# Assignment-9

1. **Functions without arguments and without return type**

• **check whether the year is Leap year**

#include <stdio.h>  
void leapyear();  
int main() {  
    leapyear();  
    return 0;  
}  
void leapyear()  
{  
     int y;  
   printf("please enter a year to check leap year or not:\n");  
   scanf("%d",&y);  
   if(y%400==0 || y%4==0&&y%100!=0)  
   printf("%d is a leap year",y);  
   else  
printf("%d is not a leap year",y);  
}  
Output-  
please enter a year to check leap year or not:  
1800  
1800 is not a leap year

**• convert binary to hexadecimal**

#include <stdio.h>  
#include<math.h>  
void conversion ();  
  
int main(){  
    printf("\"decimal to hexadecimal conversion\"\n");  
    conversion();  
    return 0;  
}  
void conversion() {  
    int b,a;  
    int hex=0,p=0;  
    printf("enter a binary no. of size 4:\n");  
    scanf("%d",&b);  
    printf("the hexadecimal value for %d is:",b);  
    while(b!=0)  
           {  
            a=b%10\*pow(2,p);  
            hex=hex+a;  
            b=b/10;  
            p++;  
           }  
          
        if(hex<=9){  
        printf("%d",hex);}  
        else{  
        switch(hex)  
        {  
            case 10:  
            printf("A");  
            break;  
            case 11:  
            printf("B");  
            break;  
            case 12:  
            printf("C");  
            break;  
            case 13:  
            printf("D");  
            break;  
            case 14:  
            printf("E");  
            break;  
            case 15:  
            printf("F");  
            break;  
             
        }  
        }  
}  
Output-  
"decimal to hexadecimal conversion"  
enter a binary no. of size 4:  
1100  
the hexadecimal value for 1100 is:C

**• count number of digits in a number**

#include <stdio.h>  
  void count();  
  
int main() {  
      
    printf("count the number of digits in a number\n");  
      count();  
    return 0;  
}  
  
void count()  
{  
     int num,i=0;  
     printf("enter a number:\n");  
     scanf("%d",&num);  
printf("the number of digits in %d is:",num);  
       
     while(num!=0)  
     {  
        num=num/10;   
        i++;  
     }  
     printf("%d",i);  
       
  
}  
Output-  
count the number of digits in a number  
enter a number:  
124577  
the number of digits in 124577 is:6

**b) Functions without arguments and with return type**

**• check Armstrong number or not**

#include <stdio.h>

int arms();

int main()

{

int result;

result=arms();

if(result==1)

printf(" armstrong number");

else

printf(" not armstrong number");

return 0;

}

int arms(){

int n,sum=0;

int num;

printf("enter a number to check armstrong or not:\n");

scanf("%d",&num);

printf("%d number is",num);

while(num!=0){

n=num % 10;

num=num/10;

sum=sum+(n\*n\*n);

}

if(sum==num){

return 0;

}

else{

return 1;

}

}

Output-

enter a number to check armstrong or not:

153

153 number is armstrong number

**• to evaluate the following using loops x + x^3 / 3! + x^5 / 5! +... upto 5 terms**

#include <stdio.h>

#include <math.h>

float cal();

int main()

{

float res;

res=cal();

printf("is %.3f",res);

}

float cal() {

int i=1,j=1,x,t;

float fac=1,sum=1;

printf("enter the value for x:\n");

scanf("%d",&x);

printf("the sum of ");

while(i<10)

{

while(j<i){

fac\*=j;

j++;

}

t=pow(x,i);

sum+=t/fac;

printf("%f\t",t/fac);

i=i+2;

}

return sum;

}

Output-

enter the value for x:

2

the sum of 2.000000 4.000000 1.333333 0.177778 0.012698 is 8.524

**• Convert temperature Fahrenheit to Celsius**

#include <stdio.h>

float temp();

int main()

{

printf("the value in fahrenheit is:%.3f",temp());

return 0;

}

float temp()

{

float c,f;

printf("enter value in centigrade for conversion to fahrenheit:\n");

scanf("%f",&c);

f=9/5\*c+32;

return f;}

output-

enter value in centigrade for conversion to fahrenheit:

5

the value in fahrenheit is:37.000

**c) Functions with arguments and without return type**

**• check prime number or not**

#include <stdio.h>

void prime(int);

int main()

{

int n;

printf("enter a number to determine prime or not:\n");

scanf("%d",&n);

prime(n);

}

void prime(int n) {

int p,i=2;

do{

if(n%i==0||n==1)

{

p=0;

break;

}

i++;

}while(i<n/2);

if(p==0&&n!=2)

printf("%d is not a prime number",n);

else

printf("%d is a prime number",n);

}

Output-

enter a number to determine prime or not:

10

10 is not a prime number

**• find all roots of the quadratic equation**

#include<stdio.h>

#include<math.h>

void quad(int,int,int);

int main()

{

int a,b,c;

printf("enter the value of a,b & c:\n ");

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

quad(a,b,c);

return 0;

}

void quad(int a, int b, int c)

{

int d;

float r1,r2;

d=b\*b-4\*a\*c;

if(d==0)

{

printf("Both roots are equal.\n");

r1=r2=-b/(2.0\*a);

printf("First Root Root1: %f\n",r1);

printf("Second Root Root2: %f\n",r2);

}

else if(d>0)

{

printf("Both roots are real and different.\n");

r1=(-b+sqrt(d))/(2\*a);

r2=(-b-sqrt(d))/(2\*a);

printf("First Root Root1: %f\n",r1);

printf("Second Root root2: %f\n",r2);

}

else

{

printf("Root are imaginary.\n");

}

}

Output-

enter the value of a,b & c:

1 5 6

Both roots are real and different.

First Root Root1: -2.000000

Second Root root2: -3.000000

**• find ASCII number to character and character to ASCII number**

#include <stdio.h>

void conversion(int,char);

int main()

{

int a;

char b;

printf("enter a character:\n");

scanf("%c",&b);

printf("enter a number:\n");

scanf("%d",&a);

conversion(a,b);

return 0;

}

void conversion(int a,char b){

char c;

printf("ASCII value of %c is %d\n", b, b);

c = a;

printf("the character assigned to ASCII value %d is %c", c,c);

}

Output-

enter a character:

d

enter a number:

65

ASCII value of d is 100

the character assigned to ASCII value 65 is A

**d) Functions with arguments and with return type**

**• check perfect or abundant or deficient number**

#include <stdio.h>

int perfect(int);

int main()

{

int n,res;

printf("enter a number to check perfect or not : ");

scanf("%d",&n);

res=perfect(n);

if(res==n)

printf("the number is perfect.");

else if(res>n)

printf("the number is abundant.");

else

printf("the number is deficient.");

return 0;

}

int perfect(int n) {

int i=1,sum=0;

printf("The factors are : ");

while(i<n)

{

if(n%i==0)

{

sum=sum+i;

printf("%d\t",i);

}

i++;

}

printf("The sum of the factors is : %d\n",sum);

return sum;

}

Output-

enter a number to check perfect or not : 28

The factors are : 1 2 4 7 14

The sum of the factors is : 28

the number is perfect.

**• calculate factorial of a number**

#include<stdio.h>

int fac(int);

int main()

{

int n,res;

printf("enter a number:\n");

scanf("%d",&n);

res=fac(n);

printf("the factorial of %d is: %d",n,res);

return 0;

}

int fac(int n)

{

int f=1;

while(n!=0)

{

f=f\*n;

n--;

}

return f;

}

Output-

enter a number:

6

the factorial of 6 is: 720

**• count number of digits in a number**

#include <stdio.h>

int count(int);

int main() {

int num,res;

printf("count the number of digits in a number\n");

printf("enter a number:\n");

scanf("%d",&num);

res=count(num);

printf("the number of digits in %d is %d",num,res);

return 0;

}

int count(int num)

{

int i=0;

while(num!=0)

{

num=num/10;

i++;

}

return i;

}

Output-

count the number of digits in a number

enter a number:

1234

the number of digits in 1234 is 4

**e) Function return Multiple values**

** Largest and Smallest of five numbers**

#include <stdio.h>

int max(int num1, int num2);

int min(int num1, int num2);

int main()

{

int num1, num2, maximum, minimum;

printf("Enter any two numbers:");

scanf("%d%d", &num1,&num2);

maximum = max(num1, num2);

minimum = min(num1, num2);

printf("\nMaximum = %d\n", maximum);

printf("Minimum = %d", minimum);

return 0;

}

int max(int num1, int num2)

{

return (num1 > num2 ) ? num1 : num2;

}

int min(int num1, int num2)

{

return (num1 > num2 ) ? num2 : num1;

}

Output-

Enter any two numbers:3 4

Maximum = 4

Minimum = 3

** Find Simple interest and compound interest**

** simple calculator (add, sub, mul, div, mod)**

#include <stdio.h>

float add(float num1, float num2);

float sub(float num1, float num2);

float mult(float num1, float num2);

float div(float num1, float num2);

int main()

{

char op;

float num1, num2, result=0.0f;

scanf("%f %c %f", &num1, &op, &num2);

switch(op)

{

case '+':

result = add(num1, num2);

break;

case '-':

result = sub(num1, num2);

break;

case '\*':

result = mult(num1, num2);

break;

case '/':

result = div(num1, num2);

break;

default:

printf("Invalid operator");

}

printf("%.2f %c %.2f = %.2f", num1, op, num2,

result);

return 0;

}

float add(float num1, float num2)

{

return num1 + num2;

}

float sub(float num1, float num2)

{

return num1 - num2;

}

float mult(float num1, float num2)

{

return num1 \* num2;

}

float div(float num1, float num2)

{

return num1 / num2;

}

Output-

6+5

6.00 + 5.00 = 11.00

**f) Nesting of Functions**

** Print the sum of series 1 + 1/2 + 1/3 + 1/4 + ... + 1/N.**

#include <stdio.h>

double sum(int n)

{

double i, s = 0.0;

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

s = s + 1/i;

return s;

}

int main()

{

int n;

printf("enter the number of terms:\n");

scanf("%d",&n);

printf("Sum is %f", sum(n));

return 0;

}

Output-

enter the number of terms:

8

Sum is 2.717857

** Find GCD and LCM of numbers**

#include <stdio.h>

long gcd(long, long);

int main()

{

long x, y, hcf, lcm;

printf("Enter two integers\n");

scanf("%ld%ld", &x, &y);

hcf = gcd(x, y);

lcm = (x\*y)/hcf;

printf("GCD of %ld and %ld = %ld\n", x, y, hcf);

printf("LCM OF %ld and %ld = %ld\n", x, y, lcm);

return 0;

}

long gcd(long x, long y) {

if (x == 0) {

return y;

}

while (y != 0) {

if (x > y)

x = x - y;

else

y = y - x;

}

return x;

}

Output-

Enter two integers

56 8

GCD of 56 and 8 = 8

LCM OF 56 and 8 = 56

** reverse a number**

#include <stdio.h>

long reverse(long);

int main()

{

long n, r;

printf("Enter a number: \n");

scanf("%ld", &n);

r = reverse(n);

printf("the reverse of the number is:%ld\n", r);

return 0;

}

long reverse(long n)

{

static long r = 0;

if (n == 0)

return 0;

r = r \* 10;

r = r + n % 10;

reverse(n/10);

return r;

}

Output-

Enter a number:

1234

the reverse of the number is:4321

**g) Recursive Functions**

** to Print Fibonacci Series**

#include<stdio.h>

int Fibonacci(int);

int main()

{

int n, i = 0, c;

printf("Enter no. of terms of fibonacci series:\n");

scanf("%d",&n);

printf("Fibonacci series\n");

for ( c = 1 ; c <= n ; c++ )

{

printf("%d\t", Fibonacci(i));

i++;

}

return 0;

}

int Fibonacci(int n)

{

if ( n == 0 )

return 0;

else if ( n == 1 )

return 1;

else

return ( Fibonacci(n-1) + Fibonacci(n-2) );

}

Output-

Enter no. of terms of fibonacci series:

7

Fibonacci series

0 1 1 2 3 5 8

** to print even or odd numbers in given range**

#include <stdio.h>

void EvenAndOdd(int Val, int n);

int main()

{

int n;

printf("Input the range to print starting from 1 :");

scanf("%d", &n);

printf("\n All even numbers from 1 to %d are :",

n);

EvenAndOdd(2, n);

printf("\n All odd numbers from 1 to %d are :", n);

EvenAndOdd(1, n);

printf("\n");

return 0;

}

void EvenAndOdd(int Val, int n)

{

if(Val >n)

return;

printf("%d ", Val);

EvenAndOdd(Val+2, n);

}

Output-

Input the range to print starting from 1 :10

All even numbers from 1 to 10 are :2 4 6 8 10

All odd numbers from 1 to 10 are :1 3 5 7 9

** to convert a decimal number to binary**

#include <stdio.h>

int decimalToBinary(int n)

{

if (n == 0)

{

return 0;

}

else

{

return (n % 2 + 10 \*

decimalToBinary(n / 2));

}

}

int main()

{

int num;

printf("Enter decimal number:");

scanf("%d", &num);

printf ("Binary form of given decimal number is:%d",decimalToBinary(num));

return 0;

}

Output-

Enter decimal number:56

Binary form of given decimal number is:111000

**h) Passing 1D Array in Functions**

**• Reverse the elements of an array**

#include<stdio.h>

void reverse(int[],int);

int main() {

int a[20];

int n,i;

printf("enter the size of array\n");

scanf("%d",&n);

printf("enter the values for the array\n");

for(i=0;i<n;i++)

{

printf("a[%d]=",i);

scanf("%d",&a[i]);

}

reverse(a,n);

return 0;

}

void reverse(int a[],int n)

{

int i;

printf("the array is: ");

for(i=0;i<n;i++)

{

printf("%d\t",a[i]);

}

printf("\nthe array in reverse order is:");

for(i=n-1;i>-1;i--)

{

printf("%d\t",a[i]);

}

}

Output-

enter the size of array

5

enter the values for the array

a[0]=1

a[1]=2

a[2]=3

a[3]=4

a[4]=5

the array is: 1 2 3 4 5

the array in reverse order is:5 4 3 2 1

**• Find the fourth largest and Third smallest element in an array**

#include<stdio.h>

void large(int[],int);

int main() {

int a[20];

int n,i;

printf("enter the size of array\n");

scanf("%d",&n);

printf("enter the values for the array\n");

for(i=0;i<n;i++)

{

printf("a[%d]=",i);

scanf("%d",&a[i]);

}

large(a,n);

return 0;

}

void large(int a[],int n)

{

int i,j ,temp;

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++){

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("the 3rd largest number is %d\n",a[2]);

printf("the 4th largest number is %d",a[3]);

}

Output-

enter the size of array

5

enter the values for the array

a[0]=45

a[1]=10

a[2]=98

a[3]=76

a[4]=8

the 3rd largest number is 45

the 4th largest number is 10

1. **Passing 2D Array in Functions**

** Sum of upper triangular and lower triangular elements of mxm array**

#include <stdio.h>

void trimatrix(int[3][3]);

int main()

{

int a[3][3],i,j,s=0;

printf("Enter Elements for 3\*3 Matrix:\n\n");

for(i=0;i<=2;i++)

for(j=0;j<=2;j++)

{

scanf("%d",&a[i][j]);

}

printf("\n3\*3 Matrix :\n\n");

for(i=0;i<=2;i++)

{

for(j=0;j<=2;j++)

{

printf("%d ",a[i][j]);

}

printf("\n");

}

trimatrix(a);

return 0;

}

void trimatrix(int a[3][3])

{

int i,j,s=0;

for(i=0;i<=1;i++)

{

for(j=0;j<=2;j++)

{

if(i<j)

{

s=s+a[i][j];

}

}

printf("\n ");

}

printf("\nSum of Upper Triangular Elements = %d\n",s);

for(i=2;i>0;i--)

{

for(j=2;j>0;j--)

{

if(i>j)

{

s=s+a[i][j];

}

}

printf("\n");

}

printf("\nSum of Lower Triangular Elements = %d",s);

}

Output-

Enter Elements for 3\*3 Matrix:

1 2 3 4 5 6 7 8 9

3\*3 Matrix :

1 2 3

4 5 6

7 8 9

Sum of Upper Triangular Elements = 11

Sum of Lower Triangular Elements = 19

** Find the maximum & minimum element in each row and each column of mxm array**

#include<stdio.h>

void smallestInRow(int mat[3][3]) ;

void smallestInCol(int mat[3][3]);

int main()

{

int i ,j ,mat[3][3];

printf("Enter Elements for 3\*3 Matrix:\n\n");

for(i=0;i<=2;i++)

for(j=0;j<=2;j++)

{

scanf("%d",&mat[i][j]);

}

printf("\n3\*3 Matrix :\n\n");

for(i=0;i<=2;i++)

{

for(j=0;j<=2;j++)

{

printf("%d ",mat[i][j]);

}

printf("\n");

}

printf("Minimum element of each row is ");

smallestInRow(mat);

printf("\nMinimum element of each column is ");

smallestInCol(mat);

return 0;

}

void smallestInRow(int mat[3][3])

{

for (int i = 0; i < 3; i++) {

int minm = mat[i][0];

for (int j = 1; j < 3; j++) {

if (mat[i][j] < minm)

minm = mat[i][j];

}

printf("%d\t" ,minm);

}

}

void smallestInCol(int mat[3][3])

{

for (int i = 0; i < 3; i++) {

int minm = mat[0][i];

for (int j = 1; j < 3; j++) {

if (mat[j][i] < minm)

minm = mat[j][i];

}

printf("%d\t",minm);

}

}

Output-

Enter Elements for 3\*3 Matrix:

1 2 3 4 5 6 7 8 9

3\*3 Matrix :

1 2 3

4 5 6

7 8 9

Minimum element of each row is 1 4 7

Minimum element of each column is 1 2 3

** Perform matrix multiplication between two mxn array**

#include <stdio.h>

void getMatrixElements(int matrix[5][5], int row , int column) {

printf("\nEnter elements: \n");

for (int i = 0; i < row; ++i) {

for (int j = 0; j < column; ++j) {

printf("Enter a%d%d: ", i + 1, j + 1);

scanf("%d", &matrix[i][j]);

}

}

}

void multiplyMatrices(int first[][10],

int second[][10],

int result[][10],

int r1, int c1, int r2, int c2) {

for (int i = 0; i < r1; ++i) {

for (int j = 0; j < c2; ++j) {

result[i][j] = 0;

}

}

for (int i = 0; i < r1; ++i) {

for (int j = 0; j < c2; ++j) {

for (int k = 0; k < c1; ++k) {

result[i][j] += first[i][k] \* second[k][j];

}

}

}

}

void display(int result[][10], int row, int column) {

printf("\nOutput Matrix:\n");

for (int i = 0; i < row; ++i) {

for (int j = 0; j < column; ++j) {

printf("%d ", result[i][j]);

if (j == column - 1)

printf("\n");

}

}

}

int main() {

int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;

printf("Enter rows and column for the first matrix: ");

scanf("%d %d", &r1, &c1);

printf("Enter rows and column for the second matrix: ");

scanf("%d %d", &r2, &c2);

while (c1 != r2) {

printf("Error! Enter rows and columns again.\n");

printf("Enter rows and columns for the first matrix: ");

scanf("%d%d", &r1, &c1);

printf("Enter rows and columns for the second matrix: ");

scanf("%d%d", &r2, &c2);

}

getMatrixElements(first, r1, c1);

getMatrixElements(second, r2, c2);

multiplyMatrices(first, second, result, r1, c1, r2, c2);

display(result, r1, c2);

return 0;

}

Output-

Enter rows and column for the first matrix: 2 2

Enter rows and column for the second matrix: 2 2

Enter elements:

Enter a11: 1

Enter a12: 2

Enter a21: 3

Enter a22: 4

Enter elements:

Enter a11: 4

Enter a12: 3

Enter a21: 2

Enter a22: 1

Output Matrix:

4 3

0 0

**j) Passing Strings in Functions**

** to perform Substring Extraction (With and Without String Handling Functions).**

** to read a string and prints if it is a palindrome or not.**

#include <stdio.h>

#include <string.h>

void palindrome(char \*string);

int main()

{

char string[20];

printf("Enter a string:");

scanf("%s", string);

palindrome(string);

return 0;

}

void palindrome(char \*string){

int i, length;

int flag = 0;

length = strlen(string);

for(i=0;i < length ;i++)

{

if(string[i] != string[length-i-1])

{

flag = 1;

break;

}

}

if (flag) {

printf("%s is not a palindrome", string);

}

else {

printf("%s is a palindrome", string);

}

}

Output-

Enter a string:iilllllii

iilllllii is a palindrome