

EXPERIMENT #3

OBJECTIVE

- Use of control flow in Programming

BASIC THEORY

Basic control flow is governed by the if...else, switch, while, do...while, and for statements

THE IF ELSE STATEMENT:

This is used to decide whether to do something at a special point or to decide between two course of action.

The following test decides whether a student has passed an exam with a pass mark of 45

```
if(result >= 45)
    printf("Pass\n");
else
    printf ("Fail \n");
```

It is possible to use the part if without the else.

```
if(temperature < 0)
    printf("Frozen\n");
```

THE SWITCH STATEMENT:

This is another form of the multi-way decision. It is well structured, but can only be used in certain cases where only one variable is tested, all branches must depend on the value of that variable. This is a function which converts an integer into a vague description.

```

estimate(number)
int number;
/* Estimate a number as none ,one ,two,
several many */
{
    switch(number) {
        case 0:
            printf("None \n");
            break;
        case 1:
            printf("One \n");
            break;
        case 2:
            printf("Two \n");
            break;
        case 3:
            printf("Several \n");
            break;
        default:
            printf("Many \n");
    }
}

```

LOOPS:

C gives you a choice of three types of loop, while, do while and for. The while loop keeps repeating an action until an associated test returns false. The do while loop is similar, but the test occurs after the loop is executed.

```
do  
{  
    statement;  
}while (expression);
```

For example:

```
int x=3;  
main()  
{  
    do {  
        printf ("x=%d\n", x--);  
    }while (x>0);  
}
```

Outputs:

```
x=3  
x=2  
x=1
```

NOTE: The postfix $x--$ operator which uses the current value of x while printing and then decrements x .

THE FOR LOOP:

The for loop works well where the number of iterations of the loop is known before the loop is entered. The head of the loop consists of three parts separated by semicolons.

- The first is run before the loop is entered. This is usually the initialization of the loop variable.
- The second is a test, the loop is exited when this returns false.

THE WHILE LOOP:

The while loop repeats a statement until the test at the top proves false. The while has the form:

```
while (expression)
{
    statement;
}
```

For example:

```
int x=3;
main()
{
    while (x>0)
        { printf ("x=%d\n",x);
          x--;
        }
}
```

Outputs:

```
x=3
x=2
x=1
```

THE DO WHILE LOOP:

This is very similar to the while loop except that the test occurs at the end of the loop body. The test then verifies the data, and loops back to read again if it was unacceptable. The loop is executed at least once after the program runs.

- The third is a statement to be run every time the loop body is completed. This is usually an increment of the loop counter.
- NOTE: C basically treats for statements as while type loops

For example:

```
int x;  
main()  
{  
    for (x=3; x>0; x--)  
        printf ("x=%d\n", x);  
}
```

Output:

```
x=3  
x=2  
x=1
```

THE BREAK STATEMENT:

It is used to exit from a loop or a switch, control passing to the first statement beyond the loop or a switch.

A break; within a loop should always be protected within an if statement which provides the test to control the exit condition.

THE CONTINUE STATEMENT:

This is similar to break but is encountered less frequently. It only works within loops where its effect is to force an immediate jump to the loop control statement.

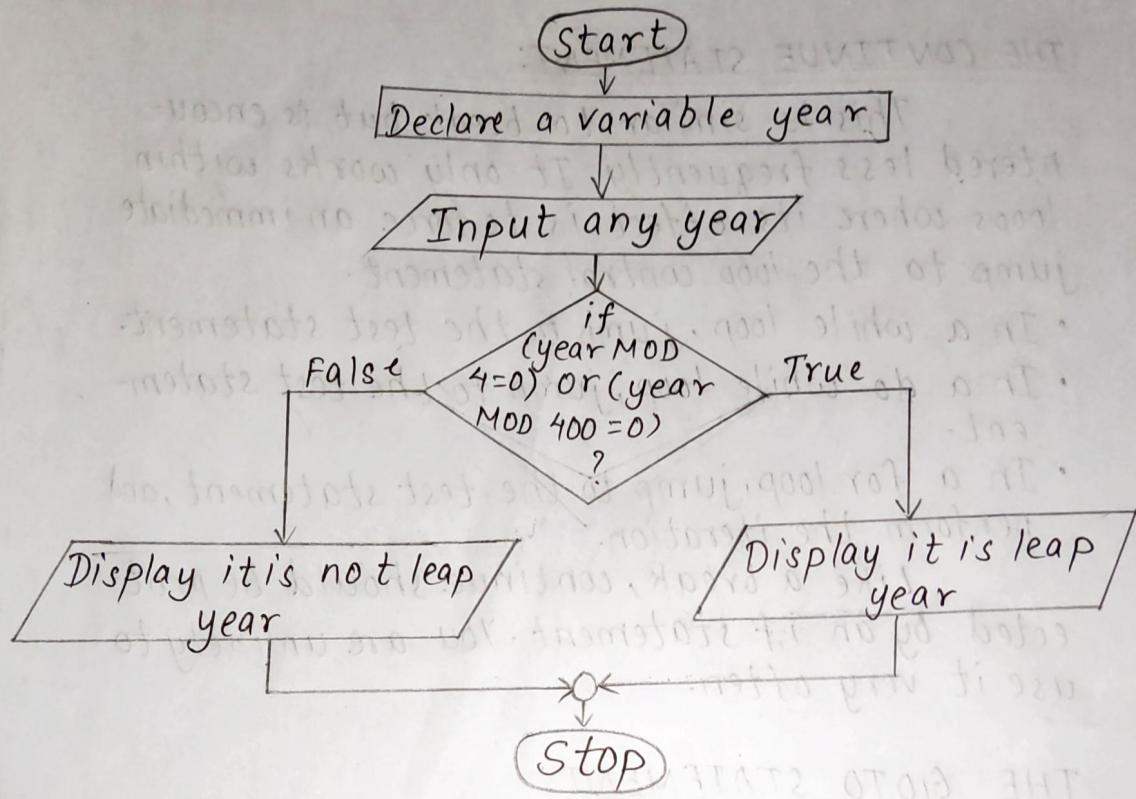
- In a while loop, jump to the test statement.
- In a do while loop, jump to the test statement.
- In a for loop, jump to the test statement, and perform the iteration.

Like a break, continue should be protected by an if statement. You are unlikely to use it very often.

THE GOTO STATEMENT:

C has a goto statement which permits unstructured jumps to be made. Its use is not recommended.

Flowchart



LAB ASSIGNMENTS :

1. An year is entered through the keyboard, write a program to determine whether it is leap or not. (Hint: For century year it should be divisible by 400 and for non century year it should be divisible by 4)

Algorithm:

Step 1: Start

Step 2: ~~Input~~ ^{declare} a year

Step 3: Input any year

Step 4: If time is exactly divisible by 400 or 4 and if time in year is not exactly divisible by 100 then,

 4.1) Display it is leap year

Step 5: Else

 5.1) Display it is not leap year

Step 6: Stop

Source code:

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int year;
    printf ("Enter any year :\n");
    scanf ("%d", &year);
    if (year % 400 == 0)
}
```

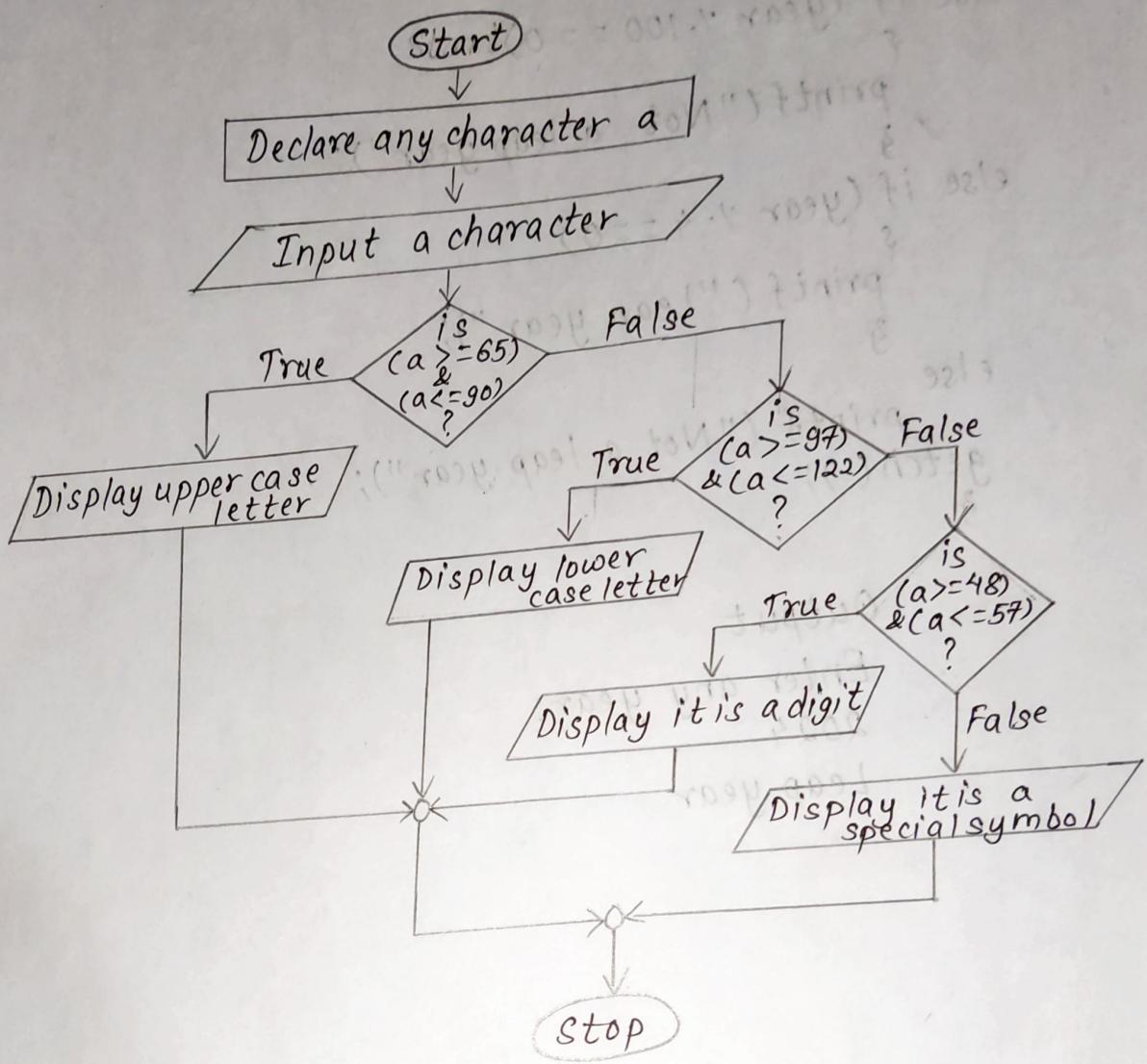
```
else if (year % 100 == 0)
{
    printf("Not a leap year");
}
else if (year % 4 == 0)
{
    printf ("Leap year");
}
else
    printf ("Not a leap year");
getch();
```

Output

Enter any year:

2024

Leap year



2. Any character is entered through the keyboard, write a program to determine whether the character entered is capital letter, small letter, a digit or a special symbol using ASCII values and built-in functions.

The following table shows the range of ASCII values:

| Characters | ASCII Values |
|-----------------|-----------------------------|
| A-Z | 65-90 |
| a-z | 97-122 |
| 0-9 | 48-57 |
| Special Symbols | 0-47, 48-64, 91-96, 123-127 |

Algorithm

Step 1: Start

Step 2: Declare a character a.

Step 3: Input a character

Step 4: If a is greater or equal to 65 and less than or equal to 90

 4.1) Display uppercase letter

Step 5: Else if a is greater than or equal to 97 and less than or equal to 122.

 5.1) Display lowercase letter

Step 6: Else if a is greater than or equal to 48 and less than or equal to 57.

 6.1) Display it is a digit

Step 7: Else

 7.1) Display it is special symbol

Step 8: Stop

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char a;
    printf("Enter any character:\n");
    scanf("%c", &a);
    printf("The %c has ASCII value %d", a, a);
    if (a >= 65 && a <= 90)
        printf("\n The entered character is capital
               letter");
    else if (a >= 97 && a <= 122)
        printf("\n Small letter");
    else if (a >= 48 && a <= 57)
        printf("\n digit");
    else
        printf ("\n Special symbols");
    getch();
}
```

Output

Enter any character:

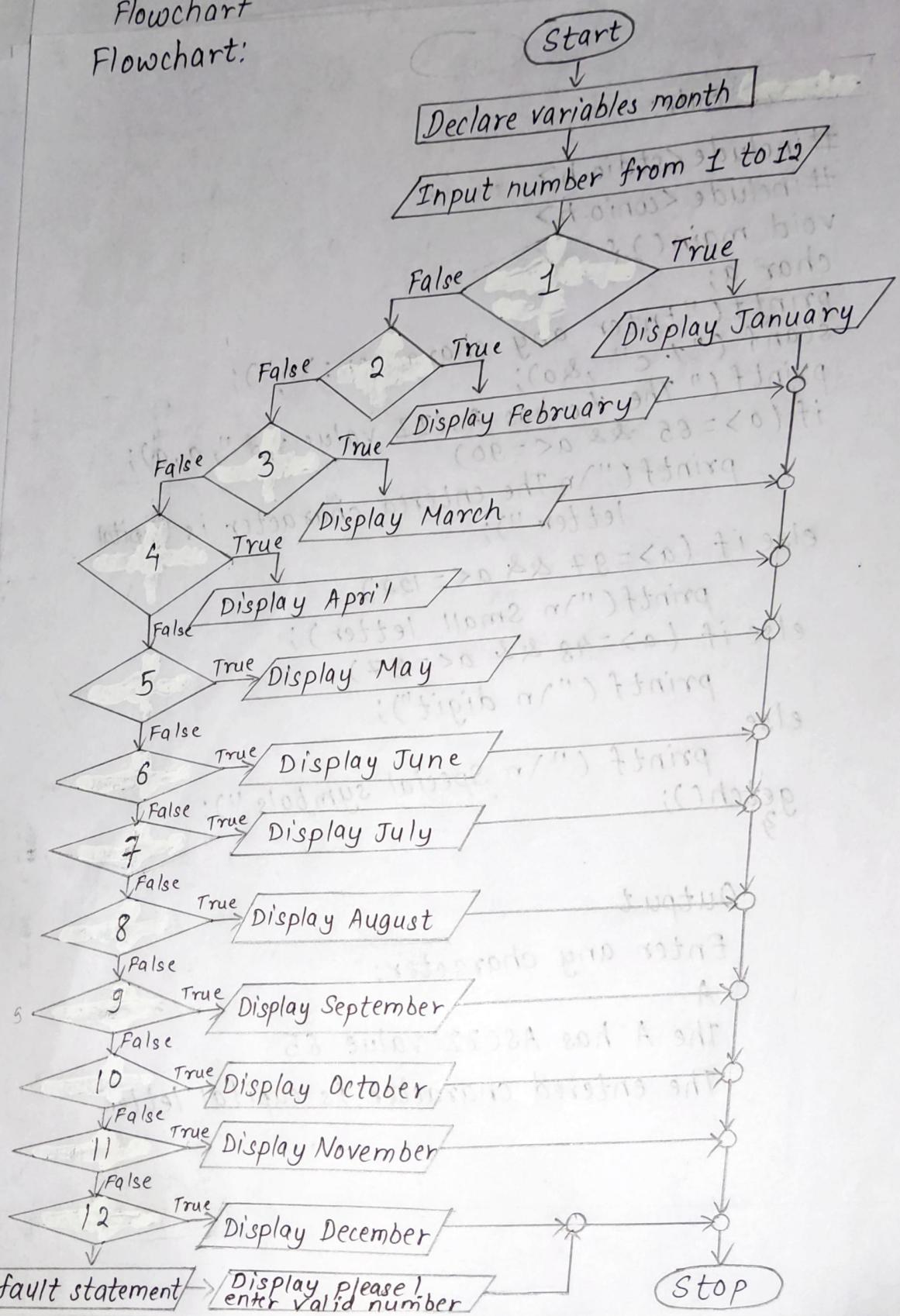
A

The A has ASCII value 65

The entered character is capital letter

Flowchart

Flowchart:



3. Read an integer value. Assume it is the number of a month of the year; print out the name of that month.

Algorithm

Step 1: start

Step 2: Declare month and number from 1 to 12.

Step 3: Input any number from 1 to 12.

Step 4: If number is equal to 1

 4.1) Display January

Step 5: If number is equal to 2

 5.1) Display February

Step 6: Else if number is equal to 3

 6.1) Display March

Step 7: Else if number is equal to 4

 7.1) Display April

Step 8: Else if number is equal to 5

 8.1) Display May

Step 9: Else if number is equal to 6

 9.1) Display June

Step 10: Else if number is equal to 7

 10.1) Display July

Step 11: Else if number is equal to 8

 11.1) Display August

Step 12: Else if number is equal to 9

 12.1) Display September

Step 13: Else if number is equal to 10

 13.1) Display October

Step 14: Else if month is equal to 11

14.1) Display November

Step 15: Else if month is equal to 12

15.1) Display December

Step 16: stop

Source code:

```
#include <stdio.h>
#include <conio.h>
int main() {
    int month;
    printf("Enter the number of month:\n");
    scanf("%d", &month);
    switch(month)
    {
        case 1:
            printf("January");
            break;
        case 2:
            printf("February");
            break;
        case 3:
            printf("March");
            break;
        case 4:
            printf("April");
            break;
        case 5:
            printf("May");
            break;
    }
}
```

```
case 6:  
    printf("June");  
    break;  
case 7:  
    printf("July");  
    break;  
case 8:  
    printf ("August");  
    break;  
case 9:  
    printf("September");  
    break;  
case 10:  
    printf ("October");  
    break;  
case 11:  
    printf ("November");  
    break;  
case 12:  
    printf ("December");  
    break;  
default:  
    printf("Please! enter valid number");  
    break;  
}  
return 0;  
}
```

Output

Enter the number of month:

11

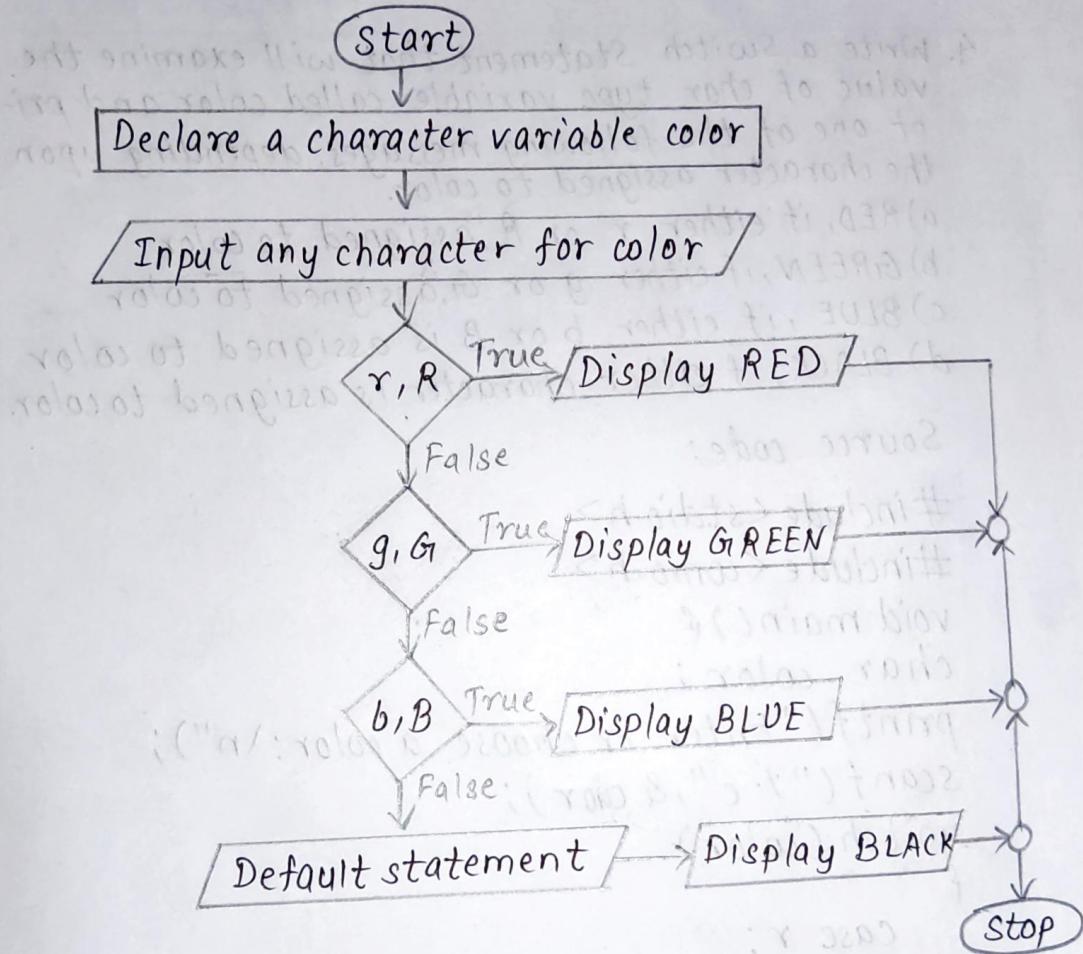
November

4. Write a Switch Statement that will examine the value of char type variable called color and print one of the following messages, depending upon the character assigned to color.
- a) RED, if either r or R is assigned to color
 - b) GREEN, if either g or G is assigned to color
 - c) BLUE, if either b or B is assigned to color
 - d) BLACK, if either character is assigned to color.

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char color;
    printf("Enter or choose a color:\n");
    scanf("%c", & color);
    switch (color)
    {
        case r:
        case R: printf("\n Red");
                    break;
        case g:
        case G: printf("\n GREEN");
                    break;
        case b:
        case B: printf("\n BLUE");
                    break;
        default: printf("\n BLACK");
    }
    getch();
}
```

Flowchart:



Algorithm

1. Start
2. Declare a character variable color
3. Input any character for color
4. If color is equal to r or R
 - 4.1) Display RED
5. If color is g or G
 - 5.1) Display GREEN
6. If color is b or B
 - 6.1) Display BLUE
7. If color is any other character
 - 7.1) Display BLACK
8. Stop

5. WAP that will read two numbers, display the following menu:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Square
6. Cube

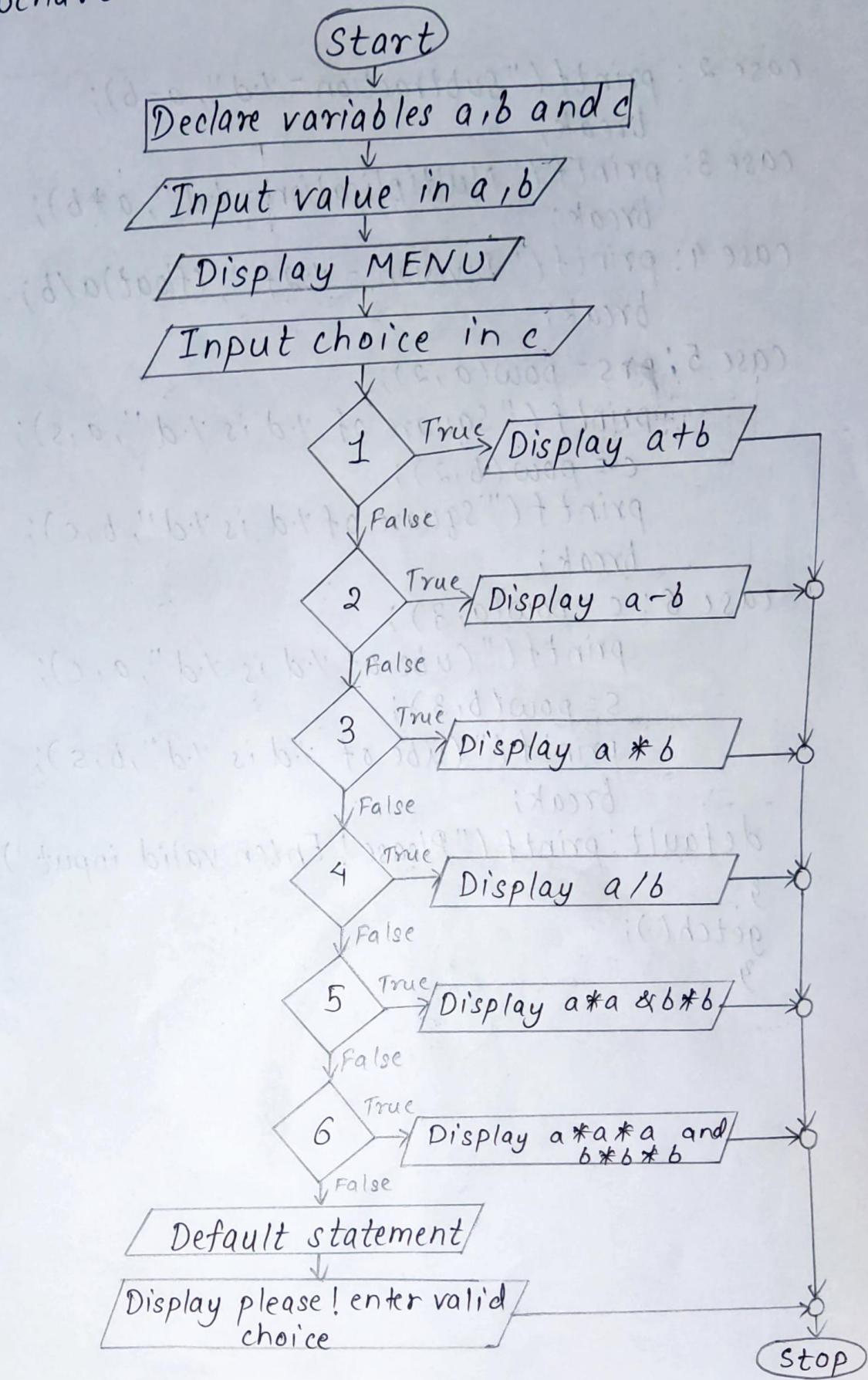
And perform task as per user's choice (use switch Statement)

Source code:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
    int a, b, c, s;
    printf("Enter two numbers: \n");
    scanf("%d %d", &a, &b);
    printf(" MENU \n 1. Addition \n 2. Subtraction \n
          3. Multiplication \n 4. Division \n
          5. Square \n 6. Cube ");
    printf(" Enter your choice : \n");
    scanf("%d", &c);
    switch(c)
    {
        case 1: printf(" Addition = %d ", a+b);
                  break;
```

```
case 2 : printf("Subtraction = %d ", a-b);
           break;
case 3 : printf("Multiplication = %d ", a*b);
           break;
case 4 : printf("Division = %.2f ", (float)a/b);
           break;
case 5 : s = pow(a,2);
           printf("Square of %d is %d ", a,s);
           c = pow(b,2);
           printf("Square of %d is %d ", b,c);
           break;
case 6 : c = pow(a,3);
           printf("Cube of %d is %d ", a,c);
           s = pow(b,3);
           printf("Cube of %d is %d ", b,s);
           break;
default: printf("Please! Enter valid input");
}
getch();
}
```

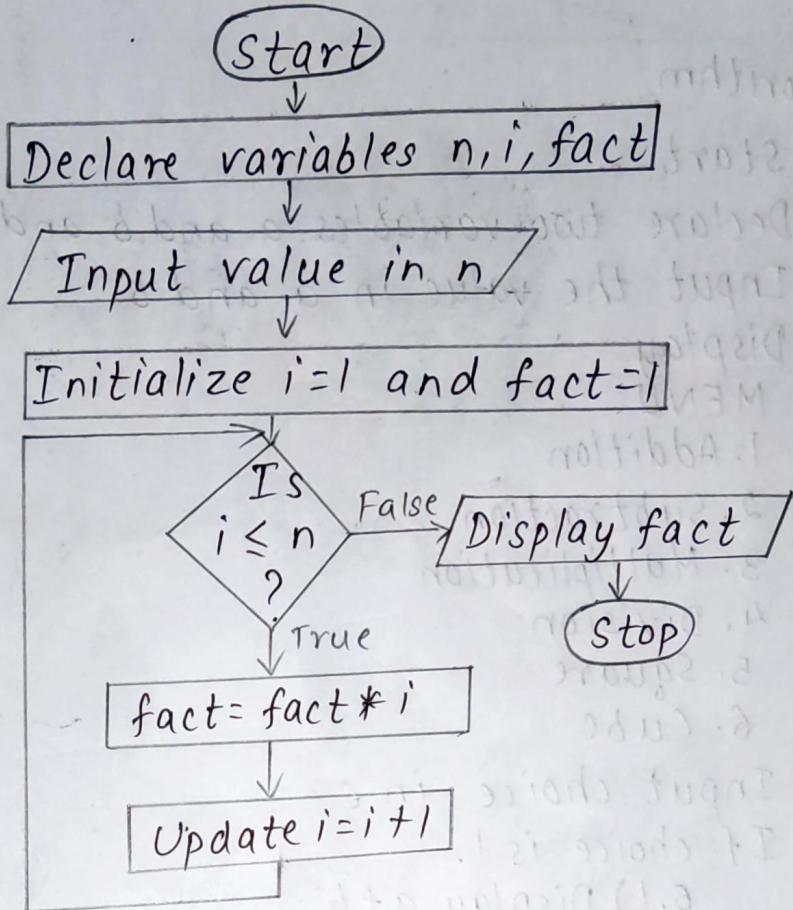
Flowchart



Algorithm

1. Start
2. Declare two variables a and b and c.
3. Input the value in a and b
4. Display
 MENU
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Square
 6. Cube
5. Input choice in c
6. If choice is 1.
 - 6.1) Display a+b
7. If choice is 2
 - 7.1) Display a-b
8. If choice is 3
 - 8.1) Display a*b
9. If choice is 4
 - 9.1) Display a/b
10. If choice is 5
 - 10.1) Display a*a and b*b
11. If choice is 6
 - 11.1) Display a*a*a and b*b*b
12. Stop

Flowchart



6. WAP to calculate the factorial of a given number.

Algorithm:

Step 1: Start

Step 2: Declare num, fact and count

Step 3: Initialize fact to 1 and count to 1.

Step 4: Enter the number in num.

Step 5: Repeat following until count is less than or equal to num.

5.1) fact = fact * count ;

5.2) count = count + 1

Step 6: print or display the factorial of the number

Step 7: Stop

Source Code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, i, fact;
    fact = 1;
    printf("Enter any number:\n");
    scanf("%d", &a);
    for(i=1; i<=a; i++)
    {
        fact = fact * i;
    }
    printf("The factorial of the number is %d", fact);
    getch();
}
```

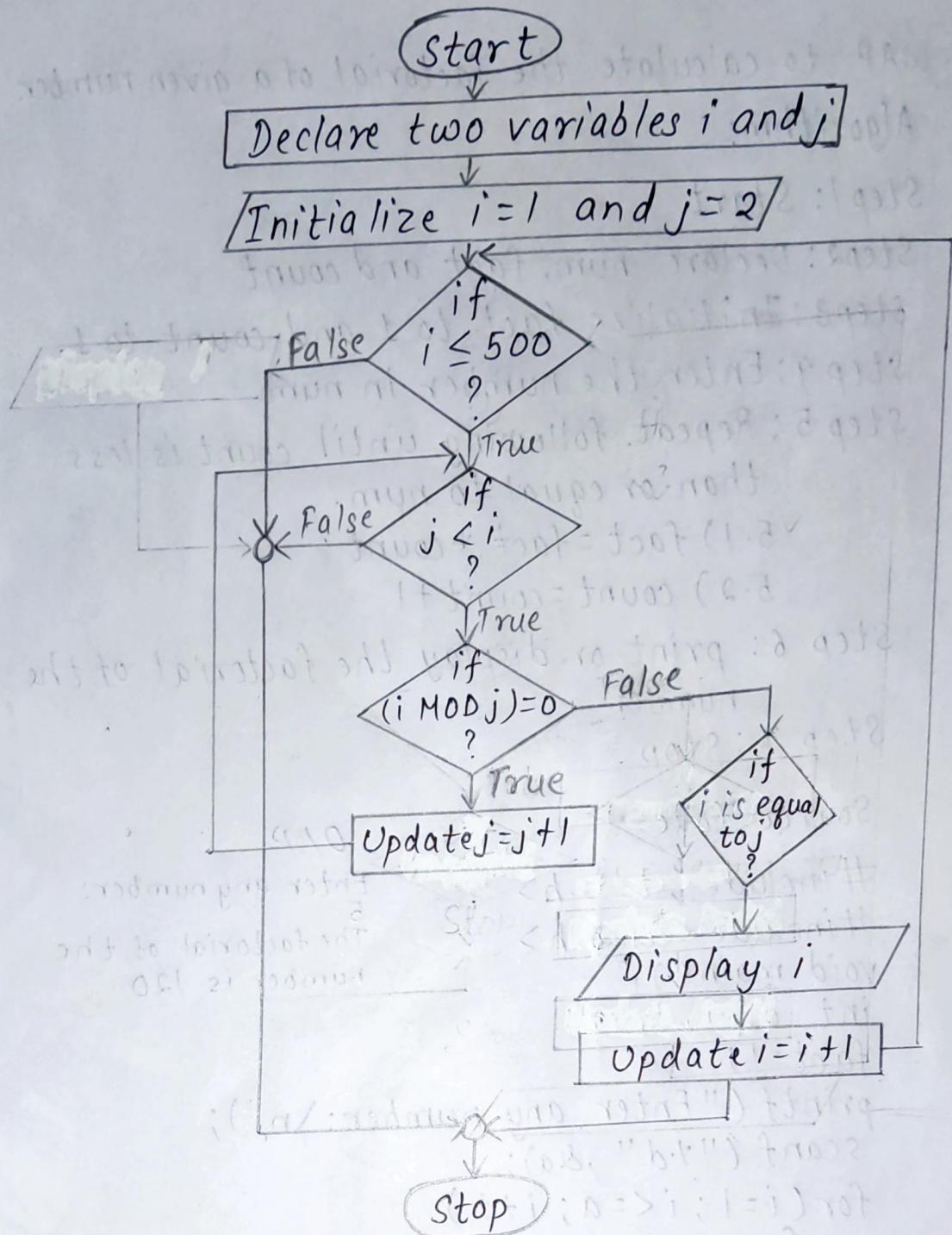
O/P

Enter any number:

5

The factorial of the number is 120

Flowchart



7. WAP to print all prime numbers from 1 to 500.
(A prime number is one which is divisible by 1 or itself)

Algorithm

Step 1: Start

Step 2: Declare variables i and j

Step 3: Initialize i to 1 and j to 2

Step 4: Repeat 5 and 6 until $i \leq 500$

 4.1) Repeat 5 and 6 until $j < i$

Step 5: is i exactly divisible by j? then

 5.1) Skip to 6.

 5.2) Update $j = j + 1$

Step 6: If i equals j then

 6.1) Display i

 6.2) Update $i = i + 1$

Step 7: stop

Source code

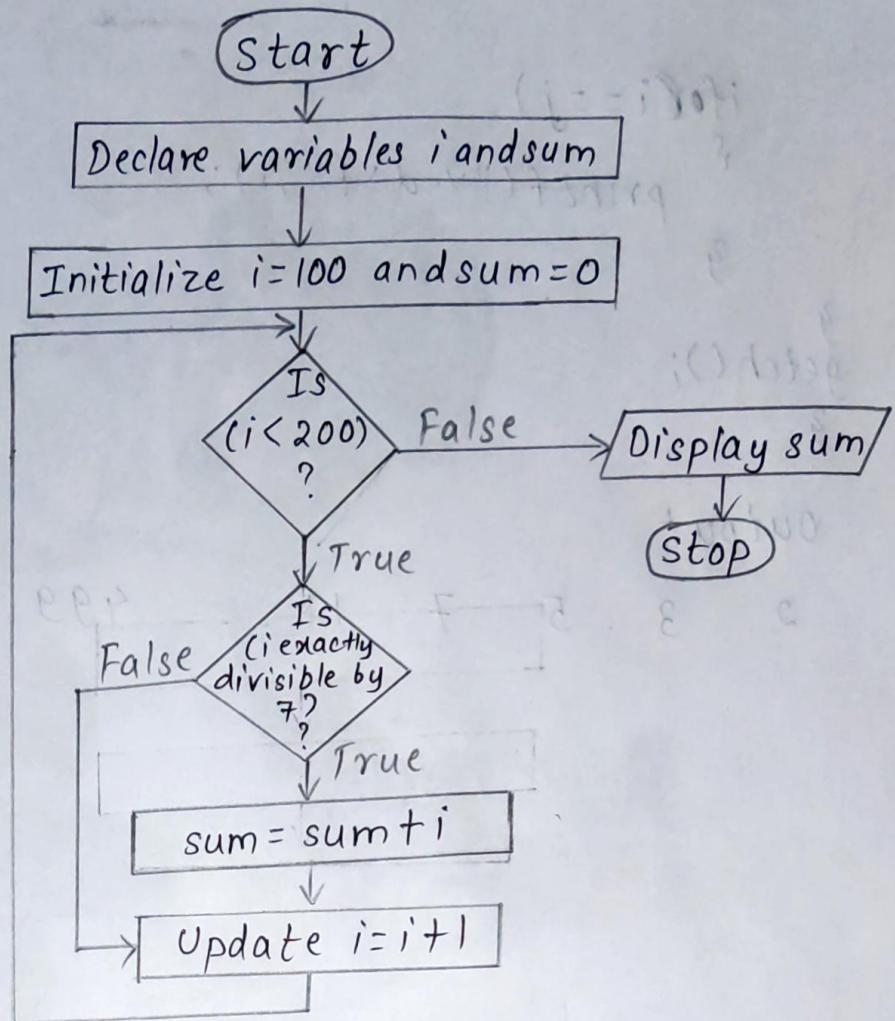
```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j;
    for(i=1; i<=500; i++)
    {
        for(j=2; j< i; j++)
        {
            if(i % j == 0)
            {
                break;
            }
        }
    }
}
```

```
if (i == j)
{
    printf("%d\t", i);
}
getch();
```

output

2 3 5 7 11 ... 499

Flowchart



8. WAP to find the number of sum of all integer greater than 100 and less than 200 that are divisible by 7.

Algorithm :

Step 1: Start

Step 2: Declare two variables i and sum

Step 3: Initialize i to 100 and sum to 0

Step 4: Repeat 5 and 6 until $i \leq 200$

Step 5: If i is exactly divisible by 7

 5.1) calculate $sum = sum + i$

Step 6: Update $i = i + 1$

Step 7: Display the value of sum

Step 8: Stop

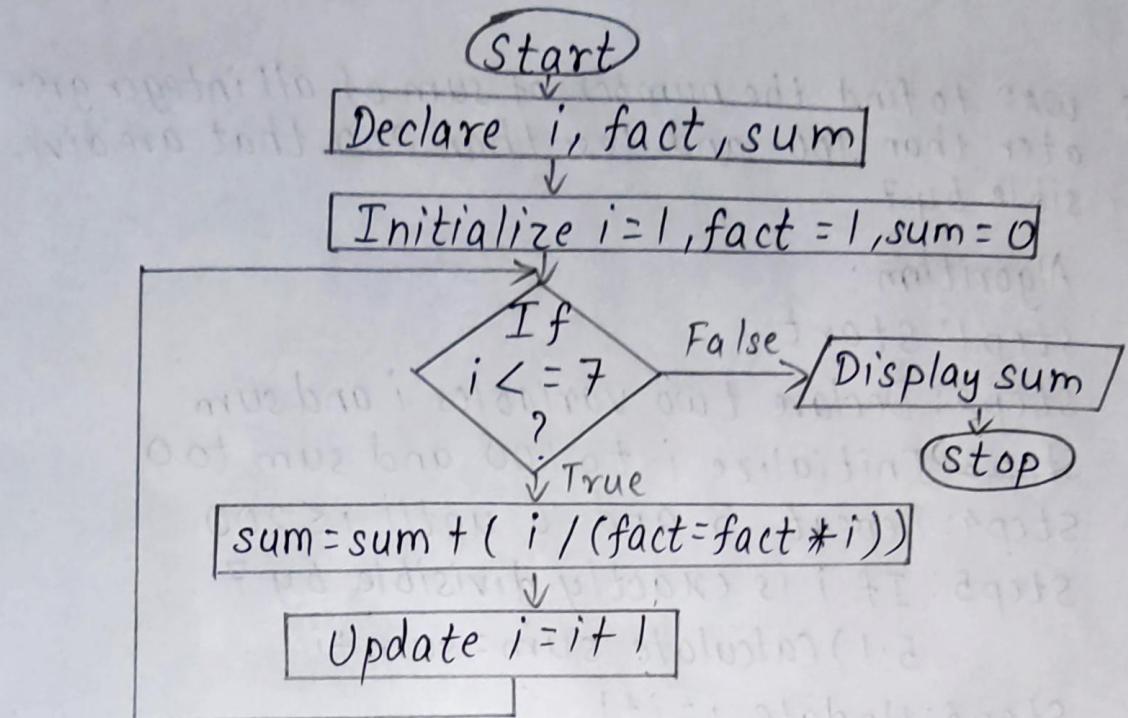
Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, sum = 0;
    for(i = 100; i <= 200; i++)
    {
        if(i % 7 == 0)
            sum += i;
    }
    printf("\n Sum = %d", sum);
    getch();
}
```

Output

Sum = 2107

Flowchart



9. WAP to add first seven terms of the following series using for loop.

$$1/1! + 2/2! + 3/3! + \dots$$

Algorithm

1. Start
2. Declare variables, i, j, fact, sum.
3. Initialize i=1 and j=1 and fact=1 and sum=0
4. Repeat 5 and 6 until i≤7 and j≤i
5. Calculate fact=fact*j
 - 5.1) sum = (i * 1.0) / fact
6. Update i=i+1 and j=j+1
7. Display sum
8. Stop

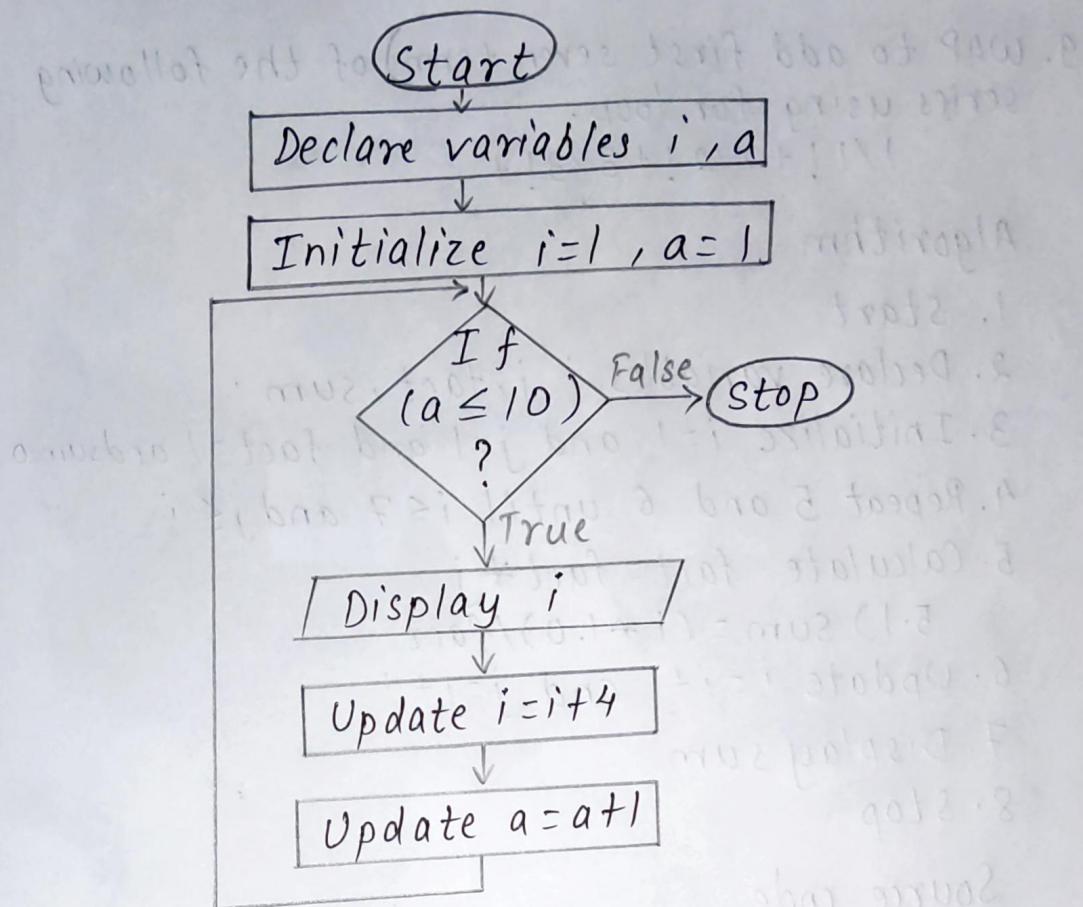
Source code

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n, fact = 1;
    float sum = 0;
    for (i = 1; i <= 7; i++)
    {
        sum += ((1.0) * i / (fact *= i));
    }
    printf("The sum is %.2f", sum);
    getch();
}
```

Output

The sum is 2.71

Flowchart



10. WAP to print first 10 terms of the following series using for loop.

1 5 9 13 . . .

Algorithm

1. Start
2. Declare variables i, a
3. Initialize i=1 and a=1.
4. Repeat 5 and 6 until a<=10.
5. Display i;
6. Update i+=4 and a=a+1
7. Stop

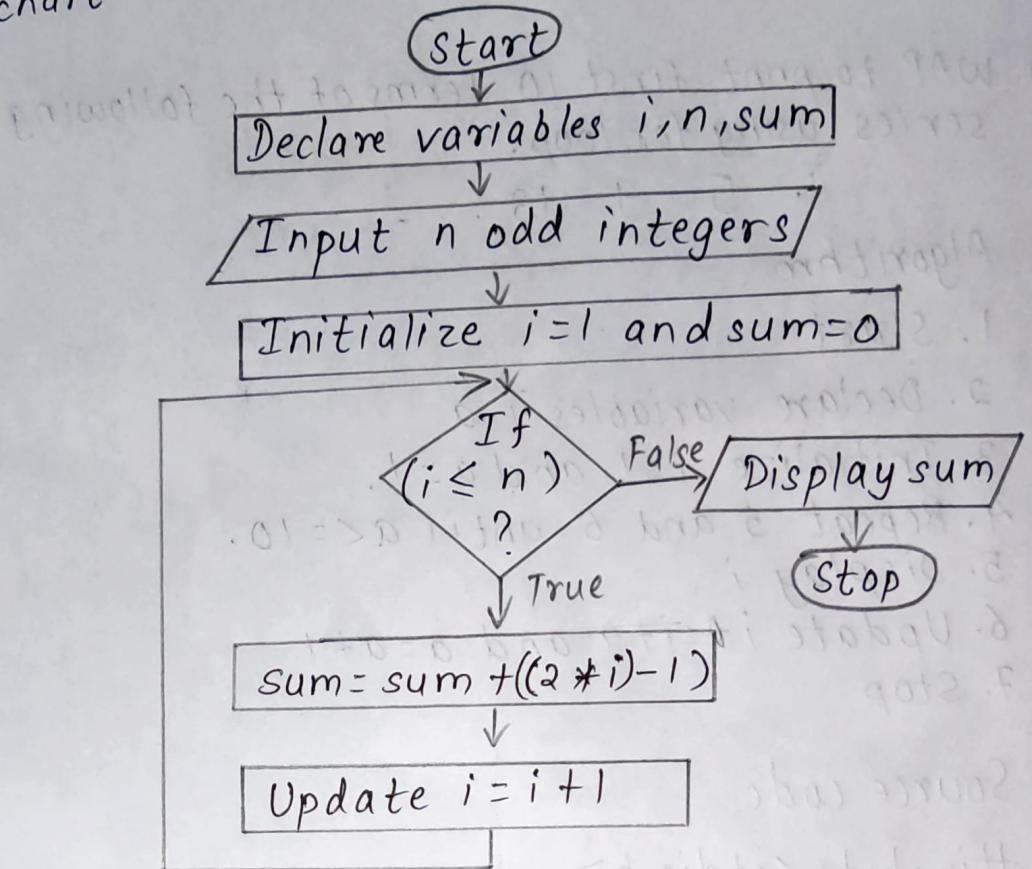
Source code

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, a;
    for(i=1, a=1; a<=10; i+=4, a++)
    {
        printf("y.d\t", i);
    }
    getch();
}
```

Output

1 5 9 13 17 21 25 29 33 37

Flowchart



11. WAP to find the sum of first n odd integers.
(i.e. $1+3+5+\dots+(2n-1)$)

Algorithm

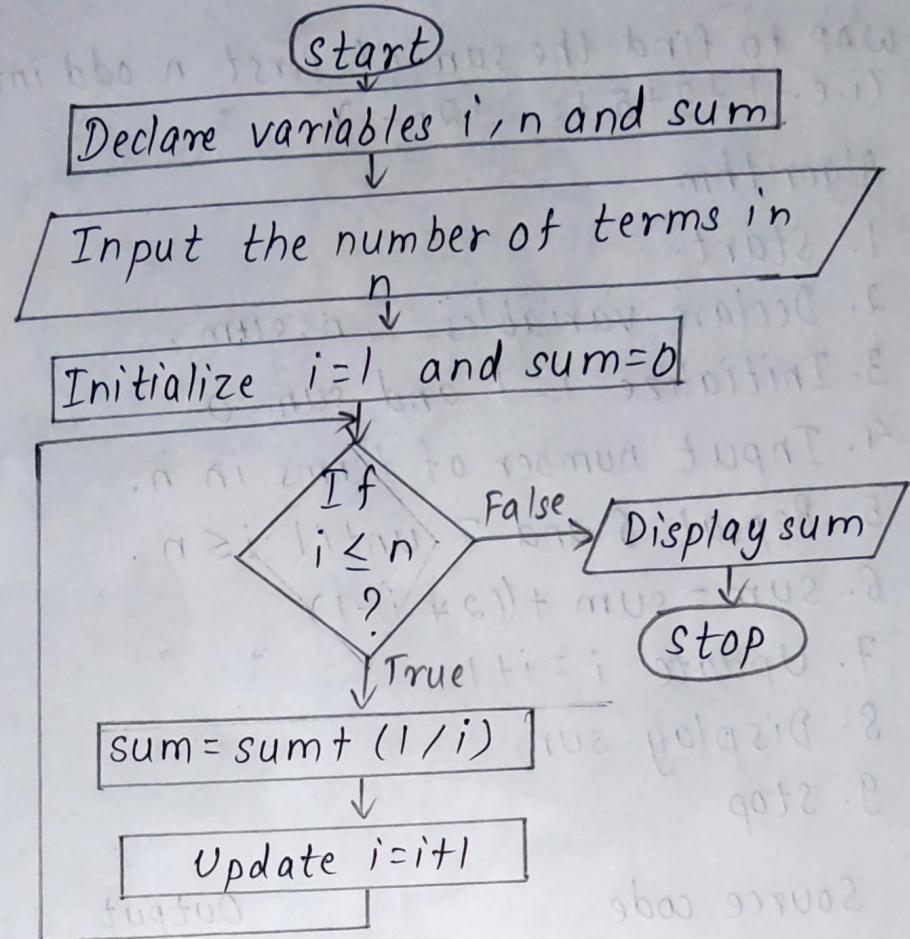
1. Start
2. Declare variables i, n, sum
3. Initialize i=1 and sum=0
4. Input number of terms in n.
5. Repeat 6 and 7 until i ≤ n.
6. sum = sum + ((2*i)-1)
7. Update i = i+1
8. Display sum
9. Stop

Source code

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,n,sum=0;
    printf("Enter the odd integers :\n");
    scanf("%d", &n);
    for(i=1;i<=n;i++)
    {
        sum += ((2*i)-1);
    }
    printf("The sum of odd integers is %d",sum);
    getch();
}
```

Output
Enter the odd integers:
5
The sum of odd integers is 25

Flowchart



12. WAP to find the sum of series
 $1/1 + 1/2 + 1/3 + \dots + 1/n$

Source code

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n;
    float sum = 0;
    printf("Enter the number of terms:\n");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        sum += (1 * 1.0) / i;
    }
    printf("The sum of series = %.2f", sum);
    getch();
}
```

Output

Enter the number of terms:

7

The sum of series = 2.59

Algorithm

1. Start
2. Declare variables i, n, sum
3. Input the number of terms in n
4. Initialize i=1 and sum=0

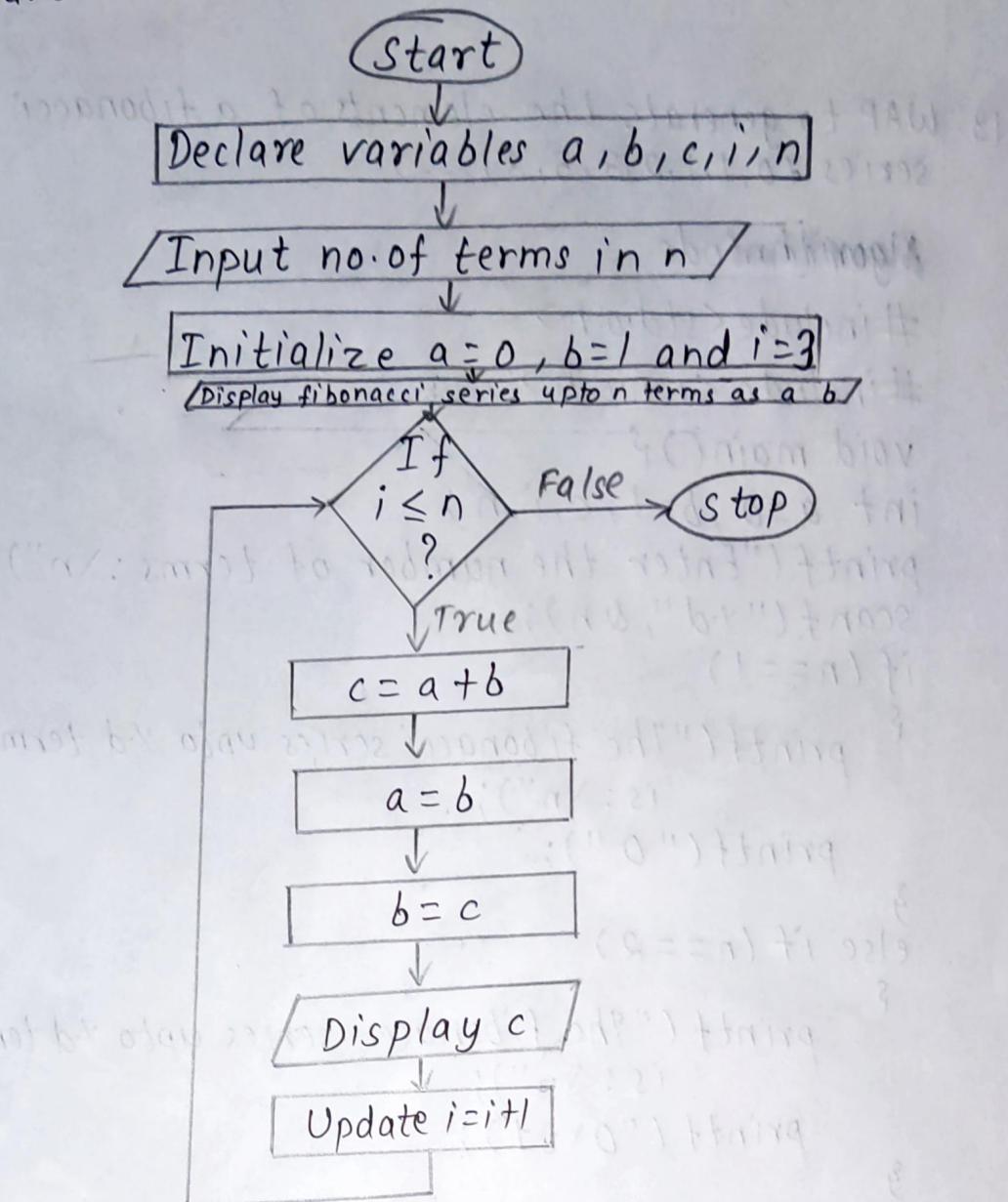
5. Repeat 6 until $i \leq n$
6. $\text{sum} = \text{sum} + (1 * 1.0) / i;$
7. Display sum
8. Stop

13. WAP to generate the elements of a fibonacci series (0,1,1,2,3,5,8,...)

Source Code

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a=0, b=1, c, i, n;
    printf("Enter the number of terms :\n");
    scanf("%d", &n);
    if (n==1)
    {
        printf("The fibonacci series upto %d terms
               is :\n");
        printf("0");
    }
    else if (n==2)
    {
        printf("The fibonacci series upto %d terms
               is :\n");
        printf("0\t1");
    }
    else
    {
        printf("The fibonacci series upto %d terms
               are :\n%d\t%d"; n, a, b);
        for (i=3; i<=n; i++)
        {
            c=a+b;
            a=b;
            b=c;
        }
    }
}
```

Flowchart



```
    printf("\t %d", c);  
}  
}  
getch();  
}
```

Algorithm

1. Start
2. Declare variables a, b, c, i, n ;
3. Initialize $a=0$ and $b=1$ also $i=3$
4. Input number of terms in n
5. Display fibonacci series upto n terms
is $\begin{matrix} a & b \end{matrix}$
6. If $i \leq n$ repeat 6.1, 6.2, 6.3 and 6.4
 - 6.1) $c = a + b$
 - 6.2) $a = b$
 - 6.3) $b = c$
 - 6.4) Display c
7. Stop

14. WAP to read the name of your college "Nepal College of Information Technology" and convert it to lowercase using loop.

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, j;
    char str[] = "Nepal College of Information
                  Technology"
    for(i=0; str[i] != '\0'; i++)
    {
        if(str[i] >= 65 && str[i] <= 90)
            str[i] += 32;
    }
    printf("%s", str);
    getch();
}
```

Output

nepal college of information technology.

15. WAP to read a string and count number of vowels in the given string.

Program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, length=0;
    char str[50];
    printf("Enter a string : \n");
    gets(str);
    for(i=0; str[i]!='\0'; i++)
    {
        if(str[i]=='a' || str[i]=='A' || str[i]=='e'
           || str[i]=='E' || str[i]=='i' || str[i] ==
           'I' || str[i]=='o' || str[i]=='O' ||
           str[i]=='u'|| str[i]=='U')
            length++;
    }
}
```

3 Output
printf ("%d", length); Enter a string:
getch(); Roshan
3 2

16. WAP to convert the string "this is c programming" in the following form by using loops and escape sequences.

This
Is
C
Programming

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i;
    char str[]="this is c programming";
    str[0]-=32;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]==' ')
        {
            str[i+1]-=32;
            printf("\n");
            continue;
        }
        printf("y.c",str[i]);
    }
    getch();
}
```

Output
This
Is
C
Programming

18. The sine of X can be calculated approximately by using the first n terms of the infinite series. (X is expressed in radians, note $\pi \text{ radians} = 180^\circ$)
$$\sin(X) = X - X^3/3! + X^5/5! - X^7/7! + \dots$$

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n;
    float x, sum, t;
    printf("Enter the value of x: \n");
    scanf("%f", &x);
    printf("\n Enter the value of n: \n");
    scanf("%d", &n);
    x = x * 3.14 / 180;
    t = x;
    sum = x;
    for (i=1; i<=n; i++)
    {
        t = (t * (-1) * x * x) / (2 * i * (2 * i + 1));
        sum = sum + t;
    }
    printf("The value of sine is %.f", sum);
    getch();
}
```

O/P

Enter the value of x:

2

Enter the value of n:

3

The value of sine is 0.034882

19. WAP to calculate the sum of cosine series
 $\cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n;
    float x, sum, t;
    printf("Enter the value of x: \n");
    scanf("f", &x);
    printf("Enter the number of terms: \n");
    scanf("d", &n);
    x = x * 3.14 / 180;
    t = 1;
    sum = 1;
    for (i=1; i<=n; i++)
    {
        t = (t * (-1) * x * x) / (2 * i * (2 * i - 1));
        sum += t;
    }
    printf("The value of cosine is %f", sum);
    getch();
}
```

Output

Enter the value of x:

2

Enter the number of terms:

3

The value of cosine is 0.999391

20. WAP to print the numbers in the following format using loops

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 2 | 1 |
| 4 | 5 | 6 | 5 | 4 |
| 7 | 8 | 9 | 8 | 7 |
| 10 | 11 | 12 | 11 | 10 |
| 13 | 14 | 15 | 14 | 13 |

Source code :

```
#include<stdio.h>
#include <conio.h>
void main()
{
    int i, j, k=0, l;
    for(i=1; i<=5; i++)
    {
        for(j=1; j<=3; j++)
        {
            printf("%d\t", ++k);
        }
        l = k;
        for(j=1; j<=2; j++)
        {
            printf("%d\t", --l);
        }
        printf("\n");
    }
    getch();
}
```

21. WAP to print the following using loops.

1
0 1
1 0 1
0 1 0 1
1 0 1 0 1

Source code:

```
#include<stdio.h>
#include <conio.h>
void main()
{
    int i,j;
    for(i=1; i<=5; i++)
    {
        for(j=1; j<=i; j++)
        {
            if((i+j)%2==0)
                printf("1\t");
            else
                printf("0\t");
        }
        printf("\n");
    }
    getch();
}
```

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

Source code:

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

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 2 | 2 | 2 |
| 3 | 3 | 0 | 3 | 3 |
| 4 | 4 | 4 | 0 | 4 |
| 5 | 5 | 5 | 5 | 0 |

Source code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j;
    for(i=1; i<=5; i++)
    {
        for(j=1; j<=5; j++)
        {
            if(i==j)
                printf("0\t");
            else
                printf("y.d\t",i);
            printf("\n");
        }
        getch();
    }
}
```