల్ (Static variable)
- డి) ఆటోమేటిక్ వేరియబుల్ (Automatic variable)
- ఇ) బాహ్య వేరియబుల్ (External variable)

ఎ) లోకల్ వేరియబుల్: ఫంక్షన్ లేదా బ్లాక్ లోపల డిక్లెర్ చేయబడిన వేరియబుల్ లోకల్ వేరియబుల్ అని పిలుస్తారు.

```
Example: void function1(){
           int x=10; // a local variable
        }
```

బి) గ్లోబల్ వేరియబుల్: ఫంక్షన్ లేదా బ్లాక్ వెలుపల ప్రకటించబడిన వేరియబుల్ను గ్లోబల్ వేరియబుల్ అంటారు. ఏదైనా ఫంక్షన్ గ్లోబల్ వేరియబుల్ విలువను మార్చగలదు. ఇది అన్ని ఫంక్షన్లకు అందుబాటులో ఉంటుంది. ఇది బ్లాక్ ప్రారంభంలో ప్రకటించబడాలి.

```
Example: int value=30; // a global variable
         void function1(){
           int a=20; // a local variable }
        }
```

సి) స్టాటిక్ వేరియబుల్: స్టాటిక్ తో డిక్లెర్ చేయబడిన వేరియబుల్ కీవర్డ్ ని స్టాటిక్ వేరియబుల్ అంటారు. ఇది బహుళ ఫంక్షన్ కాల్ల మధ్య దాని విలువను కలిగి ఉంటుంది.

```
Example, void function1() {
           int a=10; // A local variable
           static int b=10; // A static variable
           a=a+1;
           b=b+1;
           printf(“%d,%d”,a,b); }
        }
```

డి) ఆటోమేటిక్ వేరియబుల్: బ్లాక్ లోపల డిక్లేర్ చేయబడిన C లోని అన్ని వేరియబుల్స్ డిఫాల్ట్ గా ఆటోమేటిక్ వేరియబుల్స్. మనము స్పష్టంగా ఒక ప్రకటించవచ్చు. ఆటో కీవర్డ్ ఉపయోగించి ఆటోమేటిక్ వేరియబుల్.

```
Example: void main(){
    int a=80; // a local variable (it is also automatic variable)
    auto int b=50; // an automatic variable
}
```

ఇ) ఎక్స్ టర్నల్ వేరియబుల్: ఎక్స్ టర్నల్ వేరియబుల్ని ఉపయోగించడం ద్వారా మనం బహుళ సి సోర్స్ ఫైల్లలో వేరియబుల్ను షేర్ చేయవచ్చు. బాహ్య వేరియబుల్ని ప్రకటించడానికి, మీరు ఎక్స్టర్నల్ కీవర్డ్ని ఉపయోగించాలి. ఎక్స్ టర్నల్

**Syntax:** extern int a=10; // external variable (also a global variable)

3. 'C'లో డేటా రకాలు ఏమిటి? వాటిని వివరించండి.

**What are the data types in 'C' ? Explain them.**

జ. C లో ప్రధానంగా డేటా రకాలు:

1. ప్రాథమిక (ప్రాథమిక డేటా రకాలు) (Primary (Fundamental data types))
2. వినియోగదారు నిర్వచించిన డేటా రకాలు (User defined data types)
3. ఉత్పన్నమైన డేటా రకాలు (Derived data types)
4. ఖాళీ డేటా సెట్. సమాచార తరహా (Empty data set. data type)

1. **ప్రైమరీ డేటా రకాలు:** సి లాంగ్వేజ్ లో ముఖ్యముగా 4 ప్రాథమిక (లేదా) ప్రాథమిక డేటా రకాలు. అవి Int, float, char, మరియు డబుల్. వీటిని ప్రైమరీ లేదా ఫండమెంటల్ డేటాటైప్స్ అని కూడా అంటారు.

2. **వినియోగదారు నిర్వచించిన డేటా రకాలు:** User నిర్వచించిన డేటా రకాలు userల ప్రోగ్రామ్ల ద్వారా నిర్వచించబడతాయి. అవి typedef, sizeof, enum, empty or null డేటా రకాలు ఫంక్షన్ని ఉపయోగించబడతాయి. enum అనేది మరొకటి గణించబడిన డేటా రకాల లో నిర్వచించబడిన డేటా రకాలను ఉపయోగించింది.

**Syntax:** enum identifier { value 1, value2, value 3.....value n}

ఉదా: enum months { jan =1, feb=2, march =3.....dec =12}

3. **ఉత్పన్నమైన లేదా ద్వితీయ డేటా రకాలు:** ఉత్పన్నమైన డేటా రకాలు ప్రాథమిక డేటా రకాల ద్వారా నిర్వచించబడతాయి. అవి array, pointers, functions, structures and files.

4. ఖాళీ డేటా రకాలు: C లో సాధారణంగా వాయిడ్ అనేది ఖాళీ డేటా రకం మరియు రిటర్న్ టైప్ గా ఉపయోగిస్తాము. C లో పాయింట్ ని ప్రకటించడానికి శూన్యం యొక్క మరొక ఉపయోగం. అది ఖచ్చితంగా లేనప్పుడు పాయింట్ ద్వారా ఏ డేటా రకం పరిష్కరించబడుతుంది.

C లోని డేటా రకాలు:

| Type                             | Size (in Bytes) | Range                             |
|----------------------------------|-----------------|-----------------------------------|
| char or signed char              | 1               | -128 to 127                       |
| unsigned char                    | 1               | 0 to 255                          |
| int or signed int                | 2               | -32,768 to 32,767                 |
| unsigned int                     | 2               | 0 to 65,535                       |
| short int or signed short<br>int | 1               | -128 to 127                       |
| long int or unsigned long<br>int | 4               | -2,147,483,648 to<br>2,147,483,64 |
| unsigned long int                | 4               | 0 to 4,294,967,295                |
| float                            | 4               | -3.4 e-38 to 3.4 "e-38            |
| double                           | 8               | -1.7 e-308 to 3.4 e-308           |
| long double                      | 10              | -3.4e-4932 to +3.4e-49            |

4. C లో ఆపరేటర్లు ఏమిటి? వాటిని వివరించండి. What are the operators in C? Explain them.

జ. C లో ఉపయోగించే ఆపరేటర్:

1. అర్థమెటిక్ ఆపరేటర్ (Arithmetic Operators)
2. రిలేషనల్ ఆపరేటర్ (Relational Operators)
3. లాజికల్ ఆపరేటర్ (Logical Operators)
4. అసైన్మెంట్ ఆపరేటర్ (Assignment Operators)
5. ఇంక్రిమెంట్ & డిక్రిమెంట్ ఆపరేటర్ (Increment & Decrement Operators)
6. షరతులతో కూడిన ఆపరేటర్ (Conditional Operators)
7. బిట్ వైస్ ఆపరేటర్ (Bitwise Operators)
8. కామా ఆపరేటర్ (Comma Operators)

1. అంకగణిత ఆపరేటర్: ఇవి రెండు ఒపెరాండలలో అంకగణిత కార్యకలాపాలను

నిర్వహించడానికి ఉపయోగించబడతాయి. Let a = 5 and b = 3.

| Operator | Meaning                    | Example | Result |
|----------|----------------------------|---------|--------|
| +        | Addition or Unary Plus     | a+b     | 3      |
| -        | Subtraction or Unary Minus | a-b     | 2      |
| *        | Multiplication Division    | a*b     | 15     |
| /        | Division                   | a/b     | 1      |
| %        | Modulo                     | a%b     | 2      |

2. రిలేషనల్ ఆపరేటర్లు: ఈ ఆపరేటర్ రెండు ఆపరాండలు లేదా రెండు వ్యక్తీకరణల మధ్య

సంబంధాన్ని గుర్తించడానికి ఉపయోగిస్తారు. ఈ ఫలితాలు "ఒప్పు" లేదా "తప్పు". ఉదా: a = 5 మరియు b=3.

| Operator | Meaning                  | Example | Result |
|----------|--------------------------|---------|--------|
| >        | Greater than             | a>b     | True   |
| <        | Less than                | a<b     | False  |
| >=       | Greater than or equal to | a>= b   | True   |
| <=       | Less than or equal to    | a<= b   | False  |
| ==       | Is equals to             | a==b    | False  |
| !=       | Not equals               | a!=b    | True   |

3. లాజికల్ ఆపరేటర్లు: ఈ ఆపరేటర్లు రెండు వ్యక్తీకరణలను పరీక్షించడానికి ఉపయోగిస్తారు. ఈ ఫలితాలు "ఒప్పు" లేదా "తప్పు".

4. బిట్ వైస్ ఆపరేటర్లు: ఈ ఆపరేటర్లు బిట్ స్థాయిలో డేటాను మార్చడానికి ఉపయోగిస్తారు.

5. ఇంక్రిమెంట్ మరియు డిక్రిమెంట్ ఆపరేటర్లు: ఈ ఆపరేటర్లు ఒపెరాండ్కు విలువను కేటాయించడానికి ఉపయోగిస్తారు.

6. షరతులతో కూడిన ఆపరేటర్: ఈ ఆపరేటర్ మూడు వ్యక్తీకరణలపై పరీక్షించబడుతుంది. షరతు మొదటి షరతు నిజమైతే, రెండవ వ్యక్తీకరణ అమలు చేయబడుతుంది లేకపోతే మూడవ వ్యక్తీకరణ అమలు చేయబడుతుంది. **Operator: ? :** **Syntax: (exp 1 ? exp 2:exp 3)**

7. అసైన్మెంట్ ఆపరేటర్: ఈ ఆపరేటర్ ఒపెరాండ్కు విలువను కేటాయించడానికి

ఉపయోగించబడుతుంది **Operator: =** ఉదా: a = 5;



5. 'C'లో రిలేషనల్ ఆపరేటర్లు ఏమిటి? తగిన ఉదాహరణలతో వివరించండి.

What are the relational operators in 'C' ? Explain with suitable examples.

జ. రిలేషనల్ ఆపరేటర్లు: ఈ ఆపరేటర్ రెండు ఆపరాండ్లు లేదా రెండు వ్యక్తికరణల మధ్య

సంబంధాన్ని గుర్తించడానికి ఉపయోగిస్తారు. ఈ ఫలితాలు "ఒప్పు" లేదా "తప్పు". ఉదా: a= 5 మరియు b = 3.

| Operator | Meaning                  | Example | Result |
|----------|--------------------------|---------|--------|
| >        | Greater than             | a>b     | True   |
| <        | Less than                | a<b     | False  |
| >=       | Greater than or equal to | a>= b   | True   |
| <=       | Less than or equal to    | a<= b   | False  |
| ==       | Is equals to             | a==b    | False  |
| !=       | Not equals               | a!=b    | True   |

6. 'C' లో లాజికల్ ఆపరేటర్లు ఏమిటి? ఉదాహరణలతో వివరించండి.

What are the logical operators in C? Explain with examples.

జ. లాజికల్ ఆపరేటర్: లాజికల్ ఆపరేటర్లు రెండు వ్యక్తికరణల మధ్య ఉపయోగించబడతాయి

మరియు ఫలితం 'నిజం' లేదా 'తప్పు' రూపంలో అందించబడుతుంది.

- && (మరియు): రెండు ఒపెరాండ్లు నిజమైతే, లాజికల్ మరియు నిజం.
- || (లేదా) : ఒపెరాండ్ నిజం అయితే, లాజికల్ లేదా నిజం.
- ! (కాదు) : ఇది ఒపెరాండ్ను నిరాకరిస్తుంది.

ఉదా: a, b మరియు c పూర్ణాంక వేరియబుల్స్ మరియు వరుసగా 3, 5 మరియు 10 కేటాయించబడ్డాయి.

| Operator | Purpose | Expression     | Result |
|----------|---------|----------------|--------|
| &&       | And     | (a>b) && (c>a) | False  |
|          | Or      | (a>b)    (c>a) | True   |
| !        | Not     | !(a>8)         | True   |

7. ఇన్పుట్ మరియు అవుట్పుట్ స్టేట్మెంట్లు అంటే ఏమిటి? వాటిని వివరించండి.

What are input and output statements? Explain them.

జ. ఇన్పుట్ మరియు అవుట్పుట్ స్టేట్మెంట్లు:

|             | Input Statements  | Output Statements |
|-------------|-------------------|-------------------|
| Formatted   | scanf()           | printf()          |
| Unformatted | getchar(). gets() | putchar(), puts() |

**getchar():** C లైబ్రరీ ఫంక్షన్ `getchar()`ని ఉపయోగించి కంప్యూటర్లోకి ఒకే అక్షరాలను నమోదు చేయవచ్చు. ఇది ప్రామాణిక ఇన్పుట్ పరికరం నుండి ఒకే అక్షరాన్ని అందిస్తుంది. ఫంక్షన్కు ఎటువంటి వాదనలు అవసరం లేదు.

**Syntax:** <Character variable>= getchar();

**Example:** char c; c = getchar();

**putchar():** ఫంక్షన్ `putchar()`ని ఉపయోగించి ఒకే అక్షరాలు ప్రదర్శించబడతాయి. ఇది ప్రామాణిక అవుట్పుట్ పరికరానికి ఒకే అక్షరాన్ని అందిస్తుంది. ఇది ఫంక్షన్కు వాదనగా వ్యక్తీకరించబడాలి.

**Syntax:** putchar(<character variable>);

**Example:** char c; putchar(c);

**gets:** ఫంక్షన్ `gets()` ప్రామాణిక ఇన్పుట్ పరికరం నుండి స్ట్రింగ్ను అందుకుంటుంది.

**Syntax:** gets(<string type variable or array of char>);

**puts():** ఫంక్షన్ `Puts()` స్ట్రింగ్ను ప్రామాణిక అవుట్పుట్ పరికరానికి అవుట్పుట్ చేస్తుంది.

**Syntax:** puts(s); 's' అనేది `gets()`తో నిజమైన స్ట్రింగ్ ఎక్కడ ఉంది;

**Example:** main() {  
char line[80];  
gets(line); puts(line); }

**scanf():** `scanf()` ఫంక్షన్ ప్రామాణిక ఇన్పుట్ పరికరం నుండి డేటాను మెమరీలోకి ఇన్పుట్ చేయవచ్చు. ఈ ఫంక్షన్ సంఖ్యా విలువలు, ఒకే అక్షరాలు మరియు స్ట్రింగ్ల కలయికను నమోదు చేయడానికి ఉపయోగించవచ్చు. ఫంక్షన్ డేటా అంశాల సంఖ్యను అందిస్తుంది.

**Syntax:** -scanf ("control strings", &arg1,&arg2, &arg n);

నియంత్రణ స్ట్రింగ్ నిర్దిష్ట అవసరమైన ఫార్మాటింగ్ సమాచారాన్ని కలిగి ఉన్న స్ట్రింగ్ను సూచిస్తుంది మరియు arg1, arg2..arg n అనేది వ్యక్తిగత ఇన్పుట్ డేటా అంశాలను సూచించే ఆర్గ్యుమెంట్లు.

Example : `#include<stdio.h>`  
`main( ){`  
`char item[20]; int partno; float cost;`  
`scanf("%s %d %f",&item, &partno, &cost); }`

ఇక్కడ s, d, f, %తో మార్పిడి అక్షరాలు. మార్పిడి అక్షరాలు సంబంధిత డేటా రకాన్ని సూచిస్తాయి. డేటా ఇన్పుట్ నుండి సాధారణంగా ఉపయోగించే మార్పిడి అక్షరాలు.

### మార్పిడి అక్షరాలు(Conversion Characters):

| Characters | Meaning                            |
|------------|------------------------------------|
| %c         | డేటా అంశం ఒకే అక్షరం.              |
| %d         | డేటా అంశం దశాంశ పూర్ణాంకం.         |
| %f         | డేటా అంశం ఫ్లోటింగ్ పాయింట్ విలువ  |
| %e         | డేటా అంశం ఫ్లోటింగ్ పాయింట్ విలువ. |
| %g         | డేటా అంశం ఫ్లోటింగ్ పాయింట్ విలువ. |
| %h         | డేటా అంశం చిన్న పూర్ణాంకం.         |
| %s         | డేటా అంశం ఒక స్ట్రింగ్.            |
| %x         | డేటా అంశం హెక్సాడెసిమల్ పూర్ణాంకం. |
| %O         | డేటా అంశం ఆక్టల్ పూర్ణాంకం.        |

**printf( ):** కంప్యూటర్ మెమరీ నుండి డేటాను ప్రామాణిక అవుట్పుట్ పరికరంలో ప్రింట్ చేయడానికి printf() ఫంక్షన్ ఉపయోగించబడుతుంది. ఈ ఫంక్షన్ సంఖ్యా విలువలు, ఒకే అక్షరం మరియు స్ట్రింగ్ల కలయికను అవుట్పుట్ చేయడానికి ఉపయోగించవచ్చు.

**Syntax:** printf("control string", arg1, arg2.....argn);

**Example:** `#include<stdio.h>`  
`main() {`  
`char item[20];`  
`int partno; float cost;`  
`printf("%s %d %f", item, partno, cost):`  
`}` (ఇక్కడ %s %d %f మార్పిడి అక్షరాలు)

8. 'C'లో షరతులతో కూడిన ప్రకటనలు ఏమిటి? వాటిని వివరించండి.

What are the conditional statements in 'C'? Explain them.

జ. **Conditional statements:** షరతులతో కూడిన వ్యక్తీకరణలు ప్రధానంగా నిర్ణయం తీసుకోవడానికి ఉపయోగించబడతాయి. షరతులతో కూడిన ఆపరేషన్ల విధిని నిర్వహించడానికి క్రింది ప్రకటనలు ఉపయోగించబడతాయి.

1. if statement.
2. if-else statement.
3. nested else-if statement.
4. nested if-else statement.
5. switch statement.

1. **if statement:** షరతులతో కూడిన వ్యక్తీకరణలను వ్యక్తీకరించడానికి if స్టేట్‌మెంట్ ఉపయోగించబడుతుంది. ఇచ్చిన షరతు నిజమైతే, అది స్టేట్‌మెంట్‌లను అమలు చేస్తుంది లేకుంటే స్టేట్‌మెంట్‌లను దాటవేస్తుంది. మరియు తదుపరి స్టేట్‌మెంట్‌లతో అమలు కొనసాగుతుంది

'if' స్టేట్‌మెంట్ యొక్క సాధారణ నిర్మాణం `if (< conditional expression > statement-1;`

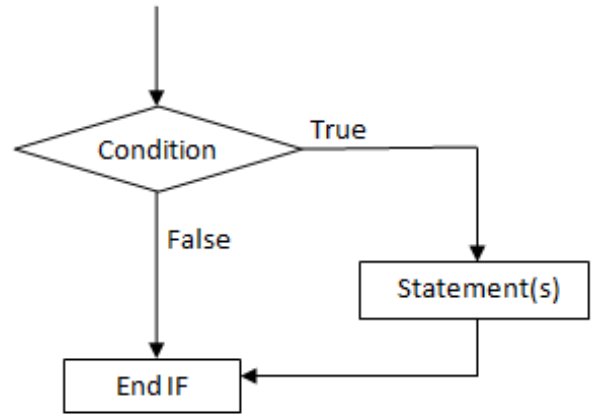


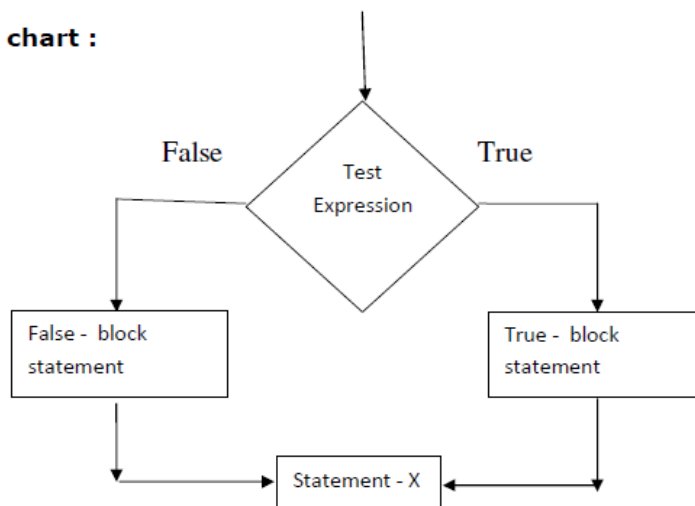
fig: Flowchart for if statement

**Example:** `a = 20; b = 10;`  
`if (a > b)`  
`printf ("big number is %d", a);`

**2. if-else statements:** షరతులతో కూడిన వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది మరియు వ్యక్తీకరణ నిజమైతే, నిజమైన బ్లాక్ స్టేట్‌మెంట్‌లు అమలు చేయబడతాయి. వ్యక్తీకరణ తప్పు అయితే, తప్పుడు బ్లాక్ స్టేట్‌మెంట్‌లు అమలు చేయబడతాయి.

**Syntax:** if (test expression) {  
     true block statements;  
 } Else {  
     false block statements; }

**flow chart :**

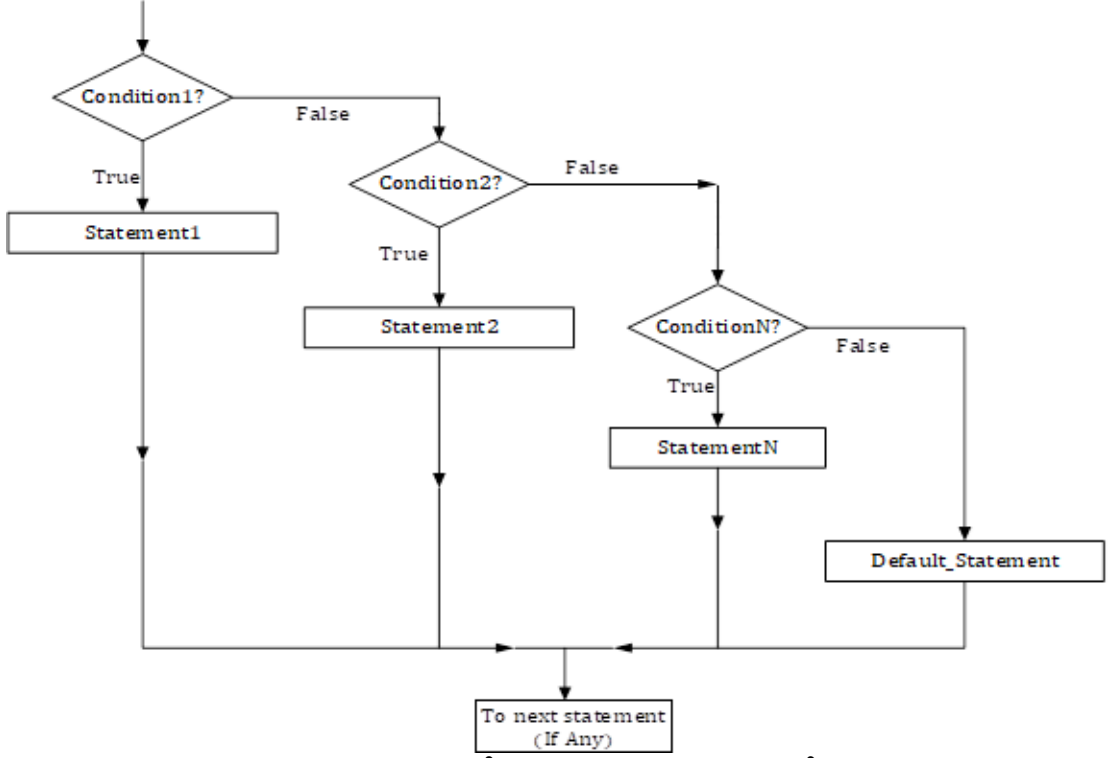


Example: if (a > b) {  
     printf ("a is greater than b"); }  
 else {  
     printf ("a is not greater than b"); }

**3. Else if ladder (or) nested else if:** కొన్ని పరిస్థితులలో, గూడు కట్టుకోవాలని కోరుకోవచ్చు బహుళ if-else స్టేట్‌మెంట్‌లు. ఈ పరిస్థితిలో అనేక విభిన్న చర్యలలో ఒకటి ఎంపిక చేయబడుతుంది. Else if ladder (or) nested else if

**Syntax:** if ( <expl> )  
     Statement-1;  
 else if ( <exp2> )  
     Statement-2,  
 else if ( <exp3> ) Statement-3;  
 else Statement-4:

లాజికల్ ఎక్స్ప్రెషన్ ని ఎదుర్కొన్నప్పుడు, దాని విలువ నిజం అయినప్పుడు సంబంధిత స్టేట్‌మెంట్‌లు ఎగ్జిక్యూట్ చేయబడతాయి మరియు మిగిలిన నెస్టెడ్ వేరే ఐఫ్ స్టేట్‌మెంట్ బైపాస్ చేయబడుతుంది. కాబట్టి నిజమైన పరిస్థితి ఎదురైన తర్వాత నియంత్రణ మొత్తం గూడు నుండి బదిలీ చేయబడుతుంది. వ్యక్తీకరణలో ఏదీ నిజం కానట్లయితే చివరి ఇతర నిబంధన వర్తించబడుతుంది.



**d. Nested if-else statement:** if-else స్టేట్‌మెంట్‌లను ఒకదానిలో ఒకటి గూడు

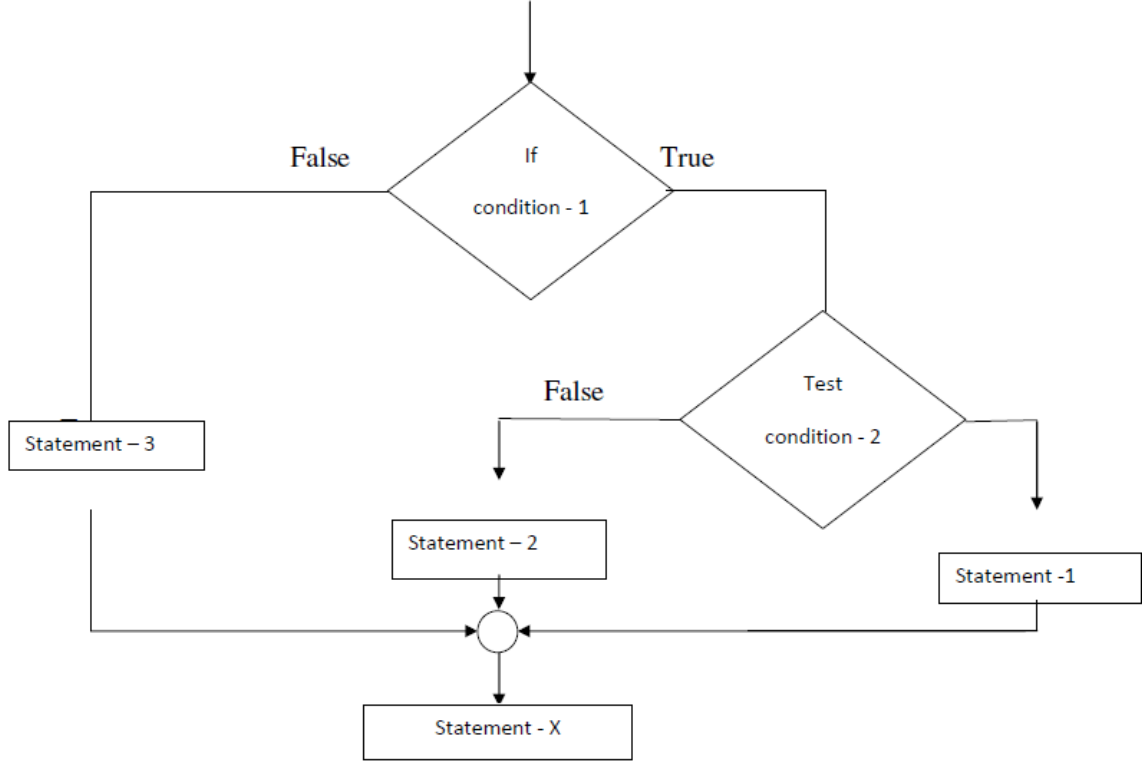
కట్టుకోవడం సాధ్యమవుతుంది. నెస్టెడ్ if-else స్టేట్‌మెంట్‌లు తీసుకోగల అనేక విధిన్న రూపాలు ఉన్నాయి.

రెండు-పొరల గూడు యొక్క అత్యంత సాధారణ రూపం

```

if(exp1)
    if(exp2)
        Statement-1;
    else
        Statement-2;
else
    if(exp3)
        Statement-3;
    else
        Statement-4;
  
```

వ్యక్తీకరణ1 ఒప్పు అయితే ఒక పూర్తి if-else స్టేట్‌మెంట్ అమలు చేయబడుతుంది మరియు ఎక్స్ ప్రెషన్లతప్పు అయితే మరొక పూర్తి if-else స్టేట్‌మెంట్ అమలు చేయబడుతుంది.



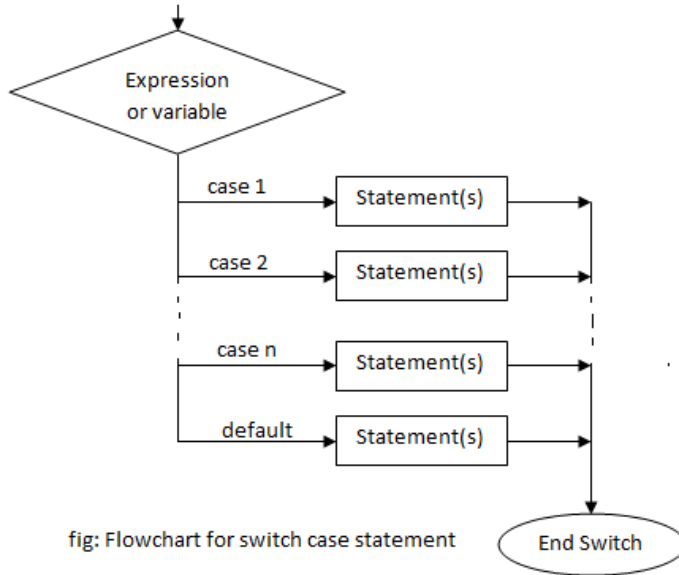
**e. Switch statement:** అనేక ప్రత్యామ్నాయాలలో స్టేట్‌మెంట్‌ను ఎంచుకోవడానికి స్విచ్ స్టేట్‌మెంట్ ఉపయోగించబడుతుంది.

**Syntax:**

```

switch (variable)
{
case label 1: statements1;
case label 2: statements2;
-----
default: statements n; }
  
```

ఇక్కడ లేబుల్ 1, లేబుల్ 2--- పూర్ణాంక స్థిరాంకాలు లేదా అక్షర స్థిరాంకాలు. స్విచ్ స్టేట్‌మెంట్ అమలు చేయబడినప్పుడు వేరియబుల్ మూల్యాంకనం చేయబడుతుంది మరియు నియంత్రణ నేరుగా స్టేట్‌మెంట్‌ల సమూహానికి బదిలీ చేయబడుతుంది, దీని కేస్ లేబుల్ విలువ విలువ యొక్క విలువతో సరిపోలుతుంది. కేస్ లేబుల్ విలువలు వేరియబుల్ విలువకు సరిపోలకపోతే, డిఫాల్ట్ పార్ట్ స్టేట్‌మెంట్‌లు అమలు చేయబడతాయి.



### 9. Break మరియు Continue statement ల మధ్య బేదాలను వ్రాయండి.

What are the differences between break-statement and continue statement?

| Break  | Continue  |
|--|---|
| <p>1. Break అనేది key word . దీనికి loop నుండి execution ను terminate చేయటానికి ఉపయోగిస్తారు.</p> <p>2. Break statment ను for, while, do-while loop, switch లలో ఉపయోగించవచ్చు Break వచ్చినప్పుడు inner loop నుండి వస్తుంది.</p> <p>3. Syntax: { Statment 1;<br/>Statement2;<br/>Statement n;<br/>break; }</p> <p>4. Switch(choice) {<br/>case 'y': Printf("Yes")<br/>break;<br/>case 'n': Printf("No")<br/>break; }</p> <p>5. ఇచ్చిన choice ఆధారంగా ఏదో ఒక statement మాత్రమే execute అవుతుంది.</p> | <p>1. Continue అనేది keyword. దీనికి loop continue చేసి తరువాత iteration తో execute చేస్తుంది.</p> <p>2. Continue statments ను for, do-while మరియు while statements లలో ఉపయోగిస్తారు. దీని ద్వారా loop ను terminate చేయకుండా next iteration తో loop ను continue చేస్తుంది.</p> <p>3. Syntax: { Statment 1;<br/>Continue ;<br/>Statment 2;<br/>Statment 3; }</p> <p>4. i=1,j=0; while(i&lt;=1) {<br/>i++;<br/>if(i==6)<br/>Continue ;<br/>j++; }</p> <p>5. పై ఉదాహరణతో i==6 అయినప్పుడు loop Continue అవుతుంది.</p> |



10. Write a C program biggest value of given three numbers.

æ. /\* Program to find biggest of three numbers \*/

```
#include <stdio.h>

int main() {
    double a,b,c;

    printf("Enter three different numbers: ");
    scanf("%lf %lf %lf", &a, &b, &c);
    if (a > b && a > c)
        printf("%.2f is the largest number.", a);
    if (b > a && b > c)
        printf("%.2f is the largest number.", b);
    if (c > a && c > b)
        printf("%.2f is the largest number.", c);
    return 0;
}
```

**Output:** Enter three different numbers: 10 8 30  
30.00 is the largest number

11. Write C program to find the given in vowel or alphabet character using switch statement.

æ. /\* find the given in vowel or alphabet charater using switch statement \*/

```
#include <stdio.h>

int main() {
    char ch; /* Input an alphabet from user */
    printf("Enter any alphabet: ");
    scanf("%c", &ch); /* Switch value of ch */
    switch(ch) {
        case 'a': printf("Vowel");
```

```
        break;
    case 'e': printf("Vowel");
        break;
    case 'i': printf("Vowel");
        break;
    case 'o': printf("Vowel");
        break;
    case 'u': printf("Vowel");
        break;
    case 'A': printf("Vowel");
        break;
    case 'E': printf("Vowel");
        break;
    case 'I': printf("Vowel");
        break;
    case 'O': printf("Vowel");
        break;
    case 'U': printf("Vowel");
        break;
    default:
        printf("Consonant");
} return 0; }
```

**Output:**

Enter any alphbet: i

i is vowel

Enter any alphbet: t

t is consonant

12. 'C' లో లూప్లు ఏమిటి? వాటిని వివరించండి.

What are the loops in C ? Explain them.

**జ. Looping statements:** వ్యక్తీకరణ నిజం అయినంత వరకు స్టేట్‌మెంట్‌లను పదేపదే అమలు చేయడానికి లూపింగ్ స్టేట్‌మెంట్‌లు ఉపయోగించబడతాయి. వ్యక్తీకరణ తప్పుగా మారినప్పుడు నియంత్రణ లూప్‌ను అనుసరించి వెంటనే స్టేట్‌మెంట్‌కు బదిలీ చేయబడుతుంది.

C లో మూడు రకాల లూప్లు ఉన్నాయి. అవి: a) while

b) do-while

c) for

**a) while loop:** పరీక్ష పరిస్థితి 'నిజం'గా ఉన్నంత వరకు స్టేట్‌మెంట్‌లు పదేపదే అమలు చేయబడతాయి. పరీక్ష పరిస్థితి 'తప్పు' అయితే, నియంత్రణ సమయంలో లూప్ నుండి బదిలీ చేయబడుతుంది.

**Syntax:** while (test condition) {  
statements;  
}

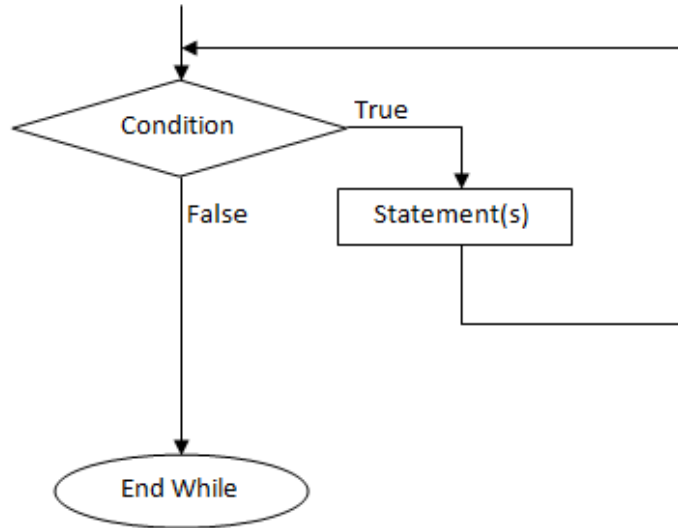


fig: Flowchart for while loop

**Example:**

```
int digit = 1;  
while (digit<=5) {  
printf("%d", digit);  
++digit;  
} The output is 1 2 3 4 5
```



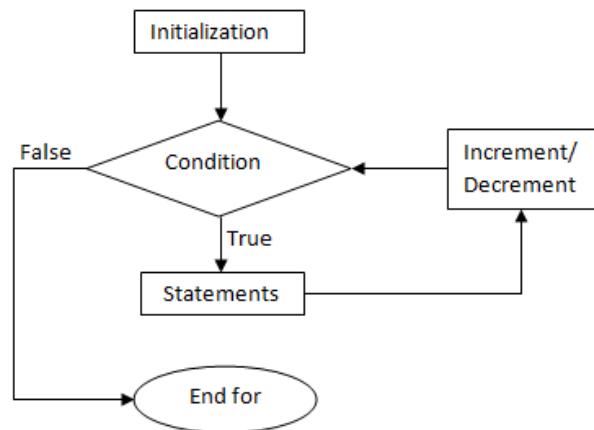


fig: Flowchart for for loop

**Example:**           for (i=1;i<=5;++i)  
                          printf ("%d",i);

The output is 1 2 3 4 5.

13. Write a C program sum of 1 to n natural number. Using while loop.

æ. /\* program to find sum of 1 to n natural number using while loop \*/

```

#include <stdio.h>

int main()
{
    int num, i, sum = 0; // declare local variables
    printf(" Enter a positive number: ");
    scanf("%d", &num); // take any positive number
    for (i = 0; i <= num; i++)
    {
        sum = sum + i;
    } printf("\n Sum of the first %d number is: %d", num, sum);
    return 0;
}
  
```

**Output:** Enter a positive number: 25

Sum of the first 25 number is: 325

14. Write C program to find the sum of 4 digit number using do... while loop.

æ. /\*program to find sum of the digits of the given number\*/

```
#include <stdio.h>
int main() {
    int n, sum = 0, m;
    printf ("Enter the integer number:\n");
    scanf ("%d", &n);
    printf ("The sum of %d digits is = ", n);
    do {
        m = n%10;
        sum = sum+m;
        n = n/10;
    }
    while (n > 0);
    printf ("%d\n", sum);
    return 0;
}
```

**Output:** Enter the integer number: 1234

The sum of 1234 digits is = 10

15. Write a C program whether given numbers is palindrome or not.

æ. /\*Program to find whether the given number is palindrome or not\*/

```
#include<stdio.h>
int main( ) {
    int n,m,s=0;
    printf("Enter the value :");
    scanf("%d",&n);
    m=n;
    while(n!=0) {
        s=s*10+n% 10;
```

```

        n=n/10;
    }
    if(s==m)
    { printf("%d is palindrome",m);
    }
    Else {
printf("%d is Not palindrome",m);
    }
    return 0;
}

```

**Output :** Enter the value: 353    353 is palindrome  
**Output :** Enter the value: 95    95 is Not palindrome

16. Write a C prog to find the factorial of given number using recursion method.

∞. /\*finds factorial by using recursion\*/

```

#include<stdio.h>
int fact(int);
int main() {
    int x,n;
    printf(" Enter the Number to Find Factorial :");
    scanf("%d",&n);
    x=fact(n);
    printf(" Factorial of %d is %d",n,x);
    return 0; }
int fact(int n) {
    if(n==0)
        return(1);
    return(n*fact(n-1));
}

```

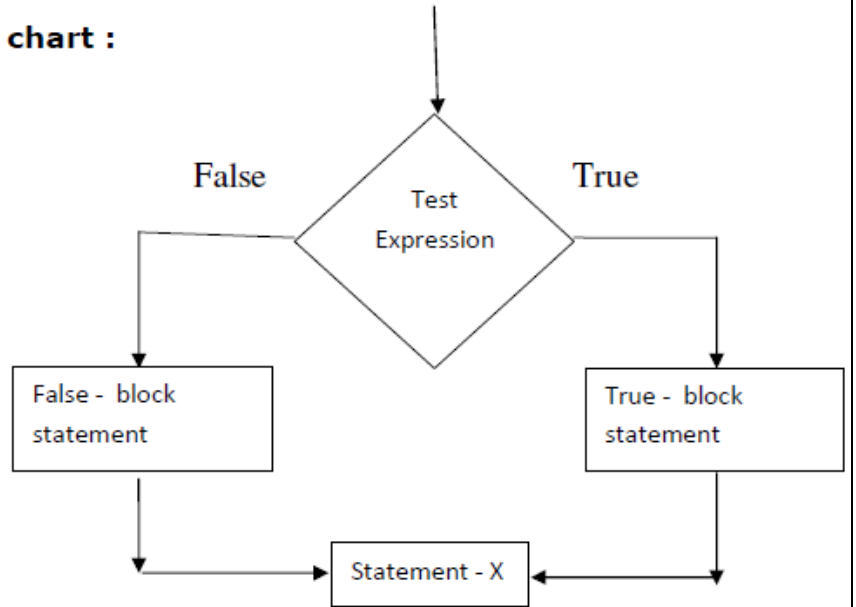
**Output:** Enter the Number to Find Factorial: 7  
 Factorial of 7 is 5040

17. C'లో if స్టేట్‌మెంట్‌లు గురించి వివరించండి Explain if else, nested if statements in 'C'.

జ. if..... else statement: if.....else స్టేట్‌మెంట్ అనేది సింపుల్ if స్టేట్‌మెంట్ యొక్క పొడిగింపు. పరీక్ష వ్యక్తీకరణ నిజమైతే, if కింద ఉన్న స్టేట్‌మెంట్‌లు అమలు చేయబడతాయి, else కింద ఉన్న స్టేట్‌మెంట్‌లు అమలు చేయబడతాయి. ఏదైనా సందర్భంలో, నిజమైన బ్లాక్ లేదా తప్పు\_బ్లాక్ రెండూ అమలు చేయబడవు.

**Syntax:** if (test expression) {  
true block statements;  
} else {  
false block statements; }

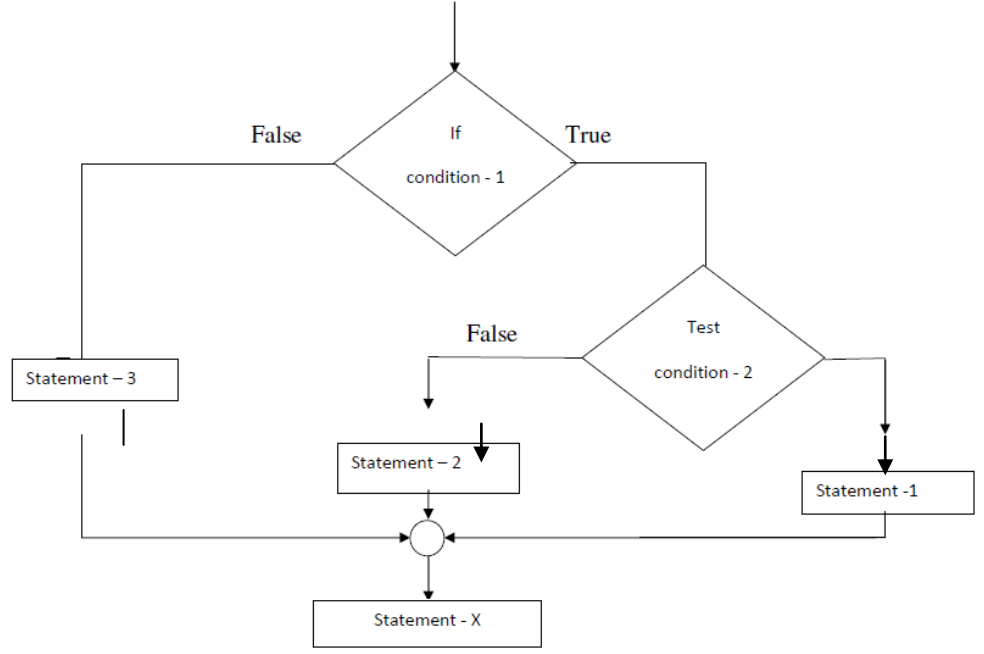
**flow chart :**



**Nested if statement:** Nested if స్టేట్‌మెంట్‌లు అంటే if స్టేట్‌మెంట్ లోపల మరొక if స్టేట్‌మెంట్.

**Syntax:** if (condition1)  
{  
// Executes when condition1 is true  
if (condition2)  
{  
// Executes when condition2 is true  
}  
}





18.if స్టేట్‌మెంట్ గురించి ఉంటే ఉదాహరణతో వివరించండి.

Explain if else statement with example.

జ. If-else statements: పరతులతో కూడిన వ్యక్తీకరణ మూల్యాంకనం చేయబడుతుంది మరియు వ్యక్తీకరణ నిజమైతే స్టేట్‌మెంట్‌లు-1 అమలు చేయబడుతుంది. వ్యక్తీకరణ తప్పు అయితే స్టేట్‌మెంట్‌లు 2 అమలు చేయబడతాయి.

**Syntax:** if ( conditional expression>  
                   statements-1  
                   else  
                   statements-2;

**Example:**

```

#include<stdio.h>
int main() {
    int num=19;
    if(num<10)
    {
        printf("The value is less than 10");
    }
    Else
  
```

```

{
    printf("The value is greater than 10");
}
return 0;
}

```

**Output:** The value is greater than 10

## 19. While మరియు and do while ల మధ్య తేడా తెల్పండి

**What are the difference between while and do while statement?**

| While   | Do While  |
|---|---|
| 1) ఇది ముందుగా పరిస్థితిని తనిఖీ చేసి, ఆపై స్టేట్‌మెంట్ (ల)ని అమలు చేస్తుంది                      | 1) ఈ లూప్ స్టేట్‌మెంట్ (ల)ని కనీసం ఒక్కసారైనా అమలు చేస్తుంది, తర్వాత పరిస్థితి తనిఖీ చేయబడుతుంది.                       |
| 2) లూప్ లూప్ బాడీని ప్రారంభించడానికి ముందు కౌంటర్ వేరియబుల్‌స్‌ను ప్రారంభించడాన్ని అనుమతిస్తుంది. | 2) లూప్ బాడీని ప్రారంభించడానికి ముందు మరియు తర్వాత కౌంటర్ వేరియబుల్‌స్‌ను ప్రారంభించడాన్ని డూ అయితే లూప్ అనుమతిస్తుంది. |
| 3) ఇది ఎంట్రీ కంట్రోల్ లూప్.  | 3) ఇది ఎగ్జిట్ కంట్రోల్ లూప్.   |
| 4) కాసేపు కండిషన్ ముగింపులో మనం సెమికోలన్‌ని జోడించాల్సిన అవసరం లేదు.                             | 4) అయితే కండిషన్ చివరిలో మనం సెమికోలన్‌ను జోడించాలి.  |
| 5) ఒకే స్టేట్‌మెంట్ విషయంలో, మనము బ్రాకెట్‌లను జోడించాలి.   | 5) బ్రాకెట్లు ఎల్లప్పుడూ అవసరం.   |
| 6) ఈ లూప్‌లో, లూప్ ప్రారంభంలో పరిస్థితి పేర్కొనబడింది.  | 6) బ్లాక్ అమలు చేయబడిన తర్వాత లూప్ పరిస్థితి పేర్కొనబడుతుంది.   |
| 7) షరతు తప్పు అయితే స్టేట్‌మెంట్ (లు) సున్నా సార్లు అమలు చేయబడతాయి.                               | 7) ప్రకటన కనీసం ఒక్కసారైనా అమలు చేయబడుతుంది.  |
| 8) while (condition) {<br>Statements; // loop body<br>}   | 8) do {<br>Statements; // loop body<br>} while (condition);   |

## 20. Explain the nested loop with example.

జ. **Nested loop:** లూప్ లోని లూప్ రాసినట్లయితే దానిని nested loop అని అంటారు.

loopలో (while,do..while,for)లోపలి మరియు బయటి లూప్లు ఒకే రకంగా ఉండవచ్చు లేదా ఒకే రకంగా ఉండకపోవచ్చు.

**Syntax:**            Loop{  
                              Loop{  
                                      Statement block  
                                      }/\* end of second loop \*/  
                              }/\* end of first loop \*/

**Example:**            for (i=0;i≤5;i++) {  
                              for (j=0; j ≤ m; j++) {  
                                      scanf("%d", a(i) (j));  
                                      }  
                              }

## 21. Write a C program to find factorial value of a given numbers.

జ. /\* To find factorial value of a given numbers Program \*/

```
#include<stdio.h>
int main() {
    int i,f=1,num;
    printf("Enter a number: ");
    scanf("%d",&num);
    for(i=1;i<=num;i++)
        f=f*i;
    printf("Factorial of %d is: %d", num,f);
    return 0;
}
```

**Output:** Enter a number n: 6

Factorial of 6 is 720

22. Write a C program to find Fibonacci series up to given numbers.

æ. /\*program to print fibonacci series\*/

```
#include<stdio.h>
void main()
{
    int i=1,n,f,f1,f2;
    printf("Enter Number of Fibonacci Values Needed : ");
    scanf("%d",&n);
    f=0;
    f1=1;
    f2=1;
    do {
        i++;
        printf("%d\n",f);
        f1=f2;
        f2=f;
        f=f1+f2;
    } while(i<=n);
}      Output: Enter how many numbers to be print: 10
```

1 1 2 3 5 8 13 21 34 55

23. Write a C program to find the sum of 4 digits number.

æ. /\*program to find sum of the digits of the given number\*/

```
#include <stdio.h>
int main() {
    int n, sum = 0, m;
    printf ("Enter the integer number::\n");
    scanf ("%d", &n);
    printf ("The sum of %d digits is = ", n);
    while (n > 0) {
```

```

        m = n%10;
        sum = sum+m;
        n = n/10;    }
        printf ("%d\n", sum);
    return 0;
}

```

**Output:** Enter the integer number: 1234

The sum of 1234 digits is = 10

**24. Write a C program to find smallest of given three numbers.**

∞. /\* program to find smallest of given three numbers \*/

```

#include<stdio.h>
#include<conio.h>
int main() {
    int a, b, c, small;
    printf("Enter three numbers : ");
    scanf("%d%d%d", &a, &b, &c);
    if(a<b && a<c)
        small = a;
    else if(b<a && b<c)
        small = b;
    else
        small = c;
    printf("\nSmallest number is: %d", small);
    getch();
    return 0;
}

```

**Output:** Enter three numbers: 18 26 35

Smallest number is: 18

---

## UNIT-III

### Array in 'C'

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Array అనగా నేమి? What is Array?

జ. **Array:** ఒకే రకమైన మూలకాల (data types చెందిన data items) సమితిని array అంటారు (లేదా) array అనునది కొన్ని objects యొక్క కలయిక. ఈ objects అన్ని memory లో ఒక ప్రక్క ప్రక్క locationలో store చేయబడతాయి. ఈ objects అన్ని ఒకే పేరులో పిలువబడుతాయి. Arrayలో ఉన్న విలువలను elements అంటారు Arrayలో ఉన్న elementsను access చేయటానికి array indexను ఉపయోగిస్తారు Array index అనునది సున్న నుండి మొదలవుతుంది.

#### 2. Array ను నిర్వచించండి. Definition of Array?

జ. **Array:** సాధారణ శ్రేణి పేరుతో వరుస మెమరీ స్థానాల్లో నిల్వ చేయబడిన డేటా అంశాల సేకరణగా అర్థేని నిర్వచించవచ్చు.

#### 3. Arrays లో ఉన్న రకాలను వివరించుము? Write the types of Arrays.

జ. Arrays ముఖ్యంగా మూడు రకాలు. అవి

1. one dimensional array (Single dimensional array)

Syntax: Data \_ type array \_ name [size];

Ex: int a [10];

2. Two diemensinol array

Syntax: Data \_ type array \_ name [row-size] [columns-size];

Ex:-int a[10] [10];

3. Multi dimensional array

Syntax: Data \_ type array \_ name [size] [row-size]

[columns-size];

Ex:-int a[10] [10] [10];

4. Single Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి.

Write the Syntax with example for Single Dimensional Array.

జ. one dimensional array (Single dimensional array)

**Syntax:** Data \_type array \_name [size];

Ex: int a [10];  
int stno [20];  
float streight [10];  
char stname [10];

5. Two Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి.

Write the Syntax with Example for Two Dimensional Array.

జ. Two diemensinol array

**Syntax:** Data \_type array \_name [row-size] [columns-size];

Ex:-int a[10] [10];

6. Multi Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి

Write the Syntax with Example for Multi Dimensional Array.

జ. Multi Dimensional Array:

**Syntax:** Data \_type array \_name [size] [row-size]  
[columns-size];

Ex:-int a[10] [10] [10];

ధీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

1. Write a 'C' Program to sort a given array of elements.

జ. /\* Program to sort a given array of elements \*/

```
#include <stdio.h>
int main( )
{
    int i, j, a, n, number[30];
    printf("Enter the value of N \n");
    scanf("%d", &n);
    printf("Enter the numbers \n");
```

```

    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);
    for (i = 0; i < n; ++i)    {
        for (j = i + 1; j < n; ++j)    {
            if (number[i] > number[j])
                {
                    a = number[i];
                    number[i] = number[j];
                    number[j] = a;
                }    }    }

printf("The numbers arranged in ascending order are given below \n");
for (i = 0; i < n; ++i)
    printf("%d\n", number[i]);
return 0;
}

```

**Output:** How many numbers you want to sort: 5

Enter 5 elemets; 12 18 3 16 4

Elements after sorting are: 3 4 12 16 18

**2. Write a 'C' Program to perform matrix addition.**

∞. **/\*Program for Addition of TwoMatrices\*/**

```

#include <stdio.h>

int main() {
    int r, c, a[5][5], b[5][5], sum[5][5], i, j;
    printf("Enter no of rows and columns for first matrix:");
    scanf("%d", &r);
    printf("Enter no of rows and columns for second matrix:");
    scanf("%d", &c);
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)

```



```

    for (j = 0; j < c; ++j) {
        printf("Enter element a%d%d: ", i + 1, j + 1);
        scanf("%d", &a[i][j]);
    }
printf("Enter elements of 2nd matrix:\n");
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("Enter element b%d%d: ", i + 1, j + 1);
        scanf("%d", &b[i][j]);
    } // adding two matrices
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        sum[i][j] = a[i][j] + b[i][j];
    } // printing the result
printf("\nSum of two matrices: \n");
for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("%d ", sum[i][j]);
        if (j == c - 1) {
            printf("\n\n");
        }
    } return 0;
}

```

**Output:** Enter no of rows and columns for first matrix:3

Enter no of rows and columns for second matrix:3

Enter elements of 1st matrix:Enter element a11: 1

Enter element a12: 2

Enter element a13: 3

Enter element a21: 4

Enter element a22: 5

Enter element a23: 6

Enter element a31: 7

Enter element a32: 8

Enter element a33: 9

Enter elements of 2nd matrix:

Enter element b11: 1

Enter element b12: 2

Enter element b13: 3

Enter element b21: 4

Enter element b22: 5

Enter element b23: 6

Enter element b31: 7

Enter element b32: 8

Enter element b33: 9

**Sum of two matrices:**

2 4 5

8 10 12

14 16 18

**3. Write a 'C' Program to perform matrix multiplication.**

æ. **/\*program to perform matrix multiplication\*/**

```
#include<stdio.h>
#include<stdlib.h>
int main() {
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
```

```

printf("enter the first matrix element=\n");
for(i=0;i<r;i++) {
for(j=0;j<c;j++) {
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
} }
printf("multiply of the matrix=\n");
for(i=0;i<r;i++) {
for(j=0;j<c;j++) {
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k] *b[k][j];
} }
} //for printing result
for(i=0;i<r;i++) {
for(j=0;j<c;j++) {
printf("%d\t",mul[i][j]);
}
printf("\n");
} return 0;
}

```

### Output :

enter the number of row=2

enter the number of column=2

enter the first matrix element=1 2 3 4

enter the second matrix element=5 6 7 8

multiply of the matrix= 19 22

43 50

### Additional Programs:

4. Write a 'C Creating a single dimensional array of numbers and displaying them

æ /\*Creating a single dimensional array of numbers and displaying them\*/

```
#include<stdio.h>
int main()
{
    int i,j,n,a[10];
    printf("\n How many elements:");
    scanf("%d",&n);
    printf("\n Enter %d values : ",n);
    for(i=0;i<n;++i)
        scanf("%d",&a[i]);
    printf("\n Given array\n");
    for(j=0;j<n;++j)
        printf("\t%d",a[j]);
    return 0;
}
```

**Output:** How many elements:5

Enter 5 values:

25 36 14 89 71

Given array

25 36 14 89 71

5. Write a 'C' program to find the sum and average of given array of elements.

∞. 'C' program to find the sum and average of given array of elements.

```
#include<stdio.h>
int main() {
    int a[25],n,i;
    float avg=0,sum=0;
    printf("\n\nEnter the Numbers of element in Array: ");
    scanf("%d",&n);
    printf("\n\nEnter the Element of Array : \n");
    for(i=1;i<=n;i++)
        scanf("%d",&a[i]);
    for(i=1;i<=n;i++)
        sum=sum+a[i];
    avg=sum/n;
    printf("\nSum of Element of Array is : %f",sum);
    printf("\nAverage of Element of Array are : %f",avg);
    return 0; }
```

**Output:**

Enter the Numbers of element in Array: 6

Enter the Element of Array : 12 65 23 58 45 12

Sum of Element of Array is : 215.000000

Average of Element of Array are : 35.833332

6. Write a 'C Program to display the elements of an array in reverse order

∞. /\* Program to display the elements of an array in reverse order \*/

```
#include<stdio.h>
int main() {
    int i,j,n,a[10];
    printf("\n How many elements: ");
    scanf("%d",&n);
```

```

printf("Enter the %d elements: ",n);
for(i=0;i<n;++i)
scanf("%d",&a[i]);
printf("\n Given array\n");
for(i=0;i<n;++i)
printf("\t%d",a[i]);
printf("\n Reverse of array\n");
for(j=n-1;j>=0;--j)
printf("\t%d",a[j]);
return (0);
}

```

**Output:**

How many elements: 4

Enter the 4 elements: 15 22 63 14

Given array 15 22 63 14

Reverse of array 14 63 22 15

**7. Write a 'C Program to find the biggest element in an array with location**

∞./\* Program to find the biggest element in an array with location\*/

```

#include <stdio.h>
int main() {
int array[100], maximum, size, c, location = 1;
// clrscr();
printf("Enter the number of elements in an array:");
scanf("%d", &size);
printf("Enter %d integers\n", size);
for (c=0; c < size; c++)
scanf("%d", &array[c]);
printf("\n Given array is \n");

```

```

for(c=0;c<size; ++c)
printf("\t%d", array[c]);
maximum = array[0];
for (c = 1; c < size; c++) {
if (array[c]> maximum) {
maximum = array[c];
location = c+1;
} }
printf("\nMaximum element is present at location number %d
and it's value is %d.\n", location, maximum);
return (0);
}

```

**Output:** Enter the number of elements in an array:5

Enter 5 integers 12      36      14      98      72

Given array is    12      36      14      98      72

Maximum element is present at location number 4 and it's value is 98.

**8. Write a 'C Program to find the length of the character array**

∞./\* Program to find the length of the character array \*/

```

#include<stdio.h>
#include<string.h>
main() {
char a[100];
int length;
printf("Enter a string to calculate it's length:\n");
gets(a);
length = strlen(a);
printf("Length of entered string is = %d\n", length);
// getch( ); }

```

**Output:** Enter a string to calculate it's length: Sankarraju

Length of entered string is = 10

### 9. Write a 'C Program to print transpose of a matrix.

∞. /\* Program to print transpose of a matrix. \*/

∞. This c program prints transpose of a matrix. It is obtained by interchanging rows and columns of a matrix. For example, if a matrix is

1 2

3 4

5 6

then transpose of above matrix will be

1 3 5

2 4 6

```
#include <stdio.h>
int main() {
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    printf("\nEnter matrix elements:\n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    // printing the matrix a[][]
    printf("\nEnter matrix: \n");
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j) {
            printf("%d ", a[i][j]);
            if (j == c - 1)
```



```

        printf("\n");
    } // computing the transpose
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
        {
            transpose[j][i] = a[i][j];
        } // printing the transpose
    printf("\nTranspose of the matrix:\n");
    for (int i = 0; i < c; ++i)
        for (int j = 0; j < r; ++j) {
            printf("%d ", transpose[i][j]);
            if (j == r-1)
                printf("\n"); }
    return 0;
}

```

**Output:** Enter rows and columns: 2 3

Enter matrix elements:

Enter element a11: 1

Enter element a12: 2

Enter element a13: 3

Enter element a21: 4

Enter element a22: 5

Enter element a23: 6

Entered matrix:    1 2 3  
                   4 5 6

Transpose of the matrix:    1 4  
                                   2 5  
                                   3 6

## UNIT-IV

### Functions

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Procedure అనగా నేమి? What is a procedure?

జ. **Procedure:** సాధారణంగా ఉపయోగించే ప్రోగ్రామ్ మాడ్యూల్‌లను విధానాలు అంటారు.

మీరు పనిని చేయాలనుకున్నప్పుడు మీరు ఒక విధానాన్ని (కాల్) చేయవచ్చు.

#### 2. Function అనగానేమి? What is function? Define a function.

జ. **Function:** function అనునది ఒక subprogram వీనిని main programలో ఒక ప్రత్యేకమైన పనిని చేయటానికి ఉపయోగించడానికి programలో ఎక్కవసార్లు ఉపయోగించే code లను మరల ఎగ్జిక్యూట్ చేయాలంటే function చక్కగా ఉపయోగపడుతుంది. functionలో function పేరు. ఒక జత బ్రాకెట్లు ( pair of parentheses' parameters మరియు function body ఉంటాయి

#### 3. వివిధ రకాల functionలను తెలుపుము? What are the types of function?

జ. ఫంక్షన్ (function) రకాలు: విధులు రెండు రకాలు

ఎ. వినియోగదారు నిర్వచించిన విధులు (User defined functions)

బి. లైబ్రరీ విధులు (Library functions)

#### 4. Function యొక్క syntax అంటే ఏమిటి? ఒక ఉదాహరణ ఇవ్వండి.

What is the syntax of function? Give an example.

జ. ఫంక్షన్ యొక్క సాధారణ రూపం క్రింది విధంగా ఉంటుంది:

```
return_type function_name(parameter test) { Body }
```

సి ప్రోగ్రామింగ్ లాంగ్వేజ్‌లో ఫంక్షన్ డెఫినిషన్‌లో ఫంక్షన్ హెడర్ మరియు ఫంక్షన్ బాడీ ఉంటాయి.

#### 5. ఫంక్షన్ యొక్క ప్రయోజనాలు ఏమిటి? What are the advantages of function?

జ . ఫంక్షన్‌ని ఉపయోగించడం వల్ల కలిగే ప్రధాన ప్రయోజనాలు:

- సరైన చిన్న ఫంక్షన్ రాయడం సులభం.
- ఫంక్షన్‌ను చదవడం మరియు డిబగ్ చేయడం సులభం.
- అటువంటి ఫంక్షన్‌ను నిర్వహించడం లేదా సవరించడం సులభం. చిన్న ఫంక్షన్లు స్వీయ డాక్యుమెంట్‌గా మరియు బాగా చదవగలిగేవిగా ఉంటాయి.
- వివిధ పారామితులతో ఏ ప్రదేశంలోనైనా దీన్ని ఎన్నిసార్లు అయినా పిలవవచ్చు.

## 6. Global variable అనగా నేమి? What is a global variable?

జ. **Global variable:** ఒక program మొత్తం చరరాశులుగా (variables) ఉపయోగించే వాటిని global variables అంటారు. global variables ను main( ) function ముందు వ్రాస్తారు. వీటిని program లోని e function నుండి అయినా call చేసుకోవచ్చును.

## 7. Local variable అనగా నేమి? What is a local variable?

జ. **లోకల్ వేరియబుల్స్:** లోకల్ వేరియబుల్ అనేది ఫంక్షన్ లోపల డిక్లెర్ చేయబడిన వేరియబుల్. స్థానిక వేరియబుల్ డిక్లెర్ చేయబడిన ఫంక్షన్ లో మాత్రమే ఉపయోగించబడుతుంది.

## 8. Parameters అంటే ఏమిటి? what are the parameters?

జ. **Parameters:** పరామితి ఒక ఫ్లెన్ హోల్డర్ లాంటిది. ఒక ఫంక్షన్ ప్రారంభించబడినప్పుడు, మనము పారామీటర్ కి విలువను పాస్ చేస్తాము. ఈ విలువ వాస్తవ పారామీటర్ లేదా వాదనగా సూచించబడుతుంది పారామీటర్ జాబితా ఒక ఫంక్షన్ యొక్క పారామీటర్ ల రకం, క్రమం మరియు సంఖ్యను సూచిస్తుంది. పారామితులు ఐచ్ఛికం; అంటే, ఒక ఫంక్షన్ లో పారామీటర్లు ఉండకపోవచ్చు.

## 9. Main ( ) function అంటే ఏమిటి? What is a Main ( ) function?

జ. **మెయిన్ ( ) ఫంక్షన్:** ప్రతి 'సి' ప్రోగ్రామ్ లో ఒక మెయిన్ ( ) ఫంక్షన్, మెయిన్ ( ) ప్రోగ్రామ్ మాత్రమే ఉంటుంది, ప్రోగ్రామ్ ఎక్కడ మొదలవుతుందో కంపైలర్ కు తెలియజేస్తుంది.

## 10. ఏదైనా 4 గణిత విధులను ఉదాహరణతో వ్రాయండి.

Write any 4 mathematical functions with example.(Or)

What any 4 arithmetical function give example.

జ. ప్రామాణిక గణిత విధులు నిర్వచించబడ్డాయి మరియు C గణిత లైబ్రరీలో భాగంగా ఉంచబడతాయి. మేము ఈ గణిత ఫంక్షన్లలో దేనినైనా ఉపయోగించాలనుకుంటే, math.h హెడర్ ఫైల్ ని చేర్చడం ద్వారా మనం తప్పనిసరిగా #include సూచనను జోడించాలి.

| Function    | Meaning          |
|-------------|------------------|
| 1) acos(X)  | Arc cosine of X  |
| 2) asin(X)  | Arc sine of X.   |
| 3) atanx(X) | Arc tangent of X |
| 4) cos(X)   | Cosine of X      |
| 5) sin (X)  | Sine of X        |
| 6) tan(X)   | Tangent of X.    |

**12. Global string అనగా నేమి? What is a global string?**

**జ. Global string:** గ్లోబల్ వేరియబుల్ (ఫంక్షన్ ప్రాంతం వెలుపల)గా ప్రకటించబడిన అక్షర శ్రేణిని గ్లోబల్ స్ట్రింగ్ అంటారు.

**13. Recursion అనగా నేమి? What is a recursion?**

**జ. Recursion:** ఒక function అదే functionను మరల మరల call చేయటాన్ని recursion అంటారు. దీనిని directగా గాని, indirect కాని call చేయవచ్చును. దీనిలో ఉపయోగించి condition తృప్తిపరిచే వరకు function తిరిగి తిరి call అవుతుంది. ఈ పద్ధతిని ఉపయోగించి program వ్రాస్తే ఎగ్జిక్యూషన్ కొంచెం ఆలస్యం అవుతుంది.

**14. Return statement ను ఉపయోగం వ్రాయండి. What is use of return statement?**

**జ. Return statement:** సమాచారం ఫంక్షన్ నుండి రిటర్న్ స్టేట్ మెంట్ ద్వారా అందించబడుతుంది.

**15. void function యొక్క ఉపయోగం ఏమిటి? What is use of void function?**

**జ. Void function:** శూన్యమైన ఫంక్షన్ కాల్ చేసిన ఫంక్షన్ కు ఏ విలువను అందించదు.

**16. 'C' లో స్ట్రింగ్ హ్యాండిల్లింగ్ ఫంక్షన్ ఏమిటి? వాటిని వివరించండి.**

**What are the string handling function in 'c'? Explian them.**

**జ. string handling function:** స్ట్రింగ్ లతో పని చేయడానికి వివిధ స్ట్రింగ్ హ్యాండిల్లింగ్ ఫంక్షన్ లను కలిగి ఉన్న 'C' భాష. ఈ విధులన్నీ హెడర్ ఫైల్ string.h తో అందుబాటులో ఉన్నాయి. 1. strlen() 2. strcat() 3. strcpy() 4. strcmp()

ధీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

**1. Function మరియు Procedureల మధ్య తేడాలను వివరించుము?**

**What are the differences between procedure and function?**

| Procedure  | Function  |
|--|---|
| 1. procedure అనునది ఒక sub program. ఇది main programe లోనే include అయి ఉంటుంది | 1. Function అనునది ఒక ప్రత్యేకమైన పని కోసం వ్రాయబడే program.          |
| 2. ఇది ఎటువంటి విలువలను main program కు return చేయము.                          | 2. Function నుండి main program values ని return చేయవచ్చు చేయక పోవచ్చు |

|   |   |
|---|---|
| 3. దీనిని తిరిగి తిరిగి call చేయలేము  | 3. ఒక్కసారి function ని వ్రాసి తరవాత ఎన్ని సార్లు అయినా call చేసుకోవచ్చు. |
| 4. దీనిలో global variablesను ఉపయోగించుకోలేము.   | 4. functionలో local variable మరియు global విలువలను ఉపయోగించుకోవచ్చు.      |
| 5. దీనిని కేవలం procedural languages అయిన foxpro, dbase వంటి వాటిలో మాత్రమే వ్రాయగలము | 5. దీనిని modular programmingలు అయిన c, c++ వంటి languageలో వ్రాయగలము.    |

## 2. Library functions ఏమిటి? తగిన ఉదాహరణలతో వివరించండి.

**What are library functions? Explain with suitable examples.**

**జ. లైబ్రరీ విధులు:** లైబ్రరీ విధులు ఇప్పటికే సిస్టమ్ లైబ్రరీలో నిర్వచించబడిన విధులు. ప్రోగ్రామర్ తన C ప్రోగ్రామ్లో సంబంధిత హెడర్ ఫైల్ను దిగుమతి చేయడం ద్వారా కార్యాచరణను సాధించవచ్చు. printf(), scanf(), sqrt(), cos(), strcmp(), strcat(), strlen() etc., లైబ్రరీ ఫంక్షన్ల వర్గానికి చెందినవి.

1. **sqrt(double x):** C లైబ్రరీ ఫంక్షన్ డబుల్ sqrt(డబుల్ x) అందిస్తుంది. x యొక్క వర్గమూలం.

2. **pow(double x, double y):** C లైబ్రరీ ఫంక్షన్ డబుల్ pow(double x, double y) xని y యొక్క శక్తికి పెంచడం. అంటే, xy.

3. **printf():** కంప్యూటర్ మెమరీ నుండి డేటాను ప్రామాణిక అవుట్పుట్ పరికరంలో ప్రింట్ చేయడానికి printf() ఫంక్షన్ ఉపయోగించబడుతుంది.

4. **scanf():** ఈ ఫంక్షన్ ప్రామాణిక ఇన్పుట్ పరికరం నుండి డేటాను మెమరీలోకి ఇన్పుట్ చేయవచ్చు.

5. **strlen():** ఇచ్చిన స్ట్రింగ్ యొక్క పొడవును కనుగొనడానికి ఈ ఫంక్షన్ ఉపయోగించబడుతుంది. ఈ ఫంక్షన్కు ఇన్పుట్ స్ట్రింగ్ మరియు ఈ ఫంక్షన్లు ఇచ్చిన స్ట్రింగ్ యొక్క పొడవుగా పూర్ణాంకం విలువను అందిస్తుంది.

ఉదా: length= strlen("Raju"); ఎగువ స్టేట్మెంట్ పొడవుగా పూర్ణాంకం విలువ 4ని అందిస్తుంది.

6. **strcat():** ఈ ఫంక్షన్ రెండు స్ట్రింగ్లను కలపడానికి ఉపయోగించబడుతుంది. రెండవ స్ట్రింగ్ మొదటి స్ట్రింగ్ చివరిలో జతచేయబడుతుంది. ఈ ఫంక్షన్కి ఇన్పుట్ రెండు స్ట్రింగ్లు. ఈ ఫంక్షన్ తుది ఫలితం వలె సంయోగం చేయబడిన స్ట్రింగ్ను అందిస్తుంది.

ఉదా: strcat("onestring", "secondstring");

7. **strcmp()**: ఈ ఫంక్షన్ ఇచ్చిన రెండు స్ట్రింగ్లు సమానంగా ఉన్నాయో లేదో తెలుసుకోవడానికి ఉపయోగించబడుతుంది. రెండు స్ట్రింగ్లు సమానంగా లేకుంటే, ఈ ఫంక్షన్ తప్పుని అందిస్తుంది, లేకపోతే ట్రూ రిటర్న్ చేస్తుంది. ఉదా: `strcmp("stringone", "string2");`

8. **strcpy()**: ఇది మరొక స్ట్రింగ్ నుండి స్ట్రింగ్ను కాపీ చేయడానికి ఉపయోగించబడుతుంది.

3. **Mathematical functions ఏమిటి? తగిన ఉదాహరణతో వివరించండి.**

**What are the mathematical functions? Explain with suitable example.**

జ. **math.h** హెడర్ వివిధ గణిత విధులను మరియు ఒక స్థూలాన్ని నిర్వచిస్తుంది. ఈ లైబ్రరీలో అందుబాటులో ఉన్న అన్ని ఫంక్షన్లు ఆర్గ్యుమెంట్గా రెట్టింపును తీసుకుంటాయి మరియు ఫలితంగా రెట్టింపును అందిస్తాయి. కొన్ని ముఖ్యమైన విధులను ఒక్కొక్కటిగా చర్చించిద్దాం.

1. **double cell(double x):** C లైబ్రరీ ఫంక్షన్ **double cell (double x)**

x కంటే ఎక్కువ లేదా దానికి సమానమైన చిన్న పూర్ణాంక విలువను అందిస్తుంది.

**Syntax: double cell(double x)**

2. **double floor(double x):** C లైబ్రరీ ఫంక్షన్ **double floor (double x)**

అతిపెద్ద పూర్ణాంక విలువను x కంటే తక్కువ లేదా సమానంగా అందిస్తుంది.

**Syntax: double floor(double x)**

3. **double fabs(double x):** C లైబ్రరీ ఫంక్షన్ **double fabs(double x)**

x యొక్క సంపూర్ణ విలువను అందిస్తుంది. **Syntax: double fabs(double x)**

4. **double log(double x):** C లైబ్రరీ ఫంక్షన్ **double log(double x)**

x యొక్క సహజ సంవర్ణమానాన్ని (base-e logarithm) అందిస్తుంది.

**Syntax: double log(double x)**

5. **double log<sub>10</sub>(double x):** C లైబ్రరీ ఫంక్షన్ **double log<sub>10</sub> (double x)**

x యొక్క సాధారణ సంవర్ణమానాన్ని (base-10 logarithm) అందిస్తుంది.

**Syntax: double log<sub>10</sub>(double x)**

6. **double fmod(double x, double y):** C లైబ్రరీ ఫంక్షన్ **double fmod**

**(double x, double y)** x యొక్క శేషాన్ని y తో భాగిస్తే అందిస్తుంది.

**Syntax: double fmod(double x, double y)**

7. **double sqrt(double x):** క్లైబ్రరీ ఫంక్షన్ **double sqrt (double x)**

'x' యొక్క వర్ణమూలాన్ని అందిస్తుంది. **Syntax: double sqrt(double x)**

8. **double pow(double x, double y):** C లైబ్రరీ ఫంక్షన్ **double pow**

(**double x, double y**)  $x$ ని  $y$  అంటే  $xy$  పవర్ కి పెంచింది.

**Syntax: double pow(double x, double y)**

9. **double modf(double x, double integer):** C లైబ్రరీ ఫంక్షన్

**double modf(double x, double \*పూర్ణాంకం)** భిన్నం కాంపోనెంట్ ను (దశాంశం తర్వాత భాగం) అందిస్తుంది మరియు పూర్ణాంకాన్ని పూర్ణాంక భాగానికి సెట్ చేస్తుంది.

**Syntax: double modf(double x, double integer)**

10. **double exp(double x):** C లైబ్రరీ ఫంక్షన్ **double exp(double x)** 'e' విలువను

$x$ వ పవర్ కి పెంచింది.

**Syntax: double exp(double x)**

4. **Local మరియు Global variables మధ్య తేడాలు తెల్పండి?**

**What are the differences between global variable and local variable.**

జ. గ్లోబల్ మరియు లోకల్ వేరియబుల్స్ మధ్య ప్రధాన తేడాలు:

- 1) గ్లోబల్ వేరియబుల్స్ ఫంక్షన్ వెలుపల నిర్వచించబడతాయి, అయితే స్థానిక వేరియబుల్స్ ఫంక్షన్ లోపల నిర్వచించబడతాయి.
- 2) ప్రోగ్రామ్ ప్రారంభమయ్యే సమయంలో గ్లోబల్ వేరియబుల్స్ సృష్టించబడతాయి మరియు ప్రోగ్రామ్ ముగిసే వరకు ఉంటాయి. స్థానిక వేరియబుల్స్ అనే ఫంక్షన్ సమయంలో సృష్టించబడతాయి మరియు ఫంక్షన్ ఎగ్జిక్యూషన్ ముగిసినప్పుడు నాశనం చేయబడతాయి.
- 3) గ్లోబల్ వేరియబుల్స్ మొత్తం ప్రోగ్రామ్ లో అందుబాటులో ఉంటాయి. ఇవి సజీవంగా, చురుకుగా మరియు నాశనం చేయబడ్డాయి. స్థానిక వేరియబుల్స్ నిర్దిష్ట ఫంక్షన్ కు మాత్రమే అందుబాటులో ఉంటాయి.

5. **Recursion method ని ఉపయోగించి GCD విధానాన్ని వివరించండి.**

**Explain GCD procedure using recursion method.**

జ. GCD (గ్రేటెస్ట్ కామన్ డివైజర్) రెండు సంఖ్యలు రెండింటినీ విభజించే అతిపెద్ద సంఖ్య.

```
int ged(int a, int b) {  
    if (b != 0)  
        return gcd(b, a % b);  
    else  
        return a; }  
}
```

## 6. Write a program to find factorial of given number.

```
జ. /* To find factorial of given number */
    # include <stdio.h>
    int main () {
    int n, f;
    printf("\n enter any value: \n");
    scanf("%d",&n);
    f= fact (n);
    printf(" factorial of %d is", n, f);
    // getch ( );
    }
    fact (int n) {
    if (n==0)
    return (1);
    else
    return (n*fact (n-1));
    }
```

**Output:** enter any value:5

factorial of 5 is 120

## 7. Call-by-value ని సహితము వివరించుము?

**Explain call by value with suitable example.**

జ. Main program నుండి sub program కు విలువలు పంపిస్తున్నప్పుడు (parameters) function లోకి copy అవుతాయి. ఈ విధంగా functionకు main programకు విలువలను పంపిస్తారు. దీనినే parameters passing అంటారు. గ్రహించిన విలువను function తమను పూర్తి చేస్తుంది. parameter మరియు రెండు రకాలు. అవి

**call-by-value:-** Main program నుండి sub program (function)కు విలువలు పంపిస్తున్నప్పుడు, ఆ functionలోని parameters కి directగా copy అవుతాయి. ఏవైనా మార్పులు (updates) చేసిన అవి functionలోని విలువలకు మాత్రమే వర్తిస్తాయి. కాని main ఒక function వేరొక function ను call చేసినప్పుడు కూడా ఈ సూత్రం వర్తిస్తుంది.



**call-by-value ఉదాహరణ ద్వారా పరిశీలిద్దాం:**

```
#include <stdio.h>

void call_by_value(int x) {
    printf("Inside call_by_value x = %d before adding 10.\n", x);
    x +=10;
    printf("Inside call_by_value x = %d after adding 10.\n", x);
}

int main(){
    int a=10;
    printf("a = %d before function call_by_value.\n", a);
    call_by_value(a);
    printf("a = %d after function call_by_value.\n", a);
    return 0;
}
```

**The output of this call by value code example will look like this:**

a= 10 before function call\_by\_value.

Inside call\_by\_value x 10 before adding 10.

Inside call\_by\_value x 20 after adding 10.

a = 10 after function call\_by\_value.

**8. Call by reference ను సహితంగా వివరించుము?**

**Explain the call by reference with suitable example.**

**జ. Call-by reference:** Main program నుండి sub programకి విలువలు పంపిస్తున్నప్పుడు విలువలు కాక ఆ విలువలు n లో ఎక్కడ నిల్వచేయుడి ఉన్నవో ఆ memory location యొక్క addressను పంపిస్తారు దీని వల్ల functionలో విలువలను ఏవైనా మార్పులు (update) చేస్తే ఆ మార్పులు Main programలో కూడా చూరతాయి. memory యొక్క address ను పంపిండాంటే pointers సంబంధించిన విషయ పరిజనం అవసరం అవుతుంది.

Let's take a look at a code example:

```
#include <stdio.h>

void call_by_reference(int *y)
{
    printf("Inside call_by_reference y = %d before adding 10.\n", *y);
    (*y) += 10;
    printf("Inside call_by_reference y = %d after adding 10.\n", *y);
}

int main() {
    int b=10;
    printf("b = %d before function call_by_reference.\n", b);
    call_by_reference(&b);
    printf("b= %d after function call_by_reference.\n", b);
    return 0;
}
```

The output of this call by reference source code example will look like this:

b = 10 before function call\_by\_reference.

Inside call\_by\_reference y = 10 before adding 10.

Inside call\_by\_reference y = 20 after adding 10.

b= 20 after function call\_by\_reference.

9. 'C' ఫంక్షన్స్ గురించి వివరించండి Explain the category of function In 'C'.

జ. Functions are categorized as 3 types.

1. function with no arguments and no values
2. function with arguments and no values
3. function with arguments and values

The following are the examples of above categories:

```
/* 1. function with no arguments and no values */
#include<stdio.h>
int main() {
int a,b;
printf("Enter any two values:");
scanf("%d%d",&a,&b);
add(a,b);
return 0;
}
add(int a, int b) {
int c;
c = a + b;
printf("Addition of %d + %d is= %d\n", a,b,c);
}
```

**Output:**

Enter any two values: 30 20

Addition of 30 and 20 is = 50

```
/* 2. function with arguments and no values */
#include<stdio.h>
int main( ) {
int a,b,c;
printf ("Enter any two values:");
scanf ("%d %d",&a,&b);
c=add(a,b);
printf("addition of two values is %d", c);
return 0;
} add(int a,int b) {
return(a+b); }
```

**Output:** Enter any two values: 20 30

addition of two values is=50

```
/* 3. function with arguments and values */
```

```
#include<stdio.h>
```

```
int sum(int x, int y) {
```

```
return(x+y);
```

```
}
```

```
int main() {
```

```
int a,b,c;
```

```
printf("Enter values for a and b: ");
```

```
scanf("%d %d", &a,&b);
```

```
c=sum(a,b);
```

```
printf("\n Sum of %d and %d is = %d",a,b,c);
```

```
return 0;
```

```
}
```

**Output:** Enter values for a and b: 6 3

Sum of 6 and 3 is = 9

10. Call-by-value మరియు Call-by reference ల మధ్య భేదాలను తెలపండి.

What are the difference between call-by-value and call-by reference.

| Call By Value  | Call By Reference   |
|--|---|
| 1) ఒక ఫంక్షన్‌కు కాల్ చేస్తున్నప్పుడు, మేము దానికి వేరియబుల్స్ విలువలను పాస్ చేస్తాము. ఇటువంటి విధులు "కాల్ బై వాల్యూస్" అంటారు. | 1) ఒక ఫంక్షన్‌కు కాల్ చేస్తున్నప్పుడు, వేరియబుల్స్ యొక్క విలువలను పాస్ చేయడానికి బదులుగా, మేము వేరియబుల్స్ చిరునామాను (వేరియబుల్స్ యొక్క స్థానం) "కాల్ బై రిఫరెన్స్" అని పిలిచే ఫంక్షన్‌కు పాస్ చేస్తాము. |
| 2) ఈ పద్ధతిలో, కాల్‌ింగ్ ఫంక్షన్‌లోని ప్రతి వేరియబుల్ విలువ కాల్డ్ ఫంక్షన్‌కి సంబంధించిన డమ్మీ వేరియబుల్స్‌కి కాపీ చేయబడుతుంది.  | 2) ఈ పద్ధతిలో, కాల్‌ింగ్ ఫంక్షన్‌లోని వాస్తవ వేరియబుల్స్ యొక్క చిరునామా కాల్ చేయబడిన ఫంక్షన్ యొక్క డమ్మీ వేరియబుల్స్‌లోకి కాపీ చేయబడుతుంది.   |

|   |  |
|---|--|
| <p>3) ఈ పద్ధతిలో, కాల్ ఫంక్షన్లోని డమ్మీ వేరియబుల్స్‌కు చేసిన మార్పులు కాలింగ్ ఫంక్షన్లోని వాస్తవ వేరియబుల్స్ విలువలపై ప్రభావం చూపవు.</p>   | <p>3) ఈ పద్ధతిలో, చిరునామాలను ఉపయోగించి మనకు వాస్తవ వేరియబుల్స్‌కు ప్రాప్యత ఉంటుంది మరియు అందువల్ల మేము వాటిని మార్చగలము.</p>  |
| <p>4) కాల్-బై-వాల్యూస్‌లో, మేము ఫంక్షన్ కాలింగ్ ద్వారా వాస్తవ వేరియబుల్స్ విలువలను మార్చలేము.</p>   | <p>4) సూచన ద్వారా కాలింగ్ మనం ఫంక్షన్ కాలింగ్ ద్వారా వేరియబుల్స్ విలువలను మార్చవచ్చు.</p>  |
| <p>5) వేరియబుల్స్ యొక్క విలువలు సాధారణ సాంకేతికత ద్వారా ఆమోదించబడతాయి.</p>  | <p>5) వేరియబుల్స్ యొక్క చిరునామా విలువలను నిల్వ చేయడానికి పాయింట్ వేరియబుల్స్ నిర్వచించాల్సిన అవసరం ఉంది.</p>  |
| <p>6) #include &lt;stdio.h&gt;<br/> void swap(int x, int y){<br/>     int temp = x;<br/>     x = y;<br/>     y = temp;<br/> }<br/> int main(){<br/>     int x = 10;<br/>     int y = 11;<br/> printf("Values before swap: x = %d, y = %d\n", x,y);<br/> swap(x,y);<br/> printf("Values after swap: x = %d, y = %d", x,y); }</p> | <p>6) #include &lt;stdio.h&gt;<br/> void swap(int *x, int *y){<br/>     int temp = *x;<br/>     *x = *y;<br/>     *y = temp;<br/> }<br/> int main() {<br/>     int x = 10;<br/>     int y = 11;<br/> printf("Values before swap: x = %d, y = %d\n", x,y);<br/> Swap(&amp;x,&amp;y);<br/> printf("Values after swap: x = %d, y = %d", x,y); }</p> |

## UNIT-V

### Structures in 'C'

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Structure అనగానేమి? What is structure?

**జ. Structure:** ఒకటిగాని అంతకన్నా ఎక్కువగాని చాలారాశులు వివిధ data typeకి చెందిన వాటిని ఒక groupగా చేసి ఒకే పేరులో పిలిచే దానినే Structure అంటారు. (లేదా) వివిధ రకములైన datatypesకి చెందిన మూలకాలు సముదాయాన్నే structure అంటారు.

#### 2. Struct syntax గురించి వ్రాయండి. What is syntax of struct?

**జ. Syntax of Struct:**

|                                |                     |
|--------------------------------|---------------------|
| struct structure name          | Ex : struct student |
| {                              | {                   |
| Datatype member 1 ;            | Int std id ;        |
| Datatype member 2 ;            | Float marks ;       |
| Datatype member 3 ;            | Char name [20] ;    |
| -----                          | };                  |
| Datatype member n ;            |                     |
| }; Note: struct is a key word. |                     |

#### 3. Structure ఉపయోగాలను తెలపండి. What are the advantages of structure?

**జ. నిర్మాణం యొక్క ప్రయోజనాలు:**

- స్ట్రక్చర్లను ఉపయోగించడం ద్వారా మన ప్రోగ్రామ్లో ఉపయోగించగల విభిన్న డేటా రకంతో వేరియబుల్స్ నిల్వ చేయవచ్చు.
- నిర్మాణం int, float, double, char వంటి డేటా రకాన్ని నిల్వ చేయడానికి అనుమతిస్తుంది.
- నిర్మాణాన్ని ఉపయోగించడం ద్వారా మనం ఇచ్చిన నిర్మాణం యొక్క శ్రేణిని సృష్టించవచ్చు మరియు ఆ నిర్మాణంలోని అన్ని వేరియబుల్స్ను యాక్సెస్ చేయవచ్చు లేదా సవరించవచ్చు.
- ప్రోగ్రామ్ యొక్క సంక్లిష్టతను తగ్గించడానికి నిర్మాణం సహాయం చేస్తుంది.

4. Nested structure అనగా నేమి? What is nested structure ?

జ. **Nested Structure:** నిర్మాణం నిర్దిష్ట సంఖ్యలో మూలకాలు / విభిన్న డేటా రకాల సభ్యులకు వెళుతుంది. స్ట్రక్చర్ లోని సభ్యులు స్ట్రక్చర్ డేటా రకానికి చెందినవారైతే, దానిని స్ట్రక్చర్ తో స్ట్రక్చర్ లేదా నెస్టెడ్ స్ట్రక్చర్ గా పేర్కొనవచ్చు.

5. Array యొక్క ఉపయోగాలను తెల్పండి. What are the advantages over an array ?

జ. శ్రేణిపై నిర్మాణం యొక్క ప్రయోజనాలు:

Arrays సజాతీయ (homogeneous) రకాల (ఇలాంటి (Similar) డేటా రకం) విలువలను నిల్వ చేయగలవు. నిర్మాణం (Structure) భిన్నమైన (heterogeneous) విలువలను (వివిధ డేటా రకాలు) నిల్వ చేయగలదు.

6. Union అనగా నేమి? What is a union?

జ. Union is a collection of heterogeneous type of data that share a single location. యూనియన్ అనేది ఒకే లొకేషన్ ను షేర్ చేసే వైవిధ్యమైన డేటా యొక్క సమాహారం.

7. Union syntax గురించి వ్రాయండి What is the syntax of union?

జ. Union syntax

```

union structure name          |x :  union student
{
    datatype member 1 ;
    datatype member 2 ;
    datatype member 3 ;
    -----
    datatype member
}

```

1. **structure** అంటే ఏమిటి? వివరంగా వివరించండి. **What is structure? Explain in detail.**

**జ. Structure:** ఒకే పేరుతో వ్యవస్థీకరించబడిన వివిధ డేటా రకాల ఒకటి లేదా అంతకంటే ఎక్కువ వేరియబుల్స్ యొక్క సమూహాన్ని స్ట్రక్చర్ అంటారు.

**ఎ. Structure Declaration** స్ట్రక్చర్ డిక్లరేషన్ వివిధ డేటా ఐటెమ్లకు ఎటువంటి వనరులను కేటాయించకుండా ఒకే యూనిట్ గా సమూహాన్ని నిర్దేశిస్తుంది.

**The syntax for declaring a structure in C is as follows:**

```
struct Structure Name>
{
    datatype member-1;
    datatype member-2;
    -----
    datatype member-n; };
```

స్ట్రక్చర్ డిక్లరేషన్ స్ట్రక్చర్ హెడర్ తో మొదలవుతుంది, ఇందులో 'struct' అనే కీవర్డ్ తో పాటు ట్యాగ్ ఉంటుంది. ట్యాగ్ స్ట్రక్చర్ పేరుగా పనిచేస్తుంది, ఇది స్ట్రక్చర్ వేరియబుల్స్ ను రూపొందించడానికి ఉపయోగించబడుతుంది. నిర్మాణం యొక్క వ్యక్తిగత సభ్యులు కర్లీ జంట కలుపుల మధ్య జతచేయబడి ఉంటాయి మరియు అవి సారూప్యమైన లేదా అసమాన డేటా రకాలుగా ఉండవచ్చు. ప్రతి వేరియబుల్ యొక్క డేటా రకం వ్యక్తిగత సభ్యుల ప్రకటనలలో పేర్కొనబడింది.

**Example:**

ఉద్యోగి సంఖ్య, పేరు మరియు జీతంతో కూడిన ఉద్యోగి డేటాబేస్ ను పరిశీలిద్దాం.

ఈ సమాచారాన్ని ఉంచడానికి నిర్మాణ ప్రకటన క్రింద చూపబడింది:

```
struct employee {
    int eno;
    char name [80];
    float sal;
};
```

**బి. Structure Variables:** ఇతర రకాల వేరియబుల్స్ లాగానే, స్ట్రక్చర్ డేటా టైప్ వేరియబుల్స్ స్ట్రక్చర్ డెఫినిషన్ ఉపయోగించి డిక్లర్ చేయబడతాయి.



```

Example: struct {
    int rollno;
    char name[20];
    float average;
} a, b;

```

**సి. Structure Initialization:** నిర్మాణం యొక్క సభ్యులు ఇతర వేరియబుల్స్ వలె ప్రారంభించబడవచ్చు. ఇది ప్రకటన సమయంలో చేయవచ్చు.

```

Example: struct address {
    char name [20];
    char desgn [10];
    char place [10];
};

```

### డి. Accessing of Structure Members

ఇంతకు ముందు చూసినట్లుగా, పీరియడ్ ఆపరేటర్ (.)ని ఉపయోగించి నిర్మాణాన్ని వ్యక్తిగతంగా గుర్తించవచ్చు, గుర్తింపు తర్వాత, వాటికి కొన్ని విలువలను కేటాయించడం ద్వారా అలాగే స్ట్రక్చర్ సభ్యులలో నిల్వ చేసిన విలువలను పొందడం ద్వారా వాటినియాక్సెస్ చేయవచ్చు. కింది ప్రోగ్రామ్ నిర్మాణ సభ్యుల యాక్సెస్ను వివరిస్తుంది.

**ఉదాహరణ:** సమయాన్ని అంగీకరించడానికి మరియు దానిని ప్రదర్శించడానికి స్ట్రక్చర్ డెఫినిషన్ని ఉపయోగించి సి ప్రోగ్రామ్ను వ్రాయండి.

```

/* Program to accept time and display it */
#include <stdio.h>
void main() {
    Struct {
        int hour, min; float seconds;
    } time;
    printf ("Enter time in Hours, min and Seconds\n");
    scanf ("%d %d %f", &time. hour, & time. min, & time. seconds);
    printf ("The accepted time is %d %d %f", time. hour, time. min. time. seconds);
}

```

2. structure members ఎలా యాక్సెస్ చేయాలి? ఉదాహరణతో వివరించండి.

**How to access structure members? Explain with example.**

జ. **Access structure members:** స్ట్రక్చర్ మెంబర్లను యాక్సెస్ చేయవచ్చు మరియు అనేక మార్గాల్లో విలువలను కేటాయించవచ్చు. ఏదైనా నిర్మాణ సభ్యునికి విలువను కేటాయించడానికి. సభ్యుని పేరు తప్పనిసరిగా డాట్ (.) ఉపయోగించి స్ట్రక్చర్ వేరియబుల్ తో లింక్ చేయబడాలి. ఆపరేటర్ పీరియడ్ లేదా మెంబర్ యాక్సెస్ ఆపరేటర్ అని కూడా పిలుస్తారు.

**Example::**  
# include <stdio.h>

```
# include <string.h>
```

```
struct student {
```

```
char name [20];
```

```
int age;
```

```
char class [10];
```

```
char marks;
```

```
};
```

```
int main ( )
```

```
{ struct student s1;
```

```
s1. Age =18;
```

```
strcpy ( s1.name, "Raju");
```

```
printf("Name of the student 1 is %s \n", s1.name);
```

```
printf ("Age of the student %d \n", s1.age);
```

```
return 0;
```

```
}
```

**Output:**

name of student1 : Raju

A age of the student : 18.

We can also use **scanf ( )** to give values to **structure members**

through terminal. **Scanf (" %s", s1.name); Scanf ("%d", s1 . age );**

### 3. Array Structure నివివరించండి. Explain the array of structures.

జ. **Array of structure:** మనము స్ట్రక్చర్ వేరియబుల్ యొక్క శ్రేణిని కూడా ప్రకటించవచ్చు, దీనిలో శ్రేణిలోని ప్రతి మూలకం స్ట్రక్చర్ వేరియబుల్ ను సూచిస్తుంది.

**ఉదాహరణ:** Struct student [5];

దిగువ ప్రోగ్రామ్ పరిమాణం 5 విద్యార్థి యొక్క శ్రేణిని నిర్వచిస్తుంది. శ్రేణి **Stu** యొక్క ప్రతి మూలకం రకంగా ఉంటుంది.

```
Student: # include <stdio.h>
struct student {
    char name [20];
    int marks;
};
struct student stu [5];
int i,j;
Valid st ( ) {
for ( i=0; i<2; i++) {
printf (“\n enter %dst student Information \n” student i++);
printf (“\n name \n”);
scanf (“ %s, stu[i]. name);
printf (“\n enter age \n”);
scanf (“%d”, & stu.age \n”);
}
```

**Array with in Structure :** స్ట్రక్ డేటా రకం శ్రేణి రకం వేరియబుల్ ను దాని సభ్యులుగా ఉంచవచ్చు. మనము నిర్మాణం యొక్క సభ్యుడిని int, float, char మొదలైన వాటికి సమానమైన Array డేటా రకంగా ప్రకటించవచ్చు.

```
ఉదాహరణ: struct student {
    char stu- name [20];
    int sno;
    int marks;
};
```

పైన పేర్కొన్నదానిలో, స్ట్రక్చర్ వేరియబుల్ విద్యార్థి సభ్యునిగా అక్షర శ్రేణి రకం స్ట్రానేమ్ను కలిగి ఉంది. దీన్ని ప్రారంభించడం సాధారణంగా చేయవచ్చు.

```
struct stuname = { " Raju", 900};
```

**Structure of function arguments:** మనం ఏదైనా ఇతర వేరియబుల్ లేదా శ్రేణిని ఫంక్షన్ ఆర్గ్యుమెంట్లుగా పాస్ చేసినట్టే మనం ఒక స్ట్రక్చర్ను ఫంక్షన్ ఆర్గ్యుమెంట్లుగా పాస్ చేయవచ్చు.

```
ఉదాహరణ:      # include <stdio.h>
                struct student {  char name [20];
                int no ;
                };
                valid output (struct student st);
                void main ( )
                { struct student. Stu;
                printf ("\n enter student Information \n" );
                printf ("\n student name \n");
                scanf (" % S",st.name);
                printf ("\n enter student . no);
                output (stu);
                }
                valid output (struct student st); {
                printf ("\n student name is %s" stu.name);
                printf ("\n no is %d", stu.no);
                }
                printf ("\n print student information");
                for (i=0; i<2' i++) {
                printf ("\n student name is %S", stu [i]. name);
                printf ("\n Aage is %d stud [i].age");
                } }
                valid main ( ) {
                st ( ): }
```

#### 4. Union అంటే ఏమిటి? వివరంగా వివరించండి. What is Union? Explain in detail.

**జ. Union:** యూనియన్ అనేది 'C'లో అందుబాటులో ఉన్న ప్రత్యేక డేటా రకం, ఇది కొంత మెమరీ లొకేషన్లో వివిధ డేటా రకాలను నిల్వ చేయడానికి అనుమతిస్తుంది. మీరు మే మెంబర్లతో యూనియన్ను నిర్వచించవచ్చు, కానీ ఒక సభ్యులు మాత్రమే ఏ సమయంలోనైనా విలువను కలిగి ఉంటారు. యూనియన్లు బహుళ ప్రయోజనాల కోసం ఒకే మెమరీ లొకేషన్ను ఉపయోగించి సమర్థతా మార్గాన్ని అందిస్తాయి.

**Defining a union:** యూనియన్ని నిర్వచించడానికి మీరు ఒక నిర్మాణాన్ని నిర్వచించేటప్పుడు అదే విధంగా యూనియన్ స్ట్రక్చర్ను ఉపయోగించాలి. యూనియన్ స్ట్రక్చర్ మీ ప్రోగ్రామ్ కోసం ఒకటి కంటే ఎక్కువ మంది సభ్యులతో కొత్త డేటా రకాన్ని నిర్వచిస్తుంది. యూనియన్ ప్రకటన ఫార్మాట్ ఈ క్రింది విధంగా ఉంది.

```
union – name
{
    data – type          member1;
    data – type          member2;
    data – type          member3;
    -----
    data – type          member n;
}
```

Union \_ పేరు ఐచ్ఛికం మరియు ప్రతి సభ్యుని నిర్వచనం అనేది Int, float, char, array లేదా ఏదైనా ఇతర చెల్లుబాటు అయ్యే వేరియబుల్ నిర్వచనం వంటి సాధారణ వేరియబుల్ నిర్వచనం. చివరి సెమికోలన్కు ముందు యూనియన్ నిర్వచనం ముగింపులో; మీరు యూనియన్ వేరియబుల్స్ లేదా మరిన్నింటిని పేర్కొనవచ్చు కానీ ఇది ఐచ్ఛికం. i, f, str ముగ్గురు సభ్యులు ఉన్న తేదీ పేరుతో యూనియన్ రకాన్ని మీరు నిర్వచించే విధానం ఇక్కడ ఉంది.

```
union data {
    int i;
    float a;
    char st [30];
} info;
```

ఈ క్రింది ఉదాహరణ ఎగువ యూనియన్ ద్వారా ఆక్రమించబడిన మొత్తం మెమరీ పరిమాణాన్ని ప్రదర్శిస్తుంది.

```

# include<stdio.h>
# include <string.h>
union data {
    int i;
    float a;
    char st [30]; }
int main ( )
union date st;
printf (“memory size occupied data :%d \n”, size of (data));
return 0; }

```

It produce the following result. Memory size occupied by data : 30.

**5. Union memebbers ఎలా యాక్సెస్ చేయాలి? ఉదాహరణతో వివరించండి.**

**How to access Union memebbers? Explain with example.**

**జ. Accessing union members:** యూనియన్‌లోని ఏదైనా సభ్యుడిని యాక్సెస్ చేయడానికి, మేము మెంబర్ యాక్సెస్ ఆపరేటర్(.) ని ఉపయోగిస్తాము. సభ్యుని యాక్సెస్ వేరియబుల్ యూనియన్ వేరియబుల్ మధ్య కాలంగా కోడ్ చేయబడింది మేము యాక్సెస్ చేయాలనుకుంటున్న పేరు మరియు యూనియన్ సభ్యుడు. యూనియన్ \_ రకం యొక్క వేరియబుల్‌ను నిర్వచించడానికి మీరు యూనియన్ అనే కీవర్డ్‌ని ఉపయోగిస్తారు. ప్రోగ్రామ్‌లో యూనియన్‌ను ఎలా ఉపయోగించాలో క్రింది ఉదాహరణ చూపిస్తుంది.

```

# include <stdio.h>
# include <string.h>
union data
{
    int i;
    float a;
    char str [30];
};
int main ( ) {
    union data st;
    Data.i = 10;

```

```

        Data.f = 322.7;
        Strcpy (data .st," the union");
        printf ("data.i %d \n", data.i);
        printf ("data.a %f \n", data.f);
        printf ("data.str: %S \n", data.str);
        return 0;
    }

```

When the above code is compiled and executed it produce the following result

Data.i : 2352612.

Data.f : 625232542567892257625.00000

Data.str: the union.

వేరియబుల్కు కేటాయించిన చివరి విలువ మెమరీ స్థానాన్ని ఆక్రమించినందున యూనియన్లోని i మరియు f సభ్యుల విలువలు పాడైపోయాయని ఇక్కడ మనం చూడవచ్చు మరియు str మెంబర్ విలువ బాగా ముద్రించబడటానికి కారణం ఇదే. ఇప్పుడు మనం ఒక సమయంలో ఒక వేరియబుల్ని ఉపయోగించే అదే ఉదాహరణను మరోసారి చూద్దాం. యూనియన్లను కలిగి ఉండటం యొక్క ముఖ్య ఉద్దేశ్యం ఏమిటి?

```

#include <stdio.h>
#include <string.h>
union data
{
    int i;
    float a;
    char str [20];
}
Main ( )
{
    union data data
    Data.i = 10
    printf ("data .i : %d \n", data . i);
    Data . a = 325.62

```

```

printf (“data.a : %f \n”);
strcpy (data.str,” the union, data.a );
printf (“data . str :%s \n”, data . str);
return 0;
}

```

When the above code is compiled and executed it produce the following result Data. i = 10 Data. F= 325.62 Data . str : the union

Here all the members are getting printed very well because one number is being used at a time.

## 6. Structure మరియు union ల మధ్య బేధాలను తెలపండి.

What is Differences between structure and union?

### జ. Differences between structure and union.

| Structure  | Union  |
|--|--|
| 1) కీవర్డ్ struct ఉపయోగించబడుతుంది ఒక నిర్మాణాన్ని నిర్వచించండి.   | 1) యూనియన్ అనే కీవర్డ్ ఉపయోగించబడుతుంది యూనియన్ నిర్వచించండి.  |
| 2) ఒక వేరియబుల్ యూనియన్ తో అనుబంధించబడినప్పుడు, కంపైయర్ ప్రతి సభ్యునికి మెమరీని కేటాయిస్తుంది. నిర్మాణం యొక్క పరిమాణం దాని సభ్యుల పరిమాణం మొత్తం కంటే ఎక్కువ లేదా సమానంగా ఉంటుంది. | 2) ఒక వేరియబుల్ యూనియన్ తో అనుబంధించబడినప్పుడు, కంపైలర్ అతిపెద్ద మెమరీ పరిమాణాన్ని పరిగణనలోకి తీసుకోవడం ద్వారా మెమరీని కేటాయిస్తుంది. కాబట్టి యూనియన్ పరిమాణం అతిపెద్ద సభ్యుల పరిమాణానికి సమానం. |
| 3) సభ్యునివిలువను మార్చడం నిర్మాణంలోని ఇతర సభ్యులపై ప్రభావం చూపదు  | 3) ఏదైనా విలువను మార్చడం ఇతర సభ్యుల విలువలను మార్చిన సభ్యుడు.  |
| 4) వ్యక్తిగత సభ్యుడిని ఒకేసారి యాక్సెస్ చేయవచ్చు.  | 4) ఒకేసారి ఒక సభ్యుడు మాత్రమే యాక్సెస్ చేయగలరు.  |
| 5) నిర్మాణంలోని అనేక మంది సభ్యులు ఒకేసారి ప్రారంభించగలరు.  | 5) నిర్మాణంలోని అనేక మంది సభ్యులు ఒకేసారి ప్రారంభించగలరు   |



# QUESTION BANK

## UNIT-I

### Introduction to Problem Solving Techniques

స్వల్ప ప్రశ్నలు

1. Procedure అనగా నేమి? What is procedure?
2. Algorithm అనగా నేమి? Define an algorithm.
3. ఇవ్వబడిన రెండు పూర్ణాంకాల సంఖ్యలలో పెద్దదాన్ని ముద్రించడానికి అల్గారిథమ్ను వ్రాయండి.

Write an algorithm to print the biggest of given two integer numbers.

4. Flow-chart ని నిర్వచించుము Define flow-chart.
5. Flow-chart చిహ్నాలు ఏమిటి? What are the flow-chart symbols ?
6. Draw a flow-chart to calculate the simple interest using ( $SI=P*N*R/100$ ).
7. Draw a flow-chart to find the remainder values of given two numbers.
8. Draw a flow-chart to find the biggest value of given two integer numbers.
9. Draw a flow-chart to calculate product of given two numbers.
10. Pseude code అనగా నేమి? What is pseudo code ?
11. Draw flow-chart to print sum of two given numbers.

ధీర్ఘకాలిక ప్రశ్నలు

1. Flow-chart ఉపయోగాలను తెల్పుండి. What are the advantages of flow-chart?
2. Alogritham మరియు Flow-chart మధ్య గల భేదాలు వ్రాయుము.  
What are the differences between algorithm and flow-chart ?

3. ఇచ్చిన మూడు పూర్ణాంకాల సంఖ్యలలో అతి చిన్న విలువను కనుగొనడానికి అల్గారిథమ్ను వ్రాయండి.

Write an algorithm to find smallest value in given three integer numbers

4. ఇచ్చిన సంఖ్య ఫ్రైమ్లో ఉందో లేదో కనుగొనడానికి అల్గారిథమ్ను వ్రాయండి.

Write an algorithm to find whether the given number in prime or not .

5. Write an algorithm to find the reverse value of given 4 digit number.
6. మూడు సంఖ్యలలో పెద్ద సంఖ్యను కనుగొనుటకు flowchart ని గీయుము .

Draw a flow - chart to find the biggest value of a given three numbers.

7. Draw a flow-chart to find pass or failed of a student of a given 5 subject marks each subject maximum marks are 50, if total marks  $\geq 180$  results is pass or fail.

8. Draw a flow-chart to find the average of 5 given numbers.

9. Draw a flow-chart to find the sum of 1 to n natural numbers.

10. Draw a flow chart to find sum of 1 to n even numbers.

11. Draw a flow-chart to find the small value of given three numbers.

12. Draw a flow-chart to find whether the given number is even or odd.

13. ఇచ్చిన సంఖ్య పాలిండ్రోమ్ కాదా అని తనిఖీ చేయడానికి అల్గారిథమ్ను వ్రాయండి.

Write an algorithm to check whether the given number palindrome or not.

14. Write an algorithm to calculate the simple interest using  $(PNR / 100)$ .

15. What are the flow-chart symbols ?

---

## UNIT-II

### Features of 'C'

స్వల్ప ప్రశ్నలు

1. C language ని ఎవరు అభివృద్ధి చేసారు? Who developed C Language?
2. C program యొక్క ఆకారమును వ్రాయుము. Write the structure of C program.
3. What is character set? Write the character which used in 'C' language.
4. Constant అనగా నేమి? What is a Constant ?
5. వివిధ రకాల Constant లను వ్రాయండి. What are different types of constants?
6. Numeric constant అనగా నేమి? What is a Numeric constant?
7. String constant అనగా నేమి? What is a String constant?
8. Variable అనగా నేమి? What is a Variable?
9. String variable అనగా నేమి? What Is a String variable?
10. Single string constant అనగా నేమి? What is a single string constant?
11. 'C' భాష యొక్క ప్రాథమిక డేటా రకాలు ఏమిటి?

What are the fundamental data types of 'C' language?

12. Arithmetical ఆపరేటర్లు ఏమిటి? What are the arithmetical operators?

13. Bitwise ఆపరేటర్లు అంటే ఏమిటి What are the bitwise operators?

14. షరతులతో కూడిన ఆపరేటర్లు ఏమిటి? What are the Conditional operators?

15. ఇన్పుట్ మరియు అవుట్పుట్ స్టేట్‌మెంట్‌లు అంటే ఏమిటి?

What are input and output statements?

16. printf () మరియు scanf() స్టేట్‌మెంట్ల సింటాక్స్(syntax) అంటే ఏమిటి?

What is syntax of printf () and scanf() statements.

17. getchar() అనగా నేమి? What is a getchar() function in C?

18. సాధారణ 'if' స్టేట్‌మెంట్ గురించి వ్రాయండి Write about Simple 'if' statement.

19. if-else స్టేట్‌మెంట్ యొక్క సింటాక్స్ ను వ్రాయండి. Write the syntax of if-else statement.

20. How many types of loops are available in C ?

21. Break statement అనగా నేమి? What is a break statement ?

22. Continue statement అనగా నేమి? What is a continue statement ?

23. switch() స్టేట్‌మెంట్ యొక్క syntax వ్రాయండి.

Write the syntax of switch () statement.

24. Goto statement గురించి వ్రాయండి. Write about 'goto' statement.

25. Numeric variable అనగా నేమి? What is a Numeric variable?

26. Logical operators వ్రాయండి. What are the Logical operators?

27. Puchar() అనగా నేమి? What is a puchar()?

28. C లో gets () function అనగా నేమి? What is a gets () function in C?

29. C లో puts () function అనగా నేమి? What Is a puts () function in C?

30. Nested loop అనగా నేమి? What is Nested loop?

ధీర్ఘకాలిక ప్రశ్నలు

1. C లో స్థిరంగా (Constant) ఉండేవి ఏమిటి? ఉదాహరణలతో వివరించండి.

What are constant in C? Explain with examples.

2. Variable అనగా నేమి? వివిధ రకాలను ఉదాహరణ లతో తెల్పండి.

What is variable? Explain the types of variable in C with example.

3. 'C'లో డేటా రకాలు ఏమిటి? వాటిని వివరించండి.

What are the data types in 'C' ? Explain them.

4. C లో ఆపరేటర్లు ఏమిటి వాటిని వివరించండి What are the operators in C? Explain them.

5. 'C'లో రిలేషనల్ ఆపరేటర్లు ఏమిటి? తగిన ఉదాహరణలతో వివరించండి.

What are the relational operators in 'C' ? Explain with suitable examples.

6. 'C' లో లాజికల్ ఆపరేటర్లు ఏమిటి? ఉదాహరణలతో వివరించండి.

What are the logical operators in C? Explain with examples.

7. ఇన్పుట్ మరియు అవుట్పుట్ స్టేట్‌మెంట్‌లు అంటే ఏమిటి? వాటిని వివరించండి.

What are input and output statements? Explain them.

8. 'C'లో షరతులతో కూడిన ప్రకటనలు ఏమిటి? వాటిని వివరించండి.

What are the conditional statements in 'C'? Explain them.

9. Break మరియు Continue statement ల మధ్య భేదాలను వ్రాయండి.

What are the differences between break-statement and continue statement?

10. Write a C program biggest value of given three numbers.

11. Write C program to find the given in vowel or alphabet character using switch statement.

12. 'C' లో లూప్‌లు ఏమిటి? వాటిని వివరించండి. What are the loops in C? Explain them.

13. Write a C program sum of 1 to n natural number. Using while loop.

14. Write C program to find the sum of 4 digit number using do... while loop.

15. Write a C program whether given numbers is palindrome or not.

16. Write a C program to find factorial of given number using recursion method.

17. C'లో if స్టేట్‌మెంట్‌లు గురించి వివరించండి. Explain if else, nested if statements in 'C'.

18. if స్టేట్‌మెంట్ గురించి ఉంటే ఉదాహరణతో వివరించండి.

Explain if else statement with example.

19. While మరియు and do while ల మధ్య తేడా తెల్పండి

What are the difference between while and do while statement?

20. Explain the nested loop with example.

21. Write a C program to find factorial value of a given numbers.

22. Write a C program to find Fibonacci series up to given numbers.

23. Write a C program to find the sum of 4 digits number.

24. Write a C program to find smallest of given three numbers.

UNIT-III  
Array in 'C'

స్వల్ప ప్రశ్నలు

1. Array అనగా నేమి? What is Array?
2. Array ను నిర్వచించండి. Definition of Array?
3. Arrays లో ఉన్న రకాలను వివరించుము? Write the types of Arrays.
4. Single Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి.  
Write the Syntax with example for Single Dimensional Array.
5. Two Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి.  
Write the Syntax with Example for Two Dimensional Array.
6. Multi Dimensional Array లో Syntax ను ఉదాహరణ తో వ్రాయండి  
Write the Syntax with Example for Multi Dimensional Array.

ధీర్ఘకాలిక ప్రశ్నలు

1. Write a 'C' Program to sort a given array of elements.
  2. Write a 'C' Program to perform matrix addition.
  3. Write a 'C' Program to perform matrix multiplication.
  4. Write a 'C' Program to create a single dimensional array of numbers and displaying them
  5. Write a 'C' program to find the sum and average of given array of elements.
  6. Write a 'C' Program to display the elements of an array in reverse order
  7. Write a 'C' Program to find the biggest element in an array with location
  8. Write a 'C' Program to find the length of the character array
  9. Write a 'C' Program to print transpose of a matrix.
-

## UNIT-IV

### Functions

స్వల్ప ప్రశ్నలు

1. Procedure అనగా నేమి? What is a procedure?
2. Function అనగానేమి? What is function? Define a function.
3. వివిధ రకాల functionలను తెలుపుము? What are the types of function?
4. Function యొక్క syntax అంటే ఏమిటి? ఒక ఉదాహరణ ఇవ్వండి.  
What is the syntax of function? Give an example.
5. ఫంక్షన్ యొక్క ప్రయోజనాలు ఏమిటి? What are the advantages of function?
6. Global variable అనగా నేమి? What is a global variable?
7. Local variable అనగా నేమి? What is a local variable?
8. Parameters అంటే ఏమిటి? what are the parameters?
9. Main () function అంటే ఏమిటి? What is a Main () function?
10. ఏదైనా 4 గణిత విధులను ఉదాహరణతో వ్రాయండి.

Write any 4 mathematical functions with example.(Or)

11. What any 4 arithmetical function give example.
12. Global string అనగా నేమి? What is a global string?
13. Recursion అనగా నేమి? What is a recursion?
14. Return statement ను ఉపయోగించండి. What is use of return statement?
15. void function యొక్క ఉపయోగం ఏమిటి? What is use of void function?
16. 'C' లో స్ట్రింగ్ హ్యాండింగ్ ఫంక్షన్ ఏమిటి? వాటిని వివరించండి.

What are the string handling function in 'c'? Explain them.

ధీర్ఘకాలిక ప్రశ్నలు

1. Function మరియు Procedureల మధ్య తేడాలను వివరించుము?

What are the differences between procedure and function?

2. Library functions ఏమిటి? తగిన ఉదాహరణలతో వివరించండి.

What are library functions? Explain with suitable examples.

3. Mathematical functions ఏమిటి? తగిన ఉదాహరణలతో వివరించండి.

What are the mathematical functions? Explain with suitable example.

4. Local మరియు Global variables మధ్య తేడాలు తెల్పండి?

What are the differences between global variable and local variable.

5. Recursion method ని ఉపయోగించి GCD విధానాన్ని వివరించండి.

Explain GCD procedure using recursion method.

6. Write a program to find factorial of given number.

7. Call-by-value ని సహితము వివరించుము?

Explain call by value with suitable example.

8. Call by reference ను సహితంగా వివరించుము?

Explain the call by reference with suitable example.

9. Explain the category of function In 'C'.

10. call-by-value మరియు call-by reference ల మధ్య భేదాలను తెలపండి.

What are the difference between call-by-value and.

---

## UNIT-V

### Structures in 'C'

స్వల్ప ప్రశ్నలు

1. Structure అనగానేమి? What is structure?

2. Struct syntax గురించి వ్రాయండి. What is syntax of struct?

3. Structure ఉపయోగాలను తెలపండి. What are the advantages of structure?

4. Nested structure అనగా నేమి? What is nested structure ?

5. Array యొక్క ఉపయోగాలను తెల్పండి.

What are the advantages over an array ?

6. Union అనగా నేమి? What is a union?

7. Union syntax గురించి వ్రాయండి What is the syntax of union?

1. structure అంటే ఏమిటి? వివరంగా వివరించండి.

What is structure? Explain in detail.

2. structure members ఎలా యాక్సెస్ చేయాలి? ఉదాహరణతో వివరించండి.

How to access structure members? Explain with example.

3. Array Structure నివివరించండి.Explain the array of structures.

4. Union అంటే ఏమిటి? వివరంగా వివరించండి.What is Union? Explain in detail.

5. Union members ఎలా యాక్సెస్ చేయాలి? ఉదాహరణతో వివరించండి.

How to access Union members? Explain with example.

6. Structure మరియు union ల మధ్య బేధాలను తెలపండి.

What is Differences between structure and union?

-----HARD WORK IS SECRETE OF SUCCESS-----