

## Question Bank (SPA SEM II)

### 1. Storage classes in C (Refer notes Page No 52)

### 2. Difference between function declaration and function definition

(This question is solved in the note book). But solution is given here again.

**Declaration of a function:** The declaration gives information about the function. First, it tells its return type. Second, it tells its name and third how many parameters are there, and what their types are. The declaration of function is always ended with semicolon.

**Definition of a function:** The definition tells what the function does.

```
void functionName(int x, double y);    /* example of declaration */
```

```
void functionName(int x, double y)    /* example of definition */
```

```
{  
    Statement1 ;  
    Statement2 ;  
    //this is function definition  
}
```

It is necessary to declare the function above the call, if the function is defined below the call.

For example the program given below will give error because 'add' function is defined below the call but it is not declared above the call.

```
#include <stdio.h>  
void main()  
{  
    float a,b,c;  
    printf("enter two floats ");  
    scanf("%f %f", &a, &b);  
    /*call the function add */  
    c = add(a,b); /* this line will give error */  
    printf("sum = %f\n", c);  
}  
float add(float x, float y)  
{  
    return x+y;  
}
```

To remove the error we must add a line above main() which is  
float add(float x , float y) ; /\*this is function declaration \*/

Note : A function which returns int value can be defined any where in the program and it is not necessary to declare such a function.

### **3 What is pointer. Example how to use pointer using example. OR**

#### **What is pointer. Explain referencing and dereferencing of pointer.**

(Refer notes Page no 73)

### **4. What are different types of functions.**

There are two types of functions

a) Library function : The in built functions defined in C language are called library functions. Examples of library functions are

clrscr() : this function is used for clear screen. It is defined in conio.h file.

sqrt() : this function is used for finding square root of a number. It is defined in math.h file.

strlen() : this function is used for finding length of a string. It is defined in string.h file.

To use any library function we must include correct .h file at the beginning of the program. This is done by using #include statement.

For example , if we want to use sqrt function in our program then we must include <math.h> as follows

```
#include <math.h>
```

User defined function : The function defined by the user in his/her program are called user defined function. For example we may define our own function 'add' which will add and return sum of 3 integers

```
int add(int x, int y , int z)
{
    return x+y+z;
}
```

## **5. What are data types in C**

Data type stands for type of data. We all know that computer can deal with numerical data and character data (like name of a person)

C supports four basic datatypes which are used to specify type of data that a variable can store.

(a) int : This data type is used for storing integers(or whole numbers) such as 15, -3 , 0, 35 and so on.

Consider the following statement

```
int x;
```

The above line is called declaration of a variable. Here variable x is declared of type int. So variable x can now store only integer type of value. Note that real numbers like 3.5 or -23.8 cannot be stored by int variable. Range of int type is -32768 to 32767.

Variable of int type consumes 2 bytes of memory.

(b) float : This data type is used for storing real numbers(or floating point values) such as 15.8, -3.67 , 0, 35.2 and so on.

Consider the following statement

```
float y;
```

The above line is called declaration of a variable. Here variable y is declared of type float. So variable y can now store any real number. Note that real number like 3 can also be stored by float variable. In such case 3 will be stored as 3.0 automatically.

Range of float data type is  $3.4 * 10^{-38}$  to  $3.4 * 10^{38}$ . This means that a float variable can store value with the above specified limits.

Variable of float type consumes 4 bytes of memory.

( c ) double : This data type is also used for storing real numbers. So double is actually same as float. but a variable of double data type can store larger real number than that a float variable. In other words range of double data type is larger than float data type.

Range of double data type is  $1.7 * 10^{-308}$  to  $1.7 * 10^{308}$  (That is a huge range of values)

This is why double data type is also called long float.

Variable of double type consumes 8 bytes of memory.

(d) char : This data type is used for storing one character such as 'a' or 'Z' etc.

The line

```
char t;
```

declares a variable t of type char. This variable t can store a single character.

Variable of char type consumes 1 bytes of memory.

### **6. What is meaning of identifier. What are rules for naming an identifier.**

Identifier : user defined name is called identifier. Example variable name is also called identifier. In the statement

```
int x;
```

x is called identifier or a variable.

Or function name is called identifier. Example

```
int abc(int x)
```

in the above line 'abc' is called identifier.

Identifier : Every variable(or identifier) has a name and there are some rule regarding naming a variable.

rule (a) : name of variable must start with alphabet or underscore.

rule (b) name of variable can contain only alphabets, digits and underscores.

Example of valid variable names are

1) a    2) a1    3) first    4) roll\_no    5) roll\_no\_of\_student

Examples of invalid variable names are

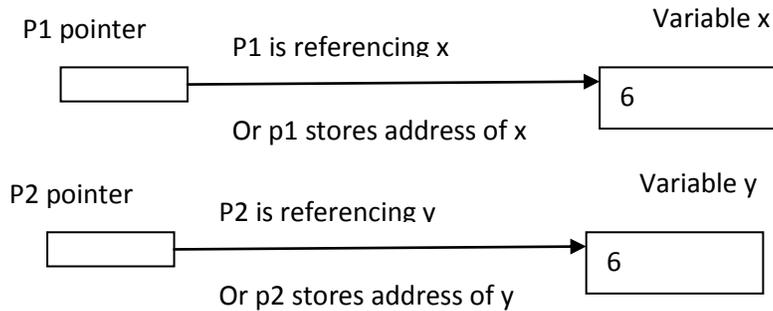
- 1) 1a (name cannot begin with a digit)
- 2) abc,def (name cannot contain comma)
- 3) abc def (name cannot contain space)

### **7. What is meant by referencing and deferencing of pointer?**

Ans : Pointer is variable which stores address(also called reference) of another variable stored in the memory. Consider a program code

```
int x = 6 , y = 7;  
int *p1 , *p2;  
p1 = &x; /*p1 is referencing to x or p1 is storing address of x */  
p2 = &y; /*p2 is referencing to y or p2 is storing address of y */
```

Hence referencing of a pointer means that pointer is storing address of a variable. Diagram shown below shows how p1 references x and p2 references y.

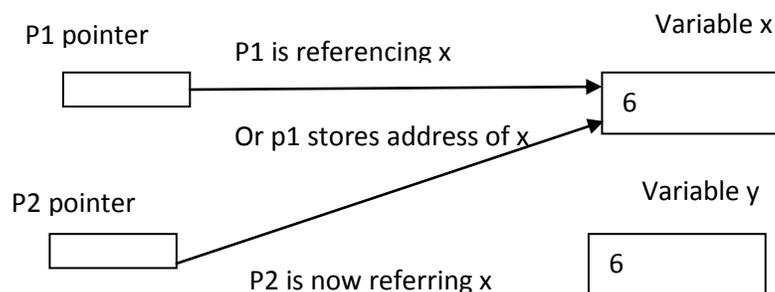


Dereferencing of pointer means that the pointer does not point to the variable. This can be done by changing value of the pointer. For example if we have a statement

```
p2 = p1;
```

Here p2 pointer points to the same variable to which p1 pointer.

So the pointer p2 will not point to variable y (dereference) but pointer p2 will now point to variable x.



### **8. What is an algorithm. Give characteristics or rules of well defined algorithms.**

1. A sequential solution of any program that written in human language is called algorithm.
2. Algorithm is first step of the solution process, after the analysis of problem, programmer writes the algorithm of that problem.

3. An algorithm is a finite set of steps defining the solution of a particular problem.
4. An algorithm can be expressed in human readable language like as English. Algorithm is language independent, well structured and detailed.

### **Rules for constructing an Algorithm**

When you are going to create an algorithms, keep following point in mind as:

- **Input:** Algorithm must be given some input.
- **Output:** At least one result is to be produced.
- **Definiteness:** Each step must be clear and unambiguous.
- **Finiteness:** If we trace the steps of an algorithm, then for all cases, the algorithm must terminate after a finite number of steps.
- **Comment Session:** Comment is additional information of program for easy understanding.

### **9. What is efficiency of an algorithm. Explain with example.**

Ans

Algorithm is said to be efficient if it satisfies following needs.

- a. Does the algorithm give desired output ?
- b. Does the algorithm solve the task given ?
- c. Is the documentation (comments) that describe the algorithm given.
- d. Is the code readable ?

There are other criteria for judging efficiency of algorithm, which are space complexity and time complexity.

An algorithm which requires lesser space (ie memory) is always better than algorithm which will consume more memory. Also algorithm should be fast to solve the given program in minimum time.

As an example , consider that we want to write an algorithm to add any 10 numbers and display their addition. The first method can be to declare 10 different variables and input each of the variable and finally find their sum. This would be a longer algorithm.

1. start
2. input a,b,c,d,e,f,g,h,i,j [input 10 integers]
3. calculate  $s = a+b+c+d+e+f+g+h+i+j$  [find sum]
4. print sum
- 5 stop

My GOD. Think what would happen if we were supposed to add 50 numbers(We can't declare 50 variables) The other method of writing the same algorithm will be to declare an array of 10 integers and find sum

1. start
2.  $i = 0$  ,  $s = 0$
3. repeat steps 3 to 6 while  $i \leq 9$
4. input  $a[i]$  [a is array of 0..9 elements]
5.  $i = i + 1$
6.  $s = s + a[i]$
7. print s
8. stop

### **10. What are various formats used in printf and scanf.**

Ans

`%d` for input/output integer  
`%f` for input/output float  
`%c` for input/output character  
`%lf` for input/output double  
`%ld` for input/output long int  
`%e` for printing float value in exponential format

### **11. What is break and continue statement**

Break statement is used for transferring the control out of a loop.[The loop can be for loop or while loop or do while loop]

Example

```
#include <stdio.h>
void main()
{
    for ( I = 1 ; I <= 10 ; i++)
    {
        if ( i == 5)
            break;

        printf(“%d\n” , i);
    }
}
```

Output of this program is

1  
2  
3  
4

The program has a for loop which repeats i from 1 to 10. For values of i = 1,2,3,4 the if condition i == 5 is false and hence the break statement does not execute. This means 1,2,3,4 is printed. But for value of i = 5 the condition i == 5 is true and hence the break statement runs which breaks the loop.

continue statement is used for transferring the control to the top of a loop.[The loop can be for loop or while loop or do while loop]

Example

```
#include <stdio.h>
void main()
{
    int i;
    for (i=1 ; i <= 10 ; i++ )
    {
        if ( i <= 5)
            continue;

        printf(“%d\n” , i);
    }
}
```

Output of this program is

```
6
7
8
9
10
```

The program has a for loop which repeats i from 1 to 10. For values of i = 1,2,3,4,5 the if condition i <= 5 is true and hence the loop continues without printing these values of i. But for values of i = 6,7,8,9,10 the condition i <= 5 is false and hence these values are printed.

## **12. What is ternary operator or conditional operator.**

Ans :

? is ternary or conditional operator. It is used as replacement of if else statement. In other words , the work done by if else statement can be done by using ? operator.

Example

```
if ( x > y)
    z = x;
else
    z = y;
```

The above if else can also be written as

$z = (x > y) ? x : y;$

This line means

If  $(x > y)$  is true then  $z$  becomes equal to  $x$  otherwise  $z$  becomes equal to  $y$ .

**13. Explain any 5 string functions.**

Ans : strlen , strcpy , strcat , strcmp , strcmpi

Refer notes for details. (pg 62)

**14. What is result of following expressions.**

i)  $10 \ \&\& \ 0$

ans : 0

reason : In C language any non zero value is treated as true and zero value is treated as false. So the statement  $10 \ \&\& \ 0$  becomes true  $\&\&$  false.

The result of true  $\&\&$  false is false. This value false is 0. Hence the result is 0.

ii)  $-5 \ || \ 2$

this is same as true  $||$  true which results in true.

Hence result of this expression is 1.

iii)  $! \ 10$

the not operator is applied to 10 (ie is true) and  $!true$  is false. Hence the result is 0.

iv)  $!! \ 10$

result : 1

**15 What are logical operators (pg 21)**

**16. Write a algorithm to find out number is odd or even?**

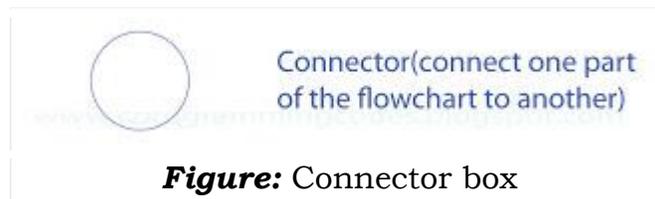
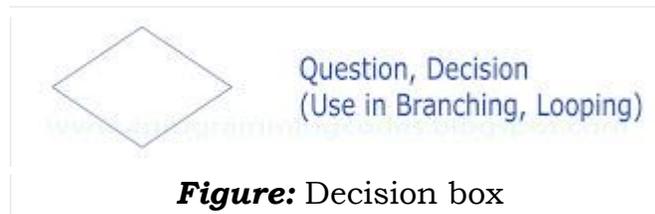
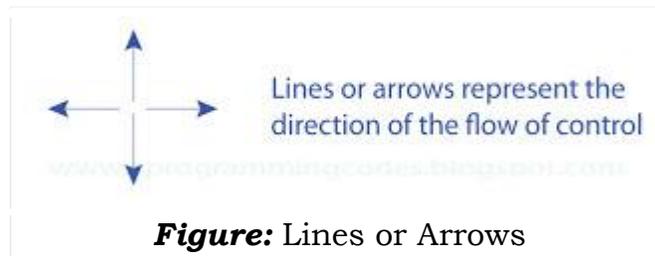
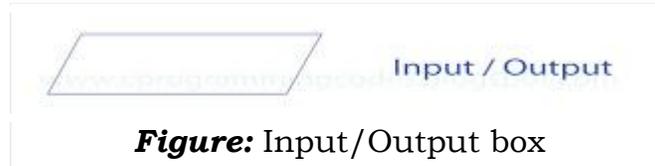
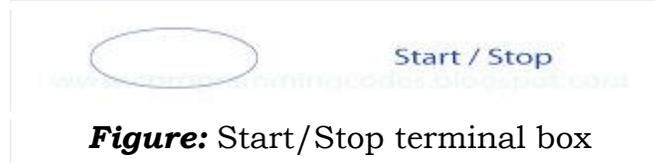
Ans.

```
step 1 : start
step 2 : input number
step 3 : rem=number mod 2
step 4 : if rem=0 then
    print "number even"
    else
    print "number odd"
    endif
step 5 : stop
```

**17. What is Flowchart. Explain with example?**

1. Graphical representation of any program is called flowchart.

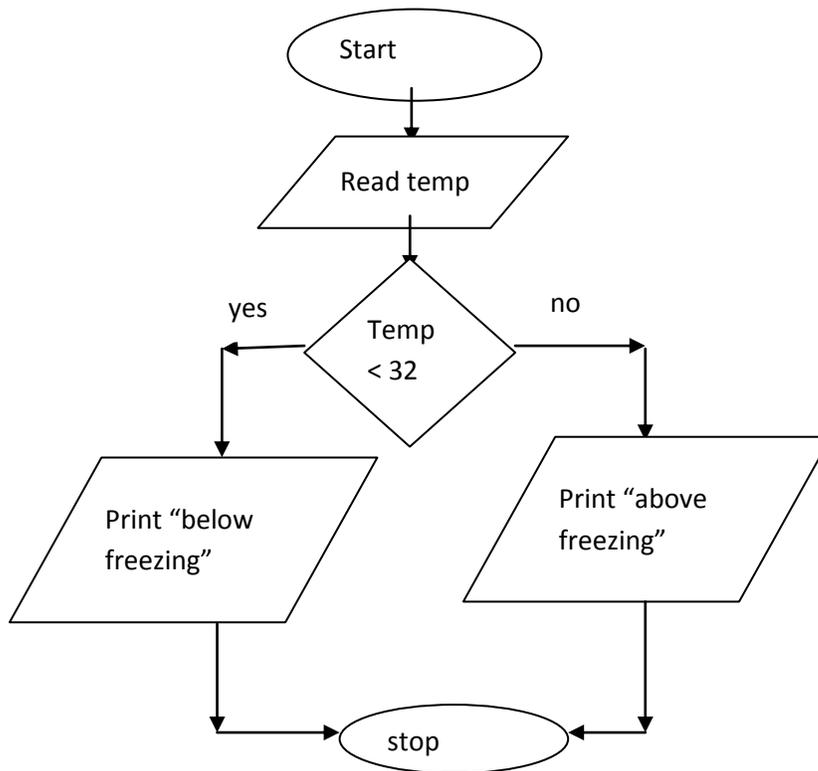
2. There are some standard graphics symbols that are used in flowchart as following:



**Example 1**

Make a flowchart to input temperature, if temperature is less than 32 then print "below freezing" otherwise print "above freezing"?

Ans.



### **Example 2**

```
/*c program to find out factorial value of a number*/  
#include<stdio.h>  
void main()  
{  
  int n,i,fact=1;  
  printf("Enter any number : ");  
  scanf("%d", &n);  
  for(i=1; i<=n; i++)  
    fact = fact * i;  
  printf("Factorial value of %d = %d",n,fact);  
}
```

### ***Algorithm for calculate factorial value of a number:***

[algorithm to calculate the factorial of a number]

step 1. Start

step 2. Read the number n

step 3. [Initialize]

i=1, fact=1

step 4. Repeat step 4 through 6 while  $i \leq n$

step 5.  $fact = fact * i$

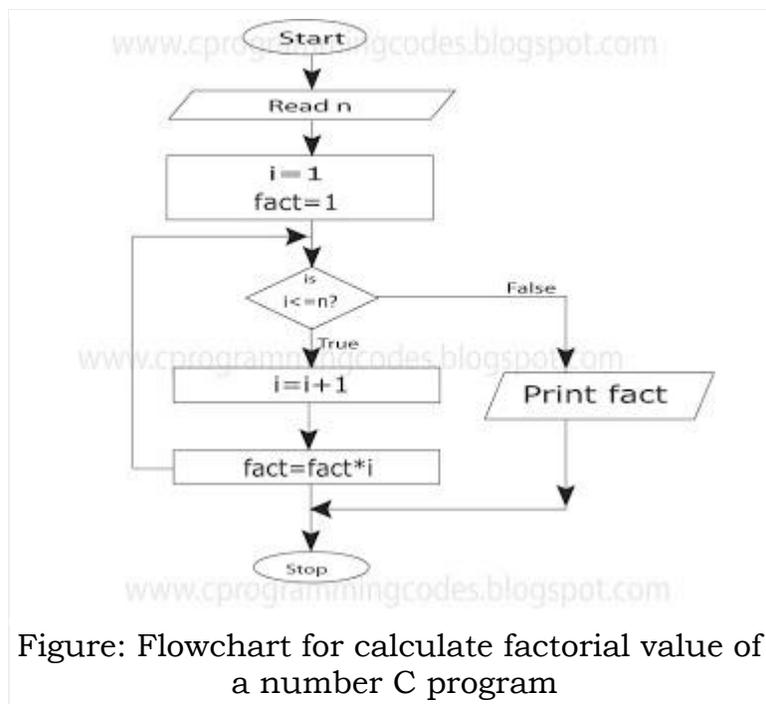
step 6.  $i = i + 1$

step 7. Print fact

step 8. Stop

[process finish of calculate the factorial value of a number]

**Flowchart for calculate factorial value of a number:**



**18. Draw Flowchart and write Algorithm for finding Armstrong number.**

**C program:**

```
#include <stdio.h>
void main()
{
    int num , rem , sum = 0 , n;

    // input the number n
    printf(" enter an integer ");
    scanf("%d" , &num);
    n = num;
    for ( ; n >= 1 ; )
    {
        rem = n % 10;
        sum = sum + rem * rem* rem;
        n = n / 10;
    }

    if ( sum == num)
        printf( "number is amstrong" );
    else
        printf("number is not amstrong " );
}
```

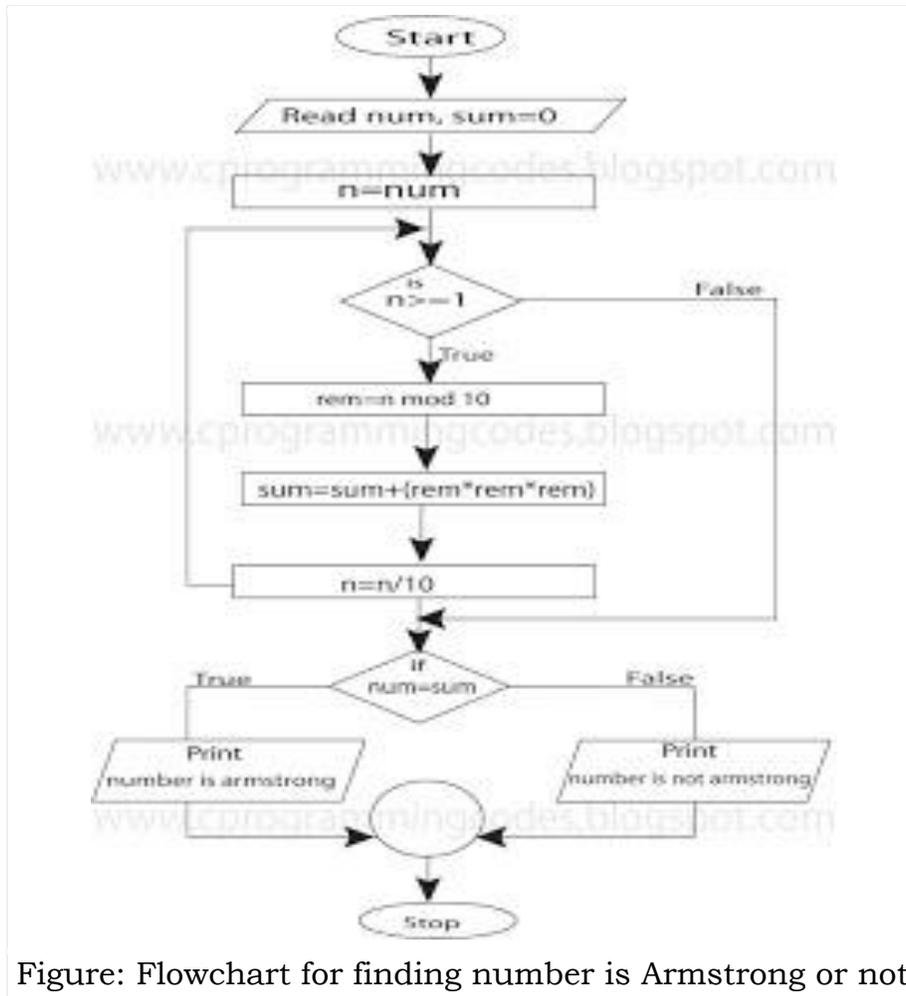


Figure: Flowchart for finding number is Armstrong or not

### Algorithm for amstrong number program

1. start
2. [read number num]  
Input num
3. [initialize sum ]  
Sum = 0
4. [initialize temperory variable n]  
n = num
5. repeat steps 6 to 8 while n >= 1
6. rem = n % 10
7. sum = sum + rem \* rem \* rem
8. n = n / 10
9. if (sum = num)  
Print "number is amstrong"

Else

Print "number is not amstrong"

10 stop

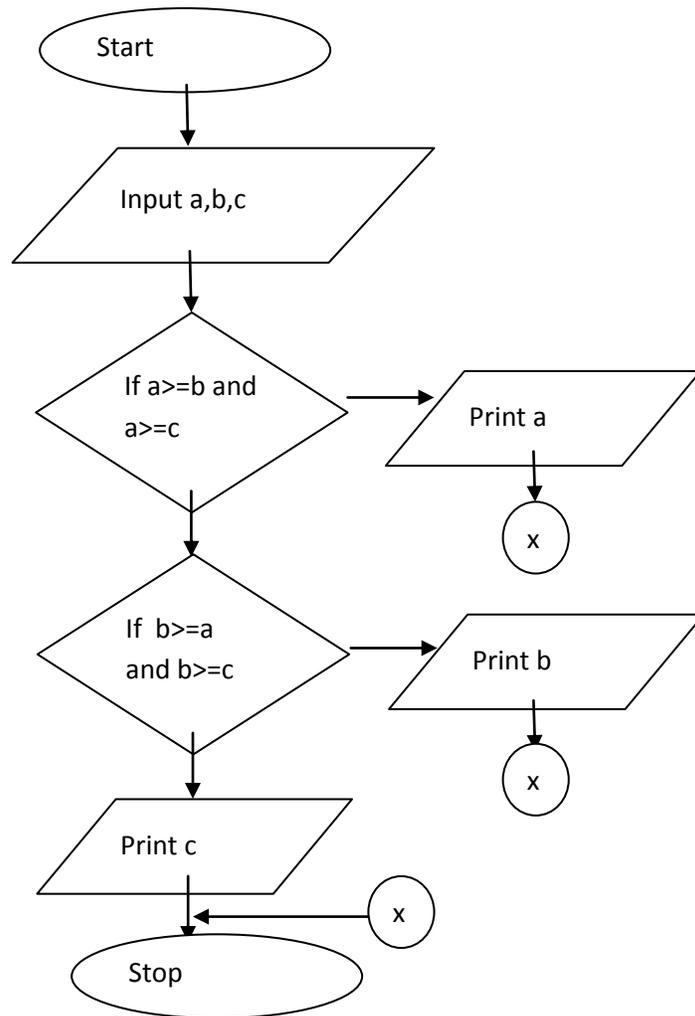
**19. Draw Flowchart and write Algorithm for finding largest of 3 numbers a,b,c**

**C program:**

```
#include <stdio.h>
void main()
{
    int a,b,c; //variables a , b and c will store three integers
    // input three integers
    printf("enter three integers ");
    scanf("%d %d %d" , &a , &b , &c);

    // now check the largest
    if (a >= b && a >= c)
        printf("%d" , a);
    else
        if ( b >= a && b >= c)
            printf("%d" , b);
        else
            printf("%d" , c);
}
```

**Flowchart to find largest of 3 numbers a,b,c**



**Algorithm to find largest of 3 numbers a,b,c**

1. start
2. [read 3 numbers a,b,c]  
Input a,b,c
3. if ( a >= b AND a >= c)  
Print a  
Goto step 6
4. if (b >= a AND b >= c)

Print b

Goto step 6

5. print c

6. stop

**20 Predict the output of following**

i)

```
void main()
{
    int x=1 , y=5;
    printf("%d" , ++(x+y));
}
```

Output

This program will give error at ++(x+y). This is because ++ operator cannot be applied to expression like (x+y). For example ++x or ++y will be correct but ++(x+y) will be error.

ii)

```
void main()
{
    int ch = 10;
    switch(ch)
    {
        default : printf("No is greater than 9");
        case 1 :
        case 3 :
        case 5 :
        case 7 :
        case 9 : printf("%d is odd" , ch);
                break;
        case 2 :
```

```
case 4 :  
case 6 :  
case 8 : printf("%d is even" , ch);  
        break;  
}  
}
```

### Output

No is greater than 9  
10 is odd

### Explanation

In this program default is written at the beginning of switch. Hence the default case executes first which prints "No is greater than 9". The default case does not have break. Hence next cases case 1 to case 9 runs and prints 10 is odd. After this break takes the control out of switch and program ends.

iii)

```
void main()  
{  
printf("%e\n" , 1234.56);  
printf("%4.0f\n" , 1234.56);  
}
```

### Output

1.23456e+03

1235

### Explanation

%e format is used for printing floating point numbers in exponential form. The output 1.23456e+03 means  $1.23456 * 10^3$ .

%4.0f means print floating pointing number such that 4 digits before the decimal point are printed and zero digits are printed after decimal point.

iv)

```
void main()
{
int i=1,j=1;
for(;;)
{
if (i>5)
break;
else
j+=i;
printf("%d\n" , j);
i+=j;
}
}
```

Output

2

5

v)

```
void main()
{
int c = 5;
while(c)
{
printf("%d " , c);
c = c - 2;
}
}
```

Output

5 3 1 -1 -3 -5 -7 -9 .....

the while loop is an infinite loop.

Initially c = 5. Then the program checks while condition

while(c)

which is same as

while(5)

value 5 is treated as true. Hence while has a true value(Actually any non zero value is true. And 0(Zero) is false.

vi)

```
int main()
{
int x = 3, y = 5 , z = 7, w;
w = x % y - y % x - z % x - x % z;

printf("%d\n" , w);

w = x / z + y / z + (x+y) / z;

printf("%d\n" , w);

w = x / z + y / z + x * y / z;

printf("%d\n" , w);

w = x % y % z + z % y % (y % x);

printf("%d\n" , w);

w = x/y/y/x + z/y/(y/x);

printf("%d\n" , w);

return 0;
}
```

Output

```
-3
1
2
3
1
```

vii)

How many times this while loop will repeat. That is how many times will x's be printed

```
int i = 5;
while (i-- > 0)
printf("x");
```

Ans

The while loop will repeat 5 times. That is x will be printed 5 times.

viii)

How many times will x's be printed

```
int i = 5;
while (i-- > 0)
printf("x");
printf("x");
```

Ans

x will be printed 6 times.

ix)

```
int main()
{
float s = 1.0;
int a = 4;
while(a <= 10)
{
s = a * 1.2;
printf("%f" , s);
}
return 0;
}
```

Output

4.8000004.8000004.800000 .....

The program will print 4.800000 for ever. The while loop is a for ever loop.

x)

```
void main()
{
int a,b,c;
a = 34;
c = 7;
a -= --a - a--;
b = --c + c + c--;
printf("%d\n%d\n%d\n%c\n" , a,b,c, 'b'-a);
}
```

Output

32

18

5

B

xi)

```
void main()
{
int x = 1 , y = 5;
printf("%d\n" , ++x+y);
}
```

Output

7

xii)

```
void main()
{
int a = 1 , b = 2 , c = 3 , d = 4.75 , x;
x = ++a + b++ * ++c % d++;
printf("%d %d %d %d\n" , a , b , c , x);
}
```

Output

2 3 4 2

xiii)

```
void main()
{
int x = 1;
printf(“%d %d %d\n”, x , (x=x+2) , (x<<2) );
printf(“%d %d %d\n” , ++x , x++ , ++x );
}
```

Output

3 3 4

6 4 4

xiv) Select the correct answer

a)

```
i = 10;
do
{
do something
}
while ( i < 10);
```

- i) do something will execute atleast once
- ii) do something will not execute at all
- iii) do while loop is not a valid loop
- iv) none of the above

Ans

- i) do something will execute atleast once

xv) select correct answer

a)

```
int a = 29 , b = 10;
```

```
float c;
```

```
c = (float) a/b;
```

the correct value of c is

i) 2.9

ii) 2

iii) 3

iv) compiler dependent

Ans

i) 2.9

xvi) State the o/p

```
void main()
```

```
{
```

```
int k = 3 , l = 4 , m;
```

```
m = ++k + l--;
```

```
cout << "value of m " << m << endl;
```

```
m = k++ + --l;
```

```
cout << "value of m " << m << endl;
```

```
}
```

o/p

value of m = 8

value of m = 6

xvii)

Q 1 State the output

a)

```
void main()
```

```
{
```

```
int a = 3;
```

```
cout << " " << a << endl;
```

```
cout << " " << a++ << endl;
```

```
cout << " " << ++a << endl;
}
o/p
3
3
5
```

b)

```
void main()
{
int x = 4, y = 9 , z;
z = (x++) + (--y) + y;
cout << "value = " << z << endl;
z = (--x) + x + (y--);
cout << "value = " << z << endl;
}
o/p
value = 20

value = 16
```

c)

```
void main()
{
int a , b, c;
a = 2 , b = 5 , c = 10;
cout << "value = " << (a+b* --c) << endl;
cout << "value = " << (-c/b * c - a) << endl;
cout << "value = " << (-a + ++b%a) << endl;
}
o/p
value = 47

value = -11

value = -2
```

xviii) State the output

```
void main()
{
int b = 3 , a = 2 , ab = 4;
int i;
int in = '2' * '2';
char ch = 'c';
printf("%c %c\n" , ch , ++ch );
printf("%d %d %d\n" , a , a , ++a);
printf("%d %d %d\n" , b , b , (++b));
printf("%d %d %d\n" , ab , ab , (++ab));
printf("%d %d\n" , a , (!a) );
}
```

o/p

d d

3 3 3

4 4 4

5 5 5

3 1

## Important programs

### Topic

#### Control structures

1. WAP to print 20 fibonacci numbers
2. WAP to print reverse of a number
3. WAP to check whether a number is prime or not
4. WAP to check whether number is amstrong number or not.
5. WAP to print multiplication table of a number.
6. WAP to print pattern. (Pg 104) (Any pattern solved in notes and note book may be asked). Here are some more pattern programs given

<p>Q6a WAP to print following o/p</p> <pre> 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5                     </pre>	<pre> #include &lt;stdio.h&gt; void main() { int x , y; for (x = 1 ; x &lt;= 5 ; x ++ ) { for (y =1 ; y &lt;= x ; y++) { printf(“%3d” , x); } }  printf(“\n”); } }                     </pre>
<p>Q6b WAP to print following o/p</p> <pre> 1 2 3 4 5 6 7 8 9 10                     </pre>	<pre> #include &lt;stdio.h&gt; void main() { int x , y , z=1; for (x = 1 ; x &lt;= 4 ; x ++ ) { for (y =1 ; y &lt;= x ; y++) { printf(“%3d” , z); z++; } }  printf(“\n”); } }                     </pre>
<p>Q6c WAP to print following o/p</p> <pre> 1 3    5 7    9    11    13 15   17   19   21   23                     </pre>	<pre> #include &lt;stdio.h&gt; void main() { int x , y , z = 1; for (x = 1 ; x &lt;= 4 ; x ++ ) { for (y =1 ; y &lt;= x ; y++) { printf(“%3d” , z); z = z + 2; } }  printf(“\n”); } }                     </pre>

<p>Q6d WAP to print following o/p</p> <pre>P P Q P Q R P Q R S P Q R S T</pre>	<pre>#include &lt;stdio.h&gt; void main() { char x , y; for (x = 'P' ; x &lt;= 'T' ; x ++ ) { for (y ='P' ; y &lt;= x ; y++) { printf("%3c" , y); } printf("\n"); } }</pre>
<p>Q6c WAP to print following o/p</p> <pre> 1 1 2 A 1 2 3 A B 1 2 3 4 A B C 1 2 3 4 5 A B C D</pre>	<pre>#include &lt;stdio.h&gt; void main() { int x , y , s = 40; char t , w = 'A'; for (x = 1 ; x &lt;= 5 ; x ++ ) { /*print spaces */ for (y=1 ; y &lt;= s ; y++) { printf(" "); }  for (y =1 ; y &lt;= x ; y++) { printf("%3d" , x); }  /*print characters only if it is not first line */ if (x != 1) { for (t = 'A' ; t &lt;= w ; t++) { printf("%3c" , t); }  w++; }</pre>

	<pre>} printf("\n"); s = s - 3; } }</pre>
--	---

Q 7 various programs given from pg 27 (nested if else and switch case ) & pg 38 on (nested if else and switch)

Q 8 WAP to print factorial of an integer.

Q 9 WAP to find sum of following series

$$S = 1/1 + 1/2 + 1/3 + \dots + 1/n$$

Q 10 WAP to find sum of following series

$$S = 1/2 + 3/4 + 5/6 + \dots + (\text{add } n \text{ such terms})$$

Q 11 Menu driven program

Q 12 WAP to print GCD and LCM of 2 integers m and n

### **Topic**

#### **Arrays (Single dimension) (Pg 79)**

1. WAP to sort array of n integers in increasing order
2. WAP to shift all elements of array one place to left.
3. WAP to shift all elements of array one place to right.
4. WAP to find sum and average of all numbers stored in array.
5. WAP to find mean and standard deviation of all numbers stored in array.

### **Topic**

#### **Matrices (Double dimension) (All these programs are solved in note book or pg 92)**

1. WAP to add 2 matrices of size m x n
2. WAP to subtract 2 matrices of size m x n
3. WAP to multiply 2 matrices of size r1 x c1 and r2 x c2

4. WAP to find transpose of matrix of size  $m \times n$
5. WAP to print diagonal elements of a square matrix.
6. WAP to print upper and lower triangle elements of a square matrix.
7. WAP to add all diagonal elements of a square matrix.

### **TOPIC**

#### **String (Array of characters) (Pg 63)**

1. WAP to check whether a string is palindrome or not (solved in note book)
2. WAP to count how many vowels, consonants and other characters are there in the sentence.
3. WAP to count number of words in the sentence. (solved in note book)
4. WAP to count how many alphabets, digits, spaces and other characters are there in the sentence. (solved in note book)
5. WAP to convert a string in such a way that first letter of every word is capitalized and remaining letters are in lower case. For example if the input is "mumbai is great" then the o/p should be "Mumbai Is Great"

### **TOPIC**

#### **Recursion (Pg 110)**

1. Write a recursive function to return factorial of a integer.
2. Write a recursive function to return sum of first  $n$  integers.(note book)
3. Write a recursive function to return value of  $x^n$ .
4. Write a recursive function to return value of  $x * y$ .
5. Write a recursive function to return value of  $n$ th fibonacci number.
6. Write a recursive function to return gcd of two numbers  $m$  &  $n$ . gcd function is defined as  
$$\text{Gcd}(m,n) = \text{gcd}(n,m) \text{ if } m < n$$
$$\text{Gcd}(m,n) = m \text{ if } n = 0$$
$$\text{Gcd}(m,n) = \text{gcd}(n, m\%n) \text{ other wise}$$
(solved in note book)

## **TOPIC**

### **Structures (pg 68)**

1. WAP to store information of n employees. Every employee has employee number , name and salary. After storing information find those employees whose salary is greater than 10000.
2. WAP to store information of n students. Every student has roll number , name and marks in one subject. After storing information find those students whose marks are greater than 75.
3. WAP to store information of n employees. Every employee has employee number , name and salary. After storing information sort all employees in increasing order of salary. Print all sorted employees in tabular format.(solved in note book)
4. WAP to store information of n students. Every student has roll number , name and marks in one subject. After storing information sort all students in decreasing order of marks. Print all sorted students in tabular format. (solved in note book)
5. WAP to store information of n employees. Every employee has employee number , name and salary. After storing information sort all employees in alphabetical order of their names. Print all sorted employees in tabular format.(hint we have to use strcmp function to compare names)
6. WAP to store information of n cricket players. Every player has name, country, number of matches and batting average. After storing information sort all players in according to country names. Print all sorted palyers in tabular format.
7. WAP to find sum of 2 complex numbers of the form  $a+bi$
8. WAP to find addition of 2 points in XY plane.

## **TOPIC**

### **POINTERS (Pg 73)**

1. Write a function to swap value of any two integer variables. (This question is solved in the note book)
2. Write a note on call by value and call by reference (address) page number 76