



GOVERNMENT POLYTECHNIC, BARGARH
DEPARTMENT OF MATHEMATICS & SCIENCE

PROGRAMMING IN C FOR DIPLOMA STUDENTS

Prepared By:
Dr. Anitarani Brahma,
Sr. Lecturer, IT

Contents

- Simple Programs
- Program using if,if-else, nested if else, switch case
- Program using while loop, do-while and for loop.
- Program using 1-D array, 2-D array
- Program using string
- Program using function

Simple C program

Program1:

```
/* Program to print HELLO*/  
#include<stdio.h>  
int main()  
{  
    printf("Hello");  
    return 0;  
}
```

Note:

#include<stdio.h> : It is the preprocessor directive. It includes all the library functions (present in header file **stdio.h**) that are being used in the program.

stdio.h: stdio stands for Standard Input Output that holds all the library function related to input and output.

printf(): It is a library function present in stdio.h header file. It's function is to print message and print value. It is a output function.

Program2:

```
/* Program to Add two numbers*/  
#include<stdio.h>  
int main()  
{  
    int a=100,b=200,sum; //a,b,sum: are variable  
    sum=a+b;  
    printf("The result of sum is %d", sum);  
    return 0;  
}
```

Note:

int: It is a primary data type that describes the type of value the variable holds and the size of the variable.

%d: It is a format specifier of int.

Primary/Built-in/Basic Data type: int (value= integer, format specifier=%d, size=2 bytes)

float (value= real, format specifier=%f, size=2 bytes)

double (value= large real, format specifier=%f, size=2 bytes)

char (value= character, format specifier=%f, size=2 bytes)

Program3:

```
/* Program to Add two numbers using scanf()*/  
#include<stdio.h>  
int main()  
{  
    int a,b,sum; //a,b,sum: are variable  
    printf("Enter two values\n");  
    scanf("%d%d", &a,&b);  
    sum=a+b;  
    printf("The result of sum is %d", sum);  
    return 0;  
}
```

Note:

scanf(): It is a library function present in stdio.h header file. It's function is to take input value of variable from keyboard and store in the address of the variable . It is a input function.

&: Called ampersand/address-of operator

Program4:

```
/* Program to perform all arithmetic operations between two numbers*/
#include<stdio.h>
int main()
{
    int a,b,sum,sub,mul;
    float div,mod;
    printf("Enter two values\n");
    scanf("%d%d", &a,&b);
    sum=a+b;
    sub=a-b;
    mul=a*b;
    div=a/b;
    mod=a%b;
    printf("a+b=%d\na-b=%d\na*b=%d\na/b=%f\na%b=%f", sum,sub,mul,div,mod);
    return 0;
}
```

Note:

\n: It is one type of escape sequence called new line character. It takes the cursor to the next line.

Program5:

```
/* Program to calculate simple interest*/  
#include<stdio.h>  
int main()  
{  
    int p,t;  
    float r,si;  
    printf("Enter principal amount, time, rate of interest \n");  
    scanf("%d%d%f", &p,&t,&r);  
    si= (p*t*r)/100;  
    printf("Simple interest is=%d", si);  
    return 0;  
}
```

Program6:

```
/* Program to area and perimeter of a circle*/  
#include<stdio.h>  
#define pi=3.141  
int main()  
{  
    int r;  
    float area,peri;  
    printf("Enter the radius value of a circle \n");  
    scanf("%d",&r);  
    area= pi*r*r;  
    peri=pi*r*r;  
    printf("area=%f\t perimeter=%f", area,peri);  
    return 0;  
}
```

Note:

\t: Escape Sequence for tab

Program7:

```
/* Program to change the temperature from fahrenheit to centigrade */
#include<stdio.h>
#define pi=3.141
int main()
{
    int r;
    float area,peri;
    printf("Enter the radius value of a circle \n");
    scanf("%d",&r);
    area= pi*r*r;
    peri=pi*r*r;
    printf("area=%f\t perimeter=%f", area,peri);
    return 0;
}
```

Note:

\t: Escape Sequence for tab

Program8:

```
/* Program to change the temperature from Fahrenheit to centigrade */  
#include<stdio.h>  
int main()  
{  
    float f,c;  
    printf("Enter the temperature value in Fahrenheit \n");  
    scanf("%f",&f);  
    c = ((f-32)*5)/9;  
    printf("temperature in celcius=%f", c);  
    return 0;  
}
```

Program9:

```
/* Program to change the temperature from Fahrenheit to centigrade */  
#include<stdio.h>  
int main()  
{  
    float f,c;  
    printf("Enter the temperature value in Celcius \n");  
    scanf("%f",&c);  
    f= (c*9/5)+32;  
    printf("temperature in celcius=%f", f);  
    return 0;  
}
```

Conditional Statement (if)

Program 1:

```
/* Program to check a even number*/  
#include<stdio.h>  
int main()  
{  
    int a;  
    printf("Enter a number\n");  
    scanf("%d",&a);  
    if(a%2==0)  
    {  
        printf("%d is an even number",a);  
    }  
    return 0;  
}
```

Note:

%= modulus operator that determines the remainder value after division.

Conditional Statement (if-else)

Program 2:

```
/* Program to check a number even or odd*/  
#include<stdio.h>  
int main()  
{  
    int a;  
    printf("Enter a number\n");  
    scanf("%d",&a);  
    if(a%2==0)  
    {  
        printf("%d is an even number",a);  
    }  
    else  
    {  
        printf("%d is a odd number",a);  
    }  
    return 0;  
}
```

Note:

If-else: if the condition is true then the if block will be executed and if the condition is false then, the else block will be executed.

Conditional Statement (if-else)

Program 3:

```
/* Program to check a number is positive or negative */
#include<stdio.h>
int main()
{
    int a;
    printf("Enter a number\n");
    scanf("%d",&a);
    if(a>0)
    {
        printf("%d is a positive number",a);
    }
    else
    {
        printf("%d is a negative number", a);
    }
    return 0;
}
```

Note:

If-else: if the condition is true then the if block will be executed and if the condition is false then, the else block will be executed.

Conditional Statement (if-else)

Program 4:

```
/* Program to check a number is positive or negative */
#include<stdio.h>
int main()
{
    int a;
    printf("Enter a number\n");
    scanf("%d",&a);
    if(a>0)
    {
        printf("%d is a positive number",a);
    }
    else
    {
        printf("%d is a negative number", a);
    }
    return 0;
}
```

Note:

If-else: if the condition is true then the if block will be executed and if the condition is false then, the else block will be executed.

Conditional Statement (if-else)

Program 5:

```
/* Program to determine largest between two numbers*/
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter two numbers\n");
    scanf("%d%d",&a&b);
    if(a>b)
    {
        printf("%d is largest", a);
    }
    else
    {
        printf("%d is a largest", b);
    }
    return 0;
}
```

Note:

If-else: if the condition is true then the if block will be executed and if the condition is false then, the else block will be executed.

Conditional Statement (Nested if-else)

Program 6:

/* Program to check a number is positive or negative*/

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

```
int main()
```

```
{
```

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

```
    printf("Enter three numbers\n");
```

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

```
    if(a>b)
```

```
    {
```

```
        if(a>c)
```

```
        {
```

```
            printf("%d is largest", a);
```

```
        }
```

```
    else
```

```
    {
```

```
        printf("%d is largest",c);
```

```
    }
```

```
}
```

```
if(b>c)
```

```
{
```

```
    printf("%d is largest", b);
```

```
}
```

```
else
```

```
{
```

```
    printf("%d is largest",c);
```

```
}
```

```
return 0;
```

```
}
```

Conditional Statement (if-else ladder)

Program 7:

```
/* Program to compare a number*/  
#include<stdio.h>  
int main()  
{  
    int a=20;  
    printf("Enter three numbers\n");  
    scanf("%d",&a);  
    if(a==10)  
        printf("a is 10");  
    else if(a==15)  
        printf("a is 15");  
    else if(a==20)  
        printf("a is 20");  
    else  
        printf("not found");  
    return 0;  
}
```

Conditional Statement (switch-case)

Program 8:

```
/* Program to check a letter if it is a vowel or
consonant*/
#include<stdio.h>
int main()
{
char d;
printf("Enter a character\n");
scanf("%c",&d);
switch( d )
{
case 'a' || 'A':
    printf("Vowel");
    break;
case 'e' || 'E':
    printf("Vowel");
    break;
case 'i' || 'I':
    printf("Vowel");
    break;
case 'o' || 'O':
    printf("Vowel");
    break;
```

```
case 'u' || 'U':
    printf("Vowel");
    break;
default:
    printf("Consonant");
return 0;
}
```

Note:

-Switch-case is a conditional statement which is an extension of if-else ladder.

-- Syntax:

```
switch(expression)
{
    case label1:
        .....
        break;
    case label2:
        .....
        break;
    case label3:
        .....
        break;
    ....
    default:
        .....
}
```

- First, the expression is evaluated, and the labels of each case are matched and that case block is executed where the case label is matched.

Conditional Statement (switch-case)

Program 9:

```
/* Program to enter a number in between 1-7 and print  
the day*/
```

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

```
int main()
```

```
{
```

```
int day;
```

```
printf("Enter a value in between 1 to 7\n");
```

```
scanf("%d",&day);
```

```
switch( d ay)
```

```
{
```

```
case 1:
```

```
    printf("Monday");
```

```
    break;
```

```
case 2:
```

```
    printf("Tuesday");
```

```
    break;
```

```
case 3:
```

```
    printf("Wednesday");
```

```
    break;
```

```
case 4:
```

```
    printf("Thursday");
```

```
    break;
```

```
case 5:
```

```
    printf("Friday");
```

```
    break;
```

```
case 6:
```

```
    printf("Saturday");
```

```
    break;
```

```
case 7:
```

```
    printf("Sunday");
```

```
    break;
```

```
default:
```

```
    printf("Not a valid number");
```

```
return 0;
```

```
}
```

Loop (while loop)

Program 1:

```
/* Program to print from 1 to 100*/  
#include<stdio.h>  
int main()  
{  
int i=1;  
while(i<=100)  
{  
printf("%d\n",i);  
}  
return 0;  
}
```

Note:

Loop: It is a block of statement executed repeatedly till a termination condition satisfied.

Program 2:

```
/* Program to print all even numbers  
between 1 to 100*/  
#include<stdio.h>  
int main()  
{  
int i=2;  
while(i<=100)  
{  
if(i%2==0)  
{  
printf("%d\n",i);  
}  
}  
return 0;  
}
```

While Loop is otherwise called entry controlled loop as the condition is first checked and then the body of the loop is executed.

Loop (do-while loop)

Program 3:

```
/* Program to print from 1 to 100*/  
#include<stdio.h>  
int main()  
{  
int i=1;  
do  
{  
printf("%d\n",i);  
i++;  
}  
while(i<=100);  
return 0;  
}
```

Note:

Loop: It is a block of statement executed repeatedly till a termination condition satisfied.

Program 4:

```
/* Program to print all even numbers  
between 1 to 100*/  
#include<stdio.h>  
int main()  
{  
int i=2;  
do  
{  
if(i%2==0)  
{  
printf("%d\n",i);  
}  
i++;  
}while(i<=100);  
return 0;  
}
```

Do-while loop is otherwise called **exit controlled loop** as the body of the loop is executed first and the condition is checked then. In this case, the body of the loop is executed at least once even if the condition is satisfied.

Loop (for loop)

Program 5:

```
/* Program to print from 1 to 100 using  
for loop*/
```

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

```
int main()
```

```
{
```

```
int i;
```

```
for(i=0;i<=100;i++)
```

```
{
```

```
    printf("%d",i);
```

```
}
```

```
return 0;
```

```
}
```

Note:

Loop: It is a block of statement executed repeatedly till a termination condition satisfied.

Program 6:

```
/* Program to print all even numbers  
between 1 to 100*/
```

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

```
int main()
```

```
{
```

```
int i;
```

```
for(i=2;i<=100;i++)
```

```
{
```

```
    if(i%2==0)
```

```
    {
```

```
        printf("%d",i);
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```

Loop (for loop)

Program 7:

```
/* Program to calculate sum and average of
   all numbers between 1 to n*/
#include<stdio.h>
int main()
{
    int i,sum=0;
    printf("Enter the values of n")
    for(i=1;i<=n;i++)
    {
        sum=sum+i;
    }
    average=sum/n;
    printf("sum is=%d\n",sum);
    printf("average is=%d\n",average);
    return 0;
}
```

Note:

Loop: It is a block of statement executed repeatedly till a termination condition satisfied.

Loop (for loop)

Program 8:

/* Program to check a number palindrome or not*/

```
#include <stdio.h>
int main()
{
    int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n; // reversed integer is stored in reversed
    variable
    while (n != 0)
    {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
        n = n/10;
    } // palindrome if original and reversed are equal
    if (original == reversed)
        printf("%d is a palindrome.", original);
    else
        printf("%d is not a palindrome.", original);
    return 0;
}
```

Loop (for loop)

Program 9:

```
/* Program to check a number prime or not*/
#include <stdio.h>
int main()
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n); // 0 and 1 are not prime numbers // change flag to 1 for
non-prime number
    if (n == 0 || n == 1) flag = 1;
    for (i = 2; i <= n / 2; ++i)
    { // if n is divisible by i, then n is not prime // change flag to 1 for non-prime
number
        if (n % i == 0)
        {
            flag = 1;
            break;
        }
    } // flag is 0 for prime numbers
    if (flag == 0)
        printf("%d is a prime number.", n);
    else
        printf("%d is not a prime number.", n);
    return 0;
}
```

Loop (for loop)

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Program 10:

/* Program to calculate sine series where x is the radian value*/

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

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    int i, n;
```

```
    float x, sum, t;
```

```
    clrscr();
```

```
    printf(" Enter the value for x : ");
```

```
    scanf("%f",&x);
```

```
    printf(" Enter the value for n : ");
```

```
    scanf("%d",&n);
```

```
    x=x*3.14159/180;
```

```
    t=x;
```

```
    sum=x;
```

```
    /* Loop to calculate the value of Sine */
```

```
    for(i=1;i<=n;i++)
```

```
    {
```

```
        t=(t*(-1)*x*x)/(2*i*(2*i+1));
```

```
        sum=sum+t;
```

```
    }
```

```
    printf(" The value of Sin(%f) = %.4f",x,sum);
```

```
    getch();
```

```
}
```

Loop (Nested for loop)

Program 11:

```
/* Program to print the following pyramid
*
**
***
****
*/
#include<stdio.h>
int main()
{
    int i;
    for(i=1;i<=4;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("*\t");
        }
        printf("\n");
    }
    return 0;
}
```

Program 12:

```
/* Program to print the following pyramid
1
1 2
1 2 3
1 2 3 4
*/
#include<stdio.h>
int main()
{
    int i;
    for(i=1;i<=4;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d\t",j);
        }
        printf("\n");
    }
    return 0;
}
```

Array

Program 1:

```
/* Program to print the element of
   array*/
#include<stdio.h>
int main()
{
    int a[]={10,20,30,40,50},i;
    printf("The value of array is\n");
    for(i=0;i<5;i++)
    {
        printf("%d\n",a[i]);
    }
    return 0;
}
```

Note:

Array: It is a collection of similar types of elements

Declaration of array

<datatype> <arr_name>[size];

Program 2:

```
/* Program to input & print the element
   of array*/
#include<stdio.h>
int main()
{
    int a[10],i;
    printf("Enter the values of array\n");
    for(i=0;i<5;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("The value of array is\n");
    for(i=0;i<5;i++)
    {
        printf("%d\n",a[i]);
    }
    return 0;
}
```

Array

Program 3:

```
/* Program to calculate the sum of all the
   element of the array*/
#include<stdio.h>
int main()
{
    int a[10],i,sum=0,avg;
    printf("Enter the values of array\n");
    for(i=0;i<5;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<5;i++)
    {
        sum=sum+a[i];
    }
    printf("The sum is=%d",sum);
    return 0;
}
```

Program 4:

```
/* Program to calculate the avg of the
   element of the array*/
#include<stdio.h>
int main()
{
    int a[10],i,sum=0,avg,n;
    printf("Enter the number of elements of
           the array");
    scanf("%d",&n);
    printf("Enter the values of array\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<5;i++)
    {
        sum=sum+a[i];
    }
    avg=sum/n;
    printf("The average is=%d",avg);
    return 0;
}
```

String

Program 1:

```
/* Program to illustrate strlen() of string library
   function*/
#include<stdio.h>
#include<string.h>
int main()
{
    char a[10];
    printf("Enter a string\n");
    scanf("%d",a);
    printf("The length of the string
           is=%d",strlen(a));
    return 0;
}
```

Note:

strlen(): It is a string library function that calculate the number of characters present in the string.

Syntax: strlen(str_name)

Program 2:

```
/* Program to illustrate strcpy() of string library
   function*/
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20]="Hello World";
    char str2[20];
    strcpy(str2,str1);
    printf("The string1 is %s",str1);
    printf("The string2 is %s",str2);
    return 0;
}
```

Note:

strcpy(): It is a string library function that copy content of one string to other string.

Syntax: strcpy(string2,string1)

String

Program 3:

```
/* Program to illustrate strrev() of string library
   function*/
#include<stdio.h>
#include<string.h>
int main()
{
    char str[10]="Hello";
    printf("Reverse string is %s",strrev(str));
    return 0;
}
```

Note:

strrev(): It is a string library function that reverse the given string.

Syntax: strrev(str_name)

Program 4:

```
/* Program to illustrate strcmp() of string library
   function*/
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[20]="Hello";
    char str2[20]="hello"
    if(strcmp(str1,str2)==0)
    {
        printf("Both the strings are equal");
    }
    else
    {
        printf("Both the strings are notequal");
    }
    return 0;
}
```

Note:

strcmp(): It is a string library function that compare two strings. It returns 0 if both are equals and returns the ascii difference between the character where they are not equal

Syntax: strcmp(string1,string2)

Function

Program 1:

```
/* Program to use a user defined
   function*/
#include<stdio.h>
int add(int,int); //function declaration
int main()
{
    int a,b,sum;
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b); //function call
    sum=add(a,b); //a,b-actual arguments
    printf("Sum =%d",sum);
    return 0;
}
int add(int i,int j) //function definition
{
    //i,j are formal arguments
    int c;
    c=i+j;
    return j;
}
```

Note:

Function declaration: It describes the name of the function, the number of arguments, the types of arguments and the type of output produced by the function.

Function Definition: Actual body of the function that define the work/purpose of the function.

Function Call: Every function needs to be called to be executed.

The function is called only by specifying its name and the arguments.

Actual arguments: Are the input to the function when it is called. They hold the actual value that are passed to the function.

Formal arguments: Are the copy of actual arguments. They have different memory locations than actual arguments.

Function

Program 2:

```
/* Program to use a user defined function*/
#include<stdio.h>
int add(int,int); //function declaration
int mul(int,int);
int sub(int,int);
int main()
{
    int a,b,sum,sub,mul;
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b); //function call
    sum=add(a,b); //a,b-actual arguments
    mul=mul(a,b);
    sub= sub(a,b);
    printf(" sum is=%d\tpproduct is=%d\t subtraction
        result=%d", sum,mul,sub);
    return 0;
}
int add(int i,int j) //function definition
{
    //i,j are formal arguments
    int c;
    c=i+j;
    return j;
}
```

```
int add(int i,int j) //function definition
{
    int c;
    c=i+j;
    return j;
}
int sub(int i,int j) //function definition
{
    int c;
    c=i-j;
    return j;
}
int mul(int i,int j) //function definition
{
    int c;
    c=i*j;
    return j;
}
```

Function

Program 3:

```
/* Program to illustrate call by value*/
#include<stdio.h>
void swap(int,int); //function declaration
int main()
{
    int a,b;
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b); //function call
    printf("Before Swap a=%d\tb=%d",a,b);
    swap(a,b); //a,b-actual arguments
    printf("After Swap a=%d\tb=%d",a,b);
    return 0;
}
void swap(int i,int j) //function definition
{
    //i,j are formal arguments
    int temp;
    Temp=i;
    i=j;
    j=temp;
}
```

Program 4:

```
/* Program to illustrate call by reference*/
#include<stdio.h>
void swap(int&,int&); //function declaration
int main()
{
    int a,b;
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b); //function call
    printf("Before Swap a=%d\tb=%d",a,b);
    swap(&a,&b); //a,b-actual arguments
    printf("After Swap a=%d\tb=%d",a,b);
    return 0;
}
void swap(int *i,int *j) //function definition
{
    //i,j are formal arguments
    int temp;
    temp=*i;
    *i=*j;
    *j=temp;
}
```

Function

Program 5:

```
/* Program to illustrate factorial using
   recursion */
#include<stdio.h>
int factorial(int); //function declaration
int main()
{
    int a,f;
    printf("Enter a number\n");
    scanf("%d",&a); //function call
    f=factorial(a)
    printf("Factorial of %d is %d",n,f);
    return 0;
}
int factorial(int n)
{
    if(n==0)
        return 0;
    else if(n==1)
        return 1;
    else
        return (n*factorial(n-1))
}
```

Program 6:

```
/* Fibonacci Series using Recursion*/
#include<stdio.h>
int fibonacii(int);
int main()
{
    int n,f;
    printf("enter a number\n");
    scanf("%d",&n)
    f=fibonacii(n);
    return 0;
}
int fibonacii(int a)
{
    If(a==0)
        return o;
    else if(a==1)
        return 1;
    else
        return fibonacii(a-1)*fibonacii(a-2)
}
```

Function

Program 7:

```
/* Program to reverse a number using function */
#include <stdio.h>
int rev(int);
int main()
{
    int Num, Rev = 0;
    printf("\nEnter the number to reverse: ");
    scanf("%d", &Num);
    Rev = rev(Num);
    printf("Reverse of %d is = %d\n", Num, Rev);
    return 0;
}

int rev(int Num)
{
    int remainder, Rev = 0;
    while (Num > 0)
    {
        remainder = Num % 10;
        Rev = Rev * 10 + remainder;
        Num = Num / 10;
    }
    return Rev;
}
```