INTRODUCTION TO ‘C’

- C is a general purpose structured programming language that is powerful, efficient and compact.
- C is an offspring of the “basic combined programming language” called B.
- This was modified by Dennis Ritchie and was implemented in the Bell laboratories in 1972.
- C is a structured programming language. C combines the features of a high –level language with the elements of the assembler.
- This flexibility allows C to be used for system programming.
- A number of implementation of C are available like Microsoft C, Quick C, Turbo C, Lattice C, and Aztec C.
- C has large number of operators and relatively small instruction set.
- It allows user to write library functions of its own.

C program is a collection of one or more functions.

Function is a module that contains one or more C statements and performs one or more tasks.

C-tokens are identifiers, constants, variables, keywords, strings, operators, special symbol.

Alphabets, numbers and special characters are used to construct constants, variables and keywords.

- **Identifiers** are name of variables, functions and arrays. They are combination of alphabets and an underscore, no spaces are allowed.
- **Constants** is a quantity that does not change. Constant can be stored at a location in the memory of the computer. E.g. 15.5, -6.7
- **Variable** is a name given to the location in the memory where constant is stored. E.g. num, sum.
- **Keywords** are the words whose meaning has already been explained to the compiler. E.g. For, while, if

Data types

(Question: Explain the various data types used in C.)

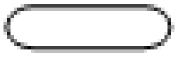
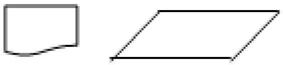
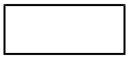
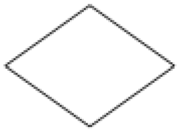
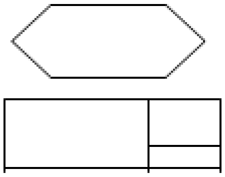
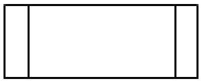
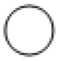
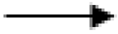
Data types available in C are integer declared as **int**, character declared as **char**, float or real declared as **float** (max upto six decimal places), long integer as **long int**, and string as **char [20]** where 20 is the length of a string.

Program is a set of instructions written in a sequence.

Flowchart is pictorial representation of the program.

(Question: Define Flowchart and draw the various symbols used.)

The following are the symbols used in flowchart.

		*START /STOP* of the program
		*INPUT /OUTPUT BOX*
		*ASSIGNMENT BOX , NEXT (FOR LOOP)*
		*DECISION (IF LOOP, WHILE LOOP, SWITCH CASE)* BOX
		*FOR LOOP*
		*FUNCTION CALL*
		*CONNECTOR*
		*FLOW ARROW*

Algorithm

(Question: Define algorithm.)

Algorithm is a step by step logic of a program written in simple English. It is independent of any programming language.

Features of C programming

(Question: List Features of C Programming)

1. It is a high-level language.
2. It follows top down approach.

3. It uses structures and pointers.
4. Unique feature of C is the preprocessor.

(Question: Give the structure of a C program)

Simple structure of a C program

```
#include<stdio.h>
main( )
{
    _________
    _________
    body of C program
    _________
    _________
 }
```

Explanation

- **#** is a **preprocessor directive** which follows simple syntax rules.
- Preprocessor as the name implies is a program that processes the source code before it passes through the compiler .
- It operates under the preprocessor command lines or directives.
- They all begin with the symbol **#** in the first column and does not end with a semicolon.
- The commonly used directives are **# include** and **# define**.

Directive	Function
#include	Specifies the file to be included.
#define	Defines a macro substitution.
#undef	Undefines a macro.
#ifdef	Tests for a macro definition.
#endif	Specifies the end of #if
#ifndef	Tests if a macro is not defined.
#if	Tests a compile time condition.
#else	Specifies alternatives when #if test fails.

- **stdio** : standard input output file
- **conio** : console input out file
- **.h** : header file.
- The **stdio.h** is an abbreviation for standard input –output header file.
- #**include<stdio.h>** tells the compiler to search for a file name stdio.h and place its content at the particular point in the program.
- The contents of the header file become part of the source code when it is compiled.
- The basic two functions are **printf** and **scanf**.

GETTING STARTED WITH C PROGRAMMING

```
#include<stdio.h>
#include<conio.h>
main( )
{
  clrscr( );
  printf(“Welcome to C programming”);
}
```

Explanation

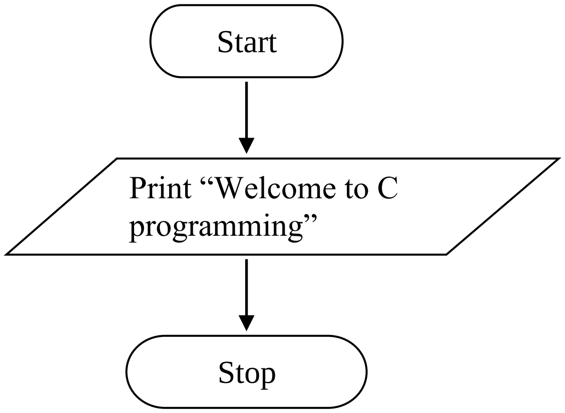
- Above written is a simple C program to print Welcome to C programming.
- Stdio.h is necessary as it has the functions **printf** in it.
- All the statements of C should end with a semicolon except #, main( ), { }, and all the loops.
- The **clrscr( )** is used to clear the screen before printing anything on it.
- Whatever is written inside the **printf** statement is printed as it is on to the output screen.

To save the file → **F2**
To compile the program → **Alt + F9**
To compile and run the program → **Control + F9**
To see the output → **Alt + F5**

The **output** of the above program is

Welcome to C programming

The flowchart is:



(Question: Write a C program to find the sum of two numbers and draw its flowchart)

```
#include<stdio.h>
#include<conio.h>
main( )
{
  int a = 10 , b = 20 , sum ;     //assignment statement
  sum = a + b ;
  printf(" the sum of two numbers is ➔ %d " , sum);
  getch( );
}
```

Explaination

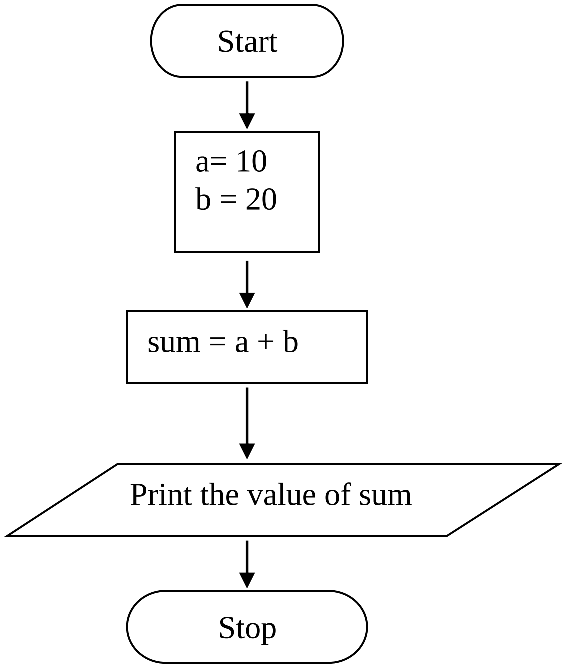
- In the above program a is assigned the value 10 and b is assigned the value 20 .
- The sum is declared as an integer variable to hold integer value.
- Then we calculate sum by adding the two values.
- To print the value of sum we introduce a new term called **format specifiers or data qualifiers** as **%d** for integer, **%f** for float, **%c** for character, **%s** for string, **%ld** for long integer.
- In this where %d is written the value of sum will be printed.

- ➢ **getch( )** is used to accept a single character from the keyboard i.e. after printing the output on the screen it will wait for the user to enter the value.

The **output** is

The sum of two numbers is ➔ **30**

The flowchart for the above program is:



Note : In the above program we have introduced assignment statement, **getch( )** and comment statements i.e. anything which is written between the // or /*…..*/ is an comment.

The above program can be modified from the assignment statement to the user's choice program, i.e. the two numbers can be entered from the keyboard.

(Question: WAP to add two numbers entered by the user.)

```
#include<stdio.h>
#include<conio.h>
main( )
{
        int a ,b, sum;
        printf(“Enter the first number➔”);
        scanf(“%d”,&a);
        printf(“\n Enter the second number ➔”);
        scanf(“%d”,&b);
        sum = a + b;
        printf(“\n The sum of two numbers %d and %d is ➔ %d “ ,a ,b, sum);
        getch( );
}
```

Explaination :

- The general syntax of the **scanf** statement whish is known as an input statement is

  ```
  scanf(“control string “, &variable 1, variable2);
  ```

- The control string contains the format of data being received.
- The symbol ‘**&‘** known as an **ampersand** before each variable specifies the variable name’s address.
- We must used this before an input variable otherwise unexpected result occur.
- The other new symbol introduced is **‘\n’** which is a new line character. It allows what ever written after it to move to the next new line.
- In the third **printf** statement the %d signs are replaced by their respective values.

Imp: Integer value has a limitation that is –32768 to +32767.

The **output** is:

```
Enter the first number➔ 30
Enter the second number ➔ 49
The sum of two numbers 30 and 49 is ➔ 79
```

Question : Draw the flowchart for the above program .

Practical problems

Also draw the flowchart.

1. Write a program to print your name, father’s name, and address, telephone number each on separate line using simple **printf** statement.
2. Write a program to calculate simple interest.

$$\text{Interest} = \frac{\text{principal * no. of years * rate of interest}}{100}$$

3. Write a program to convert the Fahrenheit temperature to Celsius.

$$\text{Celsius} = \frac{(\text{Fahrenheit} – 32.0)}{1.8}$$

4. Write a program to find sum and average of three numbers.

DECISION MAKING AND BRANCHING STATEMENT

(Question: Explain decision making and control structure of C)

- Programs are normally executed sequentially in the order in which they appear.
- This is true if no repetition or decision making is necessary.
- But number of times we need to change the sequence of program depending upon the requirement, or we need to repeat some statements number of times.
- C language allows the user for decision making and change the sequence of the program using control or decision making statements like :
  - ✓ **if** statement .
  - ✓ **switch** satetement .
  - ✓ **break** ..
  - ✓ **conditional** operators
  - ✓ **goto** statement.

if statement

(Question: Give the Syntax of if statement)

- if statement is used where we need to choose between the two things depending upon a particular value . It holds the value either true or false or yes or no . This is fur divided into
  - ✓ **simple if** statement
  - ✓ **if ….else** statement
  - ✓ **nested if …else** statement
  - ✓ **else if ladder** statement

simple if statement

The *syntax* of is

```
if ( condition )
{
statements ;
}
```

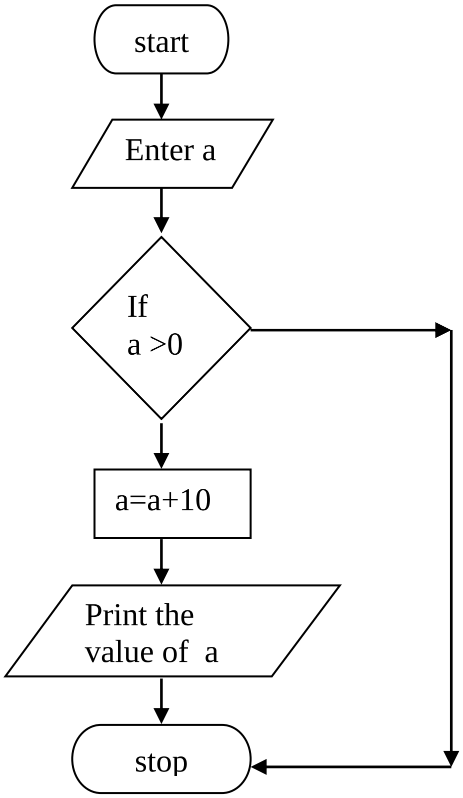
(Question: Write a program and draw its flowchart to add 10 to a number if its value is more than 0 and print it.)

```
#include <stdio.h>
#include<conio.h>
main( )
{
        int a ;
        clrscr( );
        printf(“Enter a number → “) ;
        scanf (“%d”,&a);
        if (a > 0 )
        {       a = a + 10 ;
                printf (“ the value of  a is ➔ %d “, a);
        }
        getch( );
}
```

The **output** is :

```
Enter a number → 45
the value of  a is ➔ 55
```

The flowchart for the above program is



Explaination :

- In the above program we enter a number and store in the variable a , then check whether a is greater than 0 , if the answer is yes than we add 10 to it and print the result .
- But if it does'nt satisfy the condition than it terminates the program and come out .

if …else statement

(Question: With example explain if- else statement)

```
if (condition)
{
     true block statement;
}
else
{
      false block statement ;
}
```

(Question: Write a program to find the largest of two numbers.)

```
#include<stdio.h>
#include<conio.h>
main( )
{
        int a , b ;
        clrscr( );
        printf(“ Enter the two numbers ➔”);
        scanf(“ %d  %d “,&a ,&b);
        if ( a > b )
           printf(“ \n a is large %d ”,a);
        else
           printf(“\n b is large %d  “ , b);
        getch( );
}
```

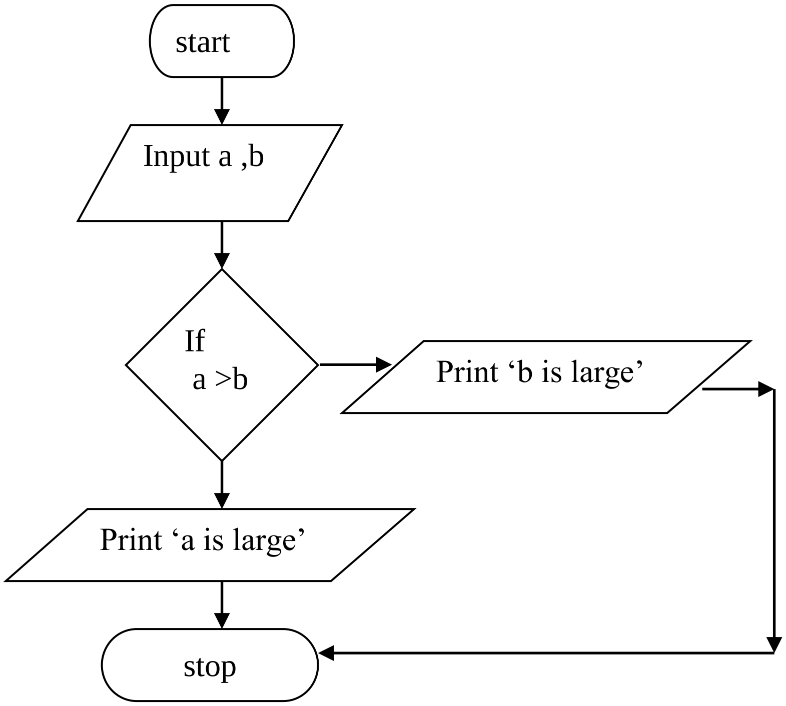
Explanation :

- In the above program we want to find the largest of two numbers.
- We input two numbers and store it in the variables a and b.
- Then with the if loop we check that whether a > b if the answer is yes or true than we print that a is large and if this statement is false than we print that the other number in this case b is large.

The **output** is :

```
Enter the two numbers ➔ 34 56
b is large 56
```

The **flowchart** for the above program is :



Nested if statement

(Question: Explain nested if with example.)

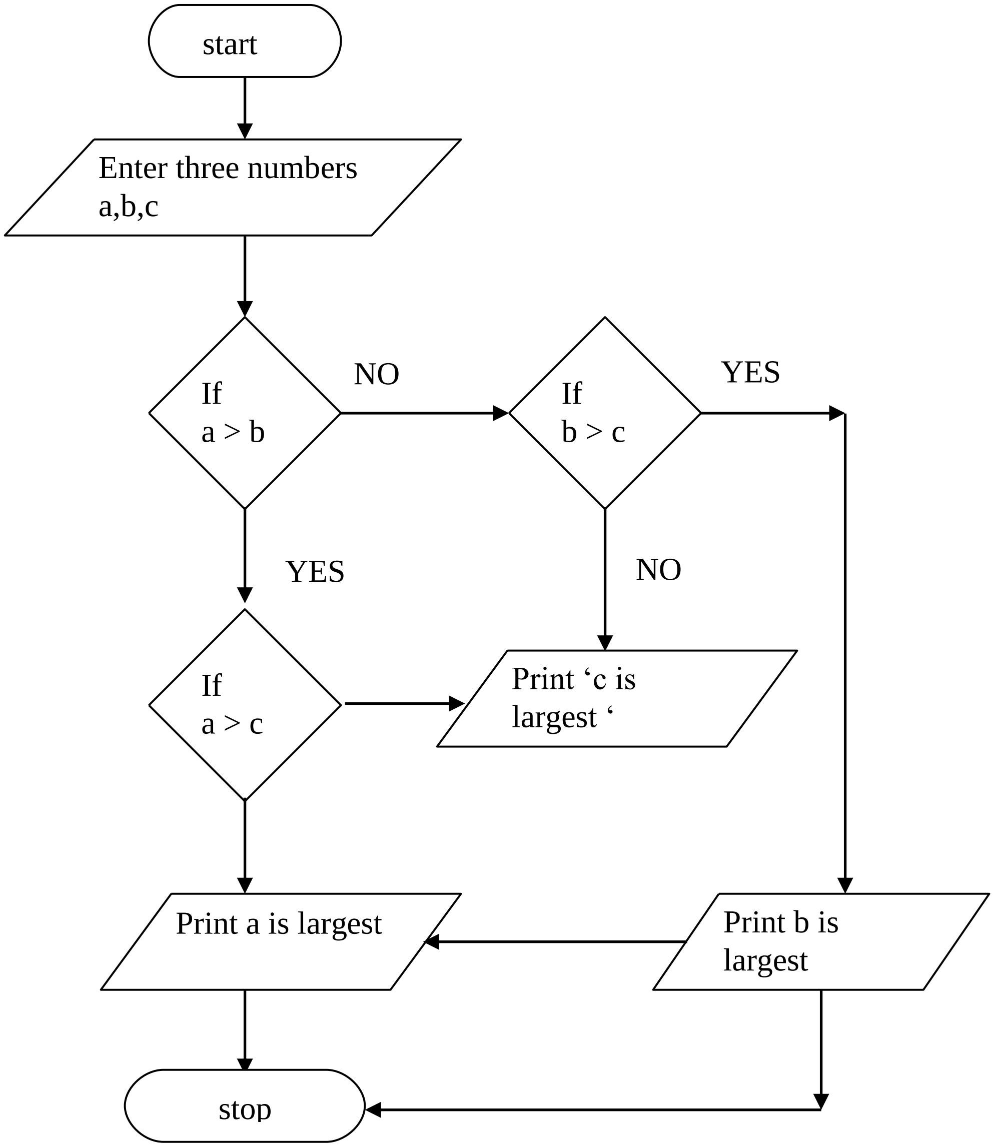
- If we need to make multiple choices then the nested if statement is used.
- If the above program is modified for the largest of three numbers than the program becomes as given below and the syntax of the nested is

```
if ( condition )
{
        if(condition)
        {
          statement ;
        }
        else
        {
          statement ;
        }
}
else
{
        if(condition)
        {
          statement ;
        }
        else
        {
          statement ;
        }
}
```

(Question: Write a program to find the largest of three numbers using if else.)

```
#include<stdio.h>
#include<conio.h>
main( )
{
        int a,b,c;
       clrscr( );
        printf( “\n Enter three numbers ”);
        scanf(“%d  %d  %d” ,&a,&b,&c);
        if ( a < b)
        {
                if( a > c)
                {
                        printf(“\na is largest of three numbers %d %d  %d        %d”,a,b,c,a);
                }
                else
                {
                        printf(“\nc is largest of three numbers %d %d  %d  %d”,a,b,c,c);
                }
        else
        {
                if (b > c)
                {
                        printf(“\nb is largest of three numbers %d %d  %d %d”,a,b,c,b);
                }
                else
                        printf(“\nc is largest of three numbers %d %d  %d  %d”,a,b,c,c);
        }
        getch( )
}
```

The **flowchart** is:



Explanation:

- The above program finds the largest of three numbers. as we input three numbers a , b, c and check whether a is larger than b
- if the answer is yes or true than if check whether a is larger than c than we print that a is largest of three numbers
- otherwise the number which is stored in the variable c is largest of three numbers
- but if the first condition fails than we check that if b is larger than c if yes than we print that b is largest of the three numbers else c is the largest .

The **output** of the above program is :

```
Enter three numbers  15 34 23
b is largest of three numbers 15 34 23  34
```

ladder else if statement

(Question: Explain ladder if else statement)

```
if ( condition )
{
        statement ;
}
else if (condition )
    {
        statement ;
    }
    else if(condition )
         {
            statement ;
         }
```

SWITCH CASE STATEMENT

(Question: Explain the concept of switch case)

The use of if …else becomes tedious if there are lot many comparisons , hence a new statement called switch case came into picture . The **syntax** is

```
switch (expression)
{
  case  option –1 :
                        statements;
                       break;

  case  option –2 :
                        statements;
                       break;
   ……..
   ……..
  default :
                        statements;
                       break;
}
```

The programs written using switch case are known as **menu driven programs**.

(Question: Write a program to add, multiply, subtract and divide a number using a switch case statement.)

```
#include<stdio.h>
#include<conio.h>
main( )
{
        int num1,num2,n=0;
        float ans;
        int choice ;
        printf(“Enter two numbers  “);
        scanf(“%d %d”,&num1,&num2);
        printf(“1:Add”);
        printf(“2:Sub”);
        printf(“3:Mul”);
        printf(“4:Div”);
        printf(“Enter your choice  “);
        scanf(“%d”,&choice);
        switch (choice)
        {
                case 1:    ans = num1+num2;
                            n=1;
                            break;
                case 2:    ans = num1-num2;
                            n=1;
                        break;
                case 3:    ans = num1*num2;
                            n=1;
                        break;
                case 4:    ans = num1/num2;
                            n=1;
                        break;
                default: printf(“u have entered wrong choice”)
                          break;
        }
        if ( n==1)
                printf(“the result is %f”,ans);
        getch( );
        }
```

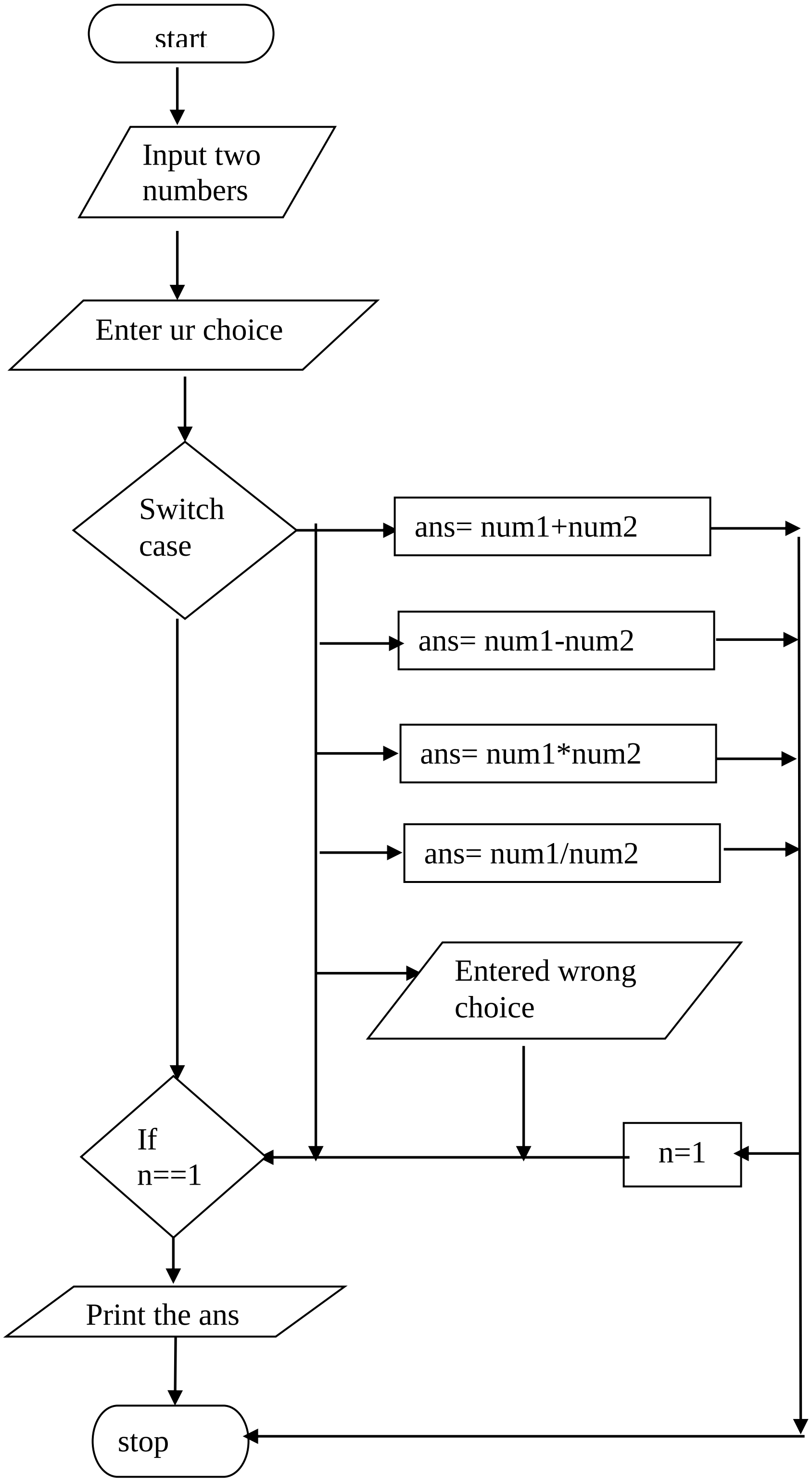
Explaination :

- In the above program instead of using if..else for the choices we use switch case statement which reduces the ambiguity of looping continuously.

The **output** is:

```
Enter two numbers  45 67
1:Add
2:Sub
3:Mul
4:Div
Enter your choice  1
the result is  112
```

The **flowchart** is:



Write a program using switch case to print whether the given number is even or odd,to print the factorial of a number .

BREAK statement:

- The break statement is used anywhere in the program to break the flow of the program and to come out of the particular loop.

The ? : operator:

- This operator is popularly known as **conditional operator** . This is useful for making two way decision .
- The general **format** is
  Conditional expression ? expression-1 :expression-2;
- In this conditional expression is evaluated first and if the result is true than the expression-1 is returned as the value of the conditional expression otherwise expression-2 is returned as the value of conditional operator.

Example : to check the largest of two numbers

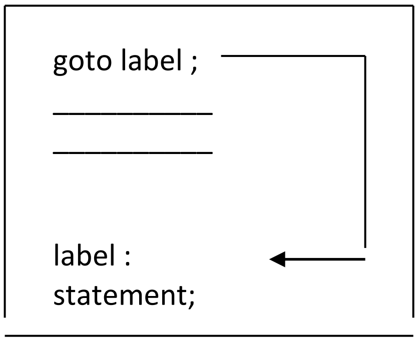
large = (a > b) ? a : b ;

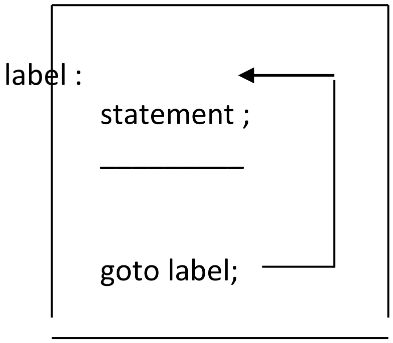
- In the above C statement it checks if the value of a is greater than b , if the answer is true than the variable large gets the value of a otherwise it gets the value of b.
- This statement is a shortcut method of if ..else statement .

goto Statement

(Question: Explain the goto statement)

- This is an unconditional branch statement .
- This is used to jump in the program from one statement to another without any condition .
- The goto requires an label to identify the place it wants to branch to.
- A label is any valid variable name and must be followed by a colon ':' .
- The label is placed immediately before the statement where the control is to be transferred.
- The goto breaks the normal sequence of the program and is very rarely used





FORWARD JUMP

BACKWARD JUMP

Questions:

1. In what ways does the switch statement differ from a if statement .
2. Explain the conditional operator with example
3. Explain the forward and the backward jump.
4. What do u mean by a break statement , explain with example.

DECISION MAKING AND LOOPING

- The looping process should include the following steps.

  (Question: Explain the lopping process in C.)

  - Setting and initializing of the counters.
  - Test for the specified condition for the execution of the loop.
  - Execution of statements in the loop.
  - Incrementing the counter.

- The C language supports three loop statements .
  - **while** loop statemnt
  - **do** loop statement.
  - **for** loop statement.

WHILE LOOP STATEMENT

(Question: Explain the while statement with eample.)

- This is the simplest of all the loops in C language . The general format for the while loop is .

```
while ( test condition )
{
        body of the loop ;
}
```

- The while loop is an entry controlled loop , which means that if the while loop condition is true than only the control be transfered to the loop inside , otherwise it will terminate the loop or come out of the loop.
- If the condition is true than the control is transferred inside the body of the loop , and after it is executed it once again check for the condition , this process continues till the condition is not satisfied.

(Question: Write a program to find the sum of first 10 numbers.)

```
#include<stdio.h>
#include<conio.h>
main( )
{         int num = 1, sum = 0;
          clrscr( );
          while ( num < = 10)
          {
                    sum = sum + num ;
                    num = num + 1;
          }
printf (“ the sum of 10 numbers is ➔ %d “, sum );
getch( );
}
```

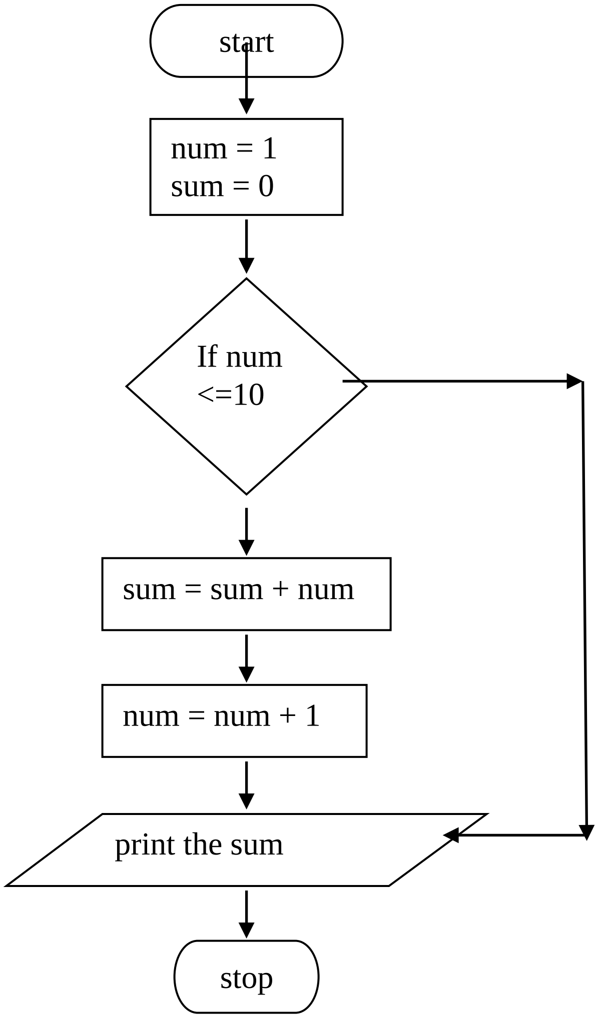
Explanation:

- The above program is used to find the sum of first 10 numbers.
- We first assign a variable called num the initial value of 1, we have to find the sum from numbers 1 to a10.
- Another variable sum is initialized to 0 to store the sum of all the numbers.
- We than check the while loop that the statements inside the loop should be executed till the value of num is more than 10.
- As initially the value of num is 1 it enters the loop and adds up the num to the variable sum and the value of num is incremented by one.
- Once again the value of num is checked in the while loop and as num is less than 10 it again enters the loop , and this procedure continued till the value of num is more than 10 .
- When the value of num is 11, it terminates the loop and print value of sum.

The **output** is

```
the sum of 10 numbers is ➔ 55
```

The **flowchart** is



Algorithm

1. Initialize the value of num to 1 and sum = 0 .
2. Check while (num <= 10) do the following
   sum = sum + num
   num = num + 1
3. Print the sum

DO LOOP STATEMENT

(Question: Explain the do while loop.)

- The while loop we studied first checks the condition and if the condition is true or if the condition is satisfied than only executes the statement in the loop else comes out of it.

- But sometimes it is required that the set of statements to be executed before the check condition , in such cases the do while loop is used .
- The syntax for it is as given below .

```
do
{
        body of loop;
}while(test condition) ;
```

(Question: Write a program to check whether the number is even or odd , the program should continue till user choses to quit .)

```
#include<stdio.h>
#include<conio.h>
main( )
{       int num ;
        char ans;
do
{       printf("Enter the number to be checked ➔");
        scanf("%d",&num);
        if (num % 2 == 0)
                printf (" \nthe number %d is even",num);
        else
                printf(" \nthe number %d is odd ",num);
        printf(" \ndo u want to continue➔ ");
        scanf ("%c ", &ans);
}while (ans =='y');
getch( );
}
```

The **output** is

```
Enter the number to be checked ➔56
the number 56 is even
do u want to continue➔y
Enter the number to be checked ➔73
the number 73 is odd
do u want to continue ➔n
```

Explanation:

- In the above progran we want to find whether the number is even or odd , and we want to check for the numbers till the user wants to quit the program for this we use a do while loop .
- We enter the number to be checked and in the if loop we check if the number divided by two the remainder is zero than the number is an even number else it is an odd number .
- then the user is asked whether the user wants to check for more numbers if the answer is 'y',then the program continues ,otherwise it comes out of the loop and program ends.

Draw the flowchart for the same .

IMP NOTE : The difference between the while loop and do ..while loop is that in the while loop it checks for the condition initially and if it is true than only the body of the loop is executed whereas in the do.. while loop the body of the loop i.e the statements are executed atleast once and then the condition is checked and if the condition is true than it re-enters the loop else comes out of it.

FOR LOOP STATEMENT

(Question: Explain the syntax of for loop)

- This is also an entry controlled loop . The general syntax for tne for loop is

```
for(initial cond ; final condition ; increment operator)
{
        body of loop;
}
```

(Question: Write a program to find the sum of first 10 numbers.)

```
#include<st dio.h>
#include<conio.h>
main( )
{
        int i , sum = 0 ;
        clrscr( );
        for( i = 1 ; i < = 10 ; i++)
        {
            sum = sum + i ;
        }
        printf("\nthe sum of first 10 numbers is ➔%d ",sum);
        getch( );
}
```

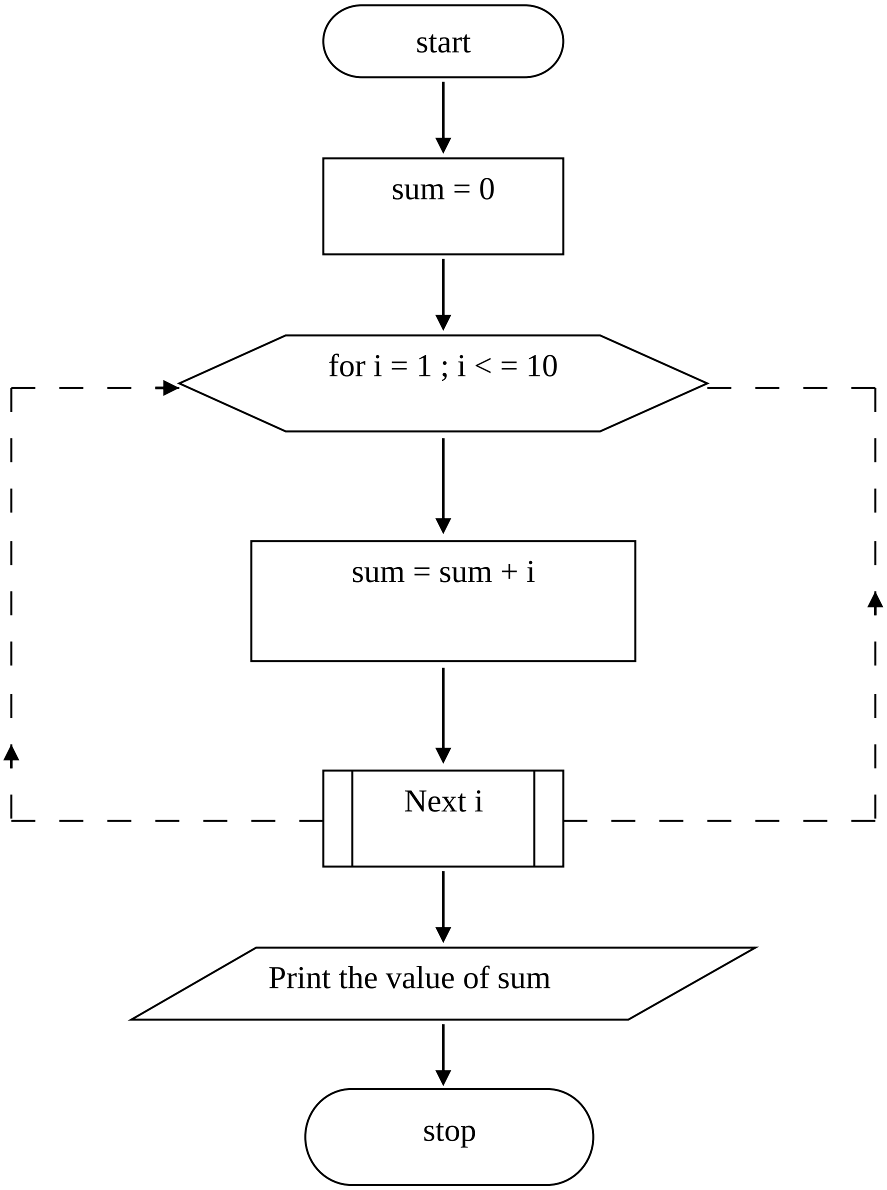
The **output** is

```
the sum of first 10 numbers is ➔55
```

Explanation:

- As we want to find the sum of first 10 numbers, so we know that our starting number is one and the last number is 10, and the number increase gradually with one.
- Sum is initialized to zero as it may take the garbage value. So the initial condition of the for loop is 1 and the termination condition or the final condition of the for loop will be 10 an the loop variable is incremented by 1. The value of sum is added to the loop variable and saved back to sum.

The **flow chart** is :



(Question: Write a program to input the10 numbers from the user and find its sum and average.)

(Question: WAP to enter 10 numbers and find the largest of the number.)

Using a nested for loop

The nested for loop is used where two or more loops are used to perform certain tasks. In ANSI C maximum of 15 levels can be used.

The syntax looks like .

```
        for ( i = 0  ; i < 5 ; i ++)
        {
                for (j = i ; i < = 5 ; j++)
                {
                        __________
                        __________
                }
        }
```

(Question: Write a program to print the following pattern)

```
        1
        2 2
        3 3 3
        4 4 4 4
        5 5 5 5 5
```

```
    #include<stdio.h>
    #include<conio.h>
    main( )
    {
            int i , j ;
            for( i = 1 ; i < = 5 ; i++)
            {
                    for ( j = 1 ; j <= i ; j ++)
                     {    printf("%d\t ",j);   }
                    printf("\n");
            }
            getch( );
    }
```

Draw the flowchart for the above program .

IMPORTANT FEATURE OF FOR LOOP

1.The most important feature of for loop is that more than one variable can be initialized in the for loop at a time .

```
                a = 1 ;
                for ( b = 0 ; b < 5 ; b ++)
```

can be written as

```
for ( a = 1 ,b = 0 ; b < 5 ; b ++)
```

- Like the initial condition all other loop variables can also have more than one variable and all of them will be separated by a **comma .**

.

2. The test condition may also have a **compound statement** in it like

```
sum = 0;
for(a = 0 ; a < 10 && sum <100 ; a++)
        sum = sum + a ;
```

3. It is also permissible to use expressions in the initialization and increment part of the loop

```
for ( a = (c+d) /2 ; a> 30 ; a = a/2)
```

4. A unique part of for loop is one or more sections can be omitted if needed

```
a= 5;
for( ; a != 100 ; )
{
   printf("%d",a) ;
   a = a+ 2 ;
}
```

- The initial and the increment conditions are omitted , but the conditions are given separately .
- If the conditions are not given then it becomes an infinite loop and can be terminated using a break or a goto statement .

5. A for loop can also be written to give a time delay and such kind of loops are called as null loop.

```
for( a = 1000 ; a > 0 ; a --);
```

- as this loop ends with a semicolon and does not have any statements in it , it will be executed for 1000 times .

Questions

1. What is the purpose of the while statement ? When is the logical expression evaluated ? What is the minimum number of times the while loop is evaluated .
2. What is the differnce between the while loop and the do..while loop . Explain with the syntax .
3. Explain the for loop and how does it differ from a while loop.

PRACTICAL

1. Write a program to print the multiplication table for ten numbers.
2. Write a program to find the sum of marks of three subjects for five students.
3. Write a program to print the Fibonacci series.
4. Write a program to print the reverse of the program 3.7
5. Write a program to find the sum of every fourth number starting from 1 till 100.

(Question: Write a program to find the factorial of a given number using while loop)

```
#include<stdio.h>
int main(){
  int i=1,f=1,num;

  printf("Enter a number: ");
  scanf("%d",&num);

  while(i<=num){
      f=f*i;
      i++;
  }

  printf("Factorial of %d is: %d",num,f);
  return 0;
}
```

Sample output:

```
Enter a number: 5
Factorial of 5 is: 120
```

(Question: Write a program to find the factorial of a given number using for loop)

```
#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;
}
```

(Question: WAP to check perfect number.)

```
#include<stdio.h>
int main(){
  int n,i=1,sum=0;

  printf("Enter a number: ");
  scanf("%d",&n);

  while(i<n){
    if(n%i==0)
       sum=sum+i;
      i++;
  }
  if(sum==n)
    printf("%d is a perfect number",i);
  else
    printf("%d is not a perfect number",i);

  return 0;
}
```

Sample output:

```
Enter a number: 6
6 is a perfect number
```

(Question: Write a program to print perfect numbers from 1 to 100.)

```
#include<stdio.h>
int main(){
  int n,i,sum;

  printf("Perfect numbers are: ");
  for(n=1;n<=100;n++){
   i=1;
   sum = 0;

   while(i<n){
    if(n%i==0)
       sum=sum+i;
      i++;
   }
```

```
    if(sum==n)
      printf("%d ",n);
  }

  return 0;
}
```

Output:
Perfect numbers are: 6 28

(Question: WAP to check the given number is Armstrong number or not using while loop.)

Those numbers which sum of the cube of its digits is equal to that number are known as armstrong numbers. For example 153 since 1^3 + 5^3 + 3^3 = 1+ 125 + 9 =153

```
#include<stdio.h>
int main(){
    int num,r,sum=0,temp;

    printf("Enter a number: ");
    scanf("%d",&num);

    temp=num;
    while(num!=0){
         r=num%10;
         num=num/10;
         sum=sum+(r*r*r);
    }
    if(sum==temp)
         printf("%d is an Armstrong number",temp);
    else
         printf("%d is not an Armstrong number",temp);

    return 0;
}
```

Sample output:
Enter a number: 153
153 is an Armstrong number

(Question: WAP to check the given number is Armstrong number or not using for loop.)

```
#include<stdio.h>
int main(){
   int num,r,sum=0,temp;

   printf("Enter a number: ");
   scanf("%d",&num);

   for(temp=num;num!=0;num=num/10){
       r=num%10;
       sum=sum+(r*r*r);
   }
   if(sum==temp)
       printf("%d is an Armstrong number",temp);
   else
       printf("%d is not an Armstrong number",temp);

   return 0;
}
```

Sample output:

```
Enter a number: 370
370 is an Armstrong number
```

Logic of Armstrong number in c

(Question: Write a program to reverse a given number.)

```
#include<stdio.h>
int main(){
   int num,r,reverse=0;

   printf("Enter any number: ");
   scanf("%d",&num);

   while(num){
       r=num%10;
       reverse=reverse*10+r;
       num=num/10;
   }

   printf("Reversed of number: %d",reverse);
```

```
  return 0;
}
```

Sample output:
Enter any number: 12
Reversed of number: 21

(Question: Write a c program to find out sum of digit of given number using while loop.)

```
#include<stdio.h>
int main(){
 int num,sum=0,r;
 printf("Enter a number: ");
 scanf("%d",&num);
 while(num){
    r=num%10;
    num=num/10;
    sum=sum+r;
 }
 printf("Sum of digits of number:  %d",sum);
 return 0;
}
```

Sample output:
Enter a number: 123
Sum of digits of number: 6

(Question: Write a c program to find out sum of digit of given number using for loop.)

```
#include<stdio.h>
int main(){
 int num,sum=0,r;
 printf("Enter a number: ");
 scanf("%d",&num);

 for(;num!=0;num=num/10){
    r=num%10;
    sum=sum+r;
 }
```

```
  printf("Sum of digits of number:  %d",sum);
  return 0;
}
```

Sample output:

```
Enter a number: 567
Sum of digits of number:  18
```

(Question: WAP to calculate power of a number.)

```
#include<stdio.h>
int main(){
  int pow,num,i=1;
  long int sum=1;
  printf("\nEnter a number: ");
  scanf("%d",&num);
  printf("\nEnter power: ");
  scanf("%d",&pow);
  while(i<=pow){
          sum=sum*num;
          i++;
  }
  printf("\n%d to the power %d is: %ld",num,pow,sum);
  return 0;
}
```

(Question: Write a c program to find largest among three numbers using binary minus operator.)

```
#include<stdio.h>
int main(){
    int a,b,c;
    printf("\nEnter 3 numbers: ");
    scanf("%d %d %d",&a,&b,&c);
    if(a-b>0 && a-c>0)
         printf("\nGreatest is a :%d",a);
    else
         if(b-c>0)
             printf("\nGreatest is b :%d",b);
         else
             printf("\nGreatest is c :%d",c);
    return 0;
}
```

(Question: Write a program to find largest among three numbers using conditional operator.)

```
#include<stdio.h>
int main(){
  int a,b,c,big;
  printf("\nEnter 3 numbers:");
  scanf("%d %d %d",&a,&b,&c);

  big=(a>b&&a>c?a:b>c?b:c);
  printf("\nThe biggest number is: %d",big);

  return 0;
}
```

(Question: WAP to print ascii value.)

```
#include<stdio.h>
int main(){
    int i;
      for(i=0;i<=255;i++)
         printf("ASCII value of character %c: %d\n",i,i);
    return 0;
}
```

(Question: WAP to find fibonacci series using for loop.)

```
#include<stdio.h>
int main(){
    int k,r;
    long int i=0l,j=1,f;

    //Taking maximum numbers form user
    printf("Enter the number range:");
    scanf("%d",&r);

    printf("FIBONACCI SERIES: ");
    printf("%ld %ld",i,j); //printing firts two values.

    for(k=2;k<r;k++){
       f=i+j;
       i=j;
       j=f;
       printf(" %ld",j);
   }

   return 0;
}
```

Sample output:

```
Enter the number range: 15
FIBONACCI SERIES: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

(Question: WAP to find fibonacci series using while loop.)

```
#include<stdio.h>
int main(){
   int k=2,r;
   long int i=0l,j=1,f;

   printf("Enter the number range:");
   scanf("%d",&r);

   printf("Fibonacci series is: %ld %ld",i,j);

   while(k<r){
       f=i+j;
       i=j;
       j=f;
       printf(" %ld",j);
        k++;
   }

   return 0;
}
```

Sample output:

```
Enter the number range: 10
Fibonacci series is: 0 1 1 2 3 5 8 13 21 34
```

Code 4:

1. **Sum of Fibonacci series in c**

```
#include<stdio.h>
int main(){
   int k,r;
   long int i=0,j=1,f;
   long int sum = 1;

   printf("Enter the number range: ");
   scanf("%d",&r);

   for(k=2;k<r;k++){
       f=i+j;
       i=j;
       j=f;
       sum = sum + j;
   }

   printf("Sum of Fibonacci series is: %ld",sum);

   return 0;
}
```

```
Sample output:
Enter the number range: 4
Sum of Fibonacci series is: 4
```

(Question: WAP to determine leap year.)

```
#include<stdio.h>
int main(){
   int year;

   printf("Enter any year: ");
   scanf("%d",&year);

   if(((year%4==0)&&(year%100!=0))||(year%400==0))
       printf("%d is a leap year",year);
   else
       printf("%d is not a leap year",year);

   return 0;
}
```

```
Sample output:
Enter any year: 2010
2010 is not a leap year
```

(Question: WAP to print prime numbers between 1-300 using break and continue.)

```
#include <math.h>
#include <stdio.h>
main(){
  int i, j;
  i = 2;
  while ( i < 300 ){
     j = 2;
     while ( j < sqrt(i) ){
        if ( i % j == 0 )
           break;
        else{
           ++j;
           continue;
        }
      }
      if ( j > sqrt(i) )
           printf("%d\t", i);
      ++i;
  }
  return 0;
}
```

(Question: WAP to find LCM with two numbers.)

```
#include<stdio.h>
int main(){
  int n1,n2,x,y;
  printf("\nEnter two numbers:");
  scanf("%d %d",&n1,&n2);
  x=n1,y=n2;
  while(n1!=n2){
      if(n1>n2)
           n1=n1-n2;
      else
      n2=n2-n1;
  }
  printf("L.C.M=%d",x*y/n1);
```

```
 return 0;
}
```

(Question: Write a c program for finding gcd (greatest common divisor) of two given numbers.)

```
#include<stdio.h>

int main(){

   int x,y,m,i;

   printf("Insert any two number: ");

   scanf("%d%d",&x,&y);
   if(x>y)
       m=y;
   else
       m=x;

   for(i=m;i>=1;i--){
       if(x%i==0&&y%i==0){
          printf("\nHCF of two number is : %d",i) ;
          break;
       }
   }
   return 0;
}
```

(Question WAP to convert string into ASCII values.)

```
#include<stdio.h>
int main(){

   char str[100];
   int i=0;
   printf("Enter any string: ");
   scanf("%s",str);
   printf("ASCII values of each characters of given string: ");
   while(str[i])
       printf("%d ",str[i++]);
          return 0;
}
```