# **Practical No : 1**

# **Problem Statement : Introduction to pointers. Write a C program to swap two values using Call by Value and Call by reference.**

# **Code :**

#include<stdio.h>

void swapf(int x, int y){

int c;

c=x;

x=y;

y=c;

printf("Enter the value of x:%d",x);

printf("\nEnter the value of y:%d",y);

}

void bubble(int \*a, int \*b){

int c;

c=\*a;

\*a=\*b;

\*b=c;

}

void main(){

int a,b;

printf("Enter the value of a:");

scanf("%d",&a);

printf("Enter the value of b:");

scanf("%d",&b);

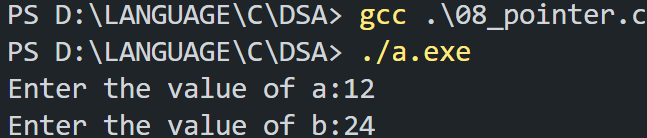
swapf(a,b);

bubble(&a,&b);

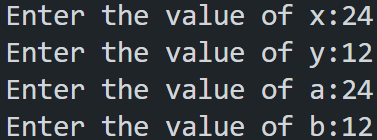
printf("\nEnter the value of a:%d",a);

printf("\nEnter the value of b:%d",b);

}

**Input :**

**Output :**

****

# **Practical No : 2**

# **Problem Statement :**

# **Code :**

//paste your code here.

# **Input :**

**Output :**

# **Practical No : 3**

# **Problem Statement : Implement a program for stack that performs following operations using array.**

# **(a) PUSH (b) POP (c) PEEP (d) CHANGE(Replace top of stack value) (e) DISPLAY**

# **Code :**

#include<stdio.h>

#define size 5

int a[size],x,top=-1,c;

int isfull()

{

if(top==size-1) return 1;

else return 0;

}

int isempty()

{

if(top==-1) return 1;

else return 0;

}

void push(int x)

{

if(isfull())

{

printf("\nArray is full!!");

}

else{

top++;

a[top]=x;

}

}

void peep()

{

x=a[top];

printf("The top value of array is: %d",x);

}

int pop()

{

if(isempty()){

printf("Array is empty.");}

else{

x=a[top];

top--;

}

}

void display()

{

int i;

if(!isempty()){

printf("The Elements Of Array: ");

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

{

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

}

}

}

void main()

{

do

{

printf("\nStack Operations:");

printf("\n1.Push \n2.Peep \n3.Pop \n4.Display");

printf("\nEnter your choice:");

scanf("%d",&c);

switch(c)

{

case 1:

printf("\nEnter The value of Element:");

scanf("%d",&x);

push(x);

break;

case 2:

peep();

break;

case 3:

pop();

break;

case 4:

display();

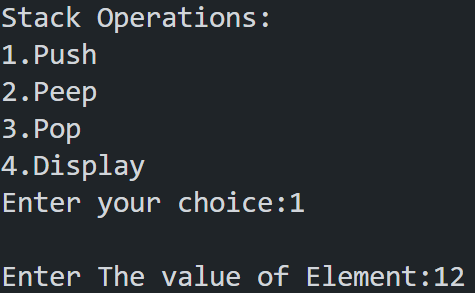
break;

}

} while(c!=0);

}

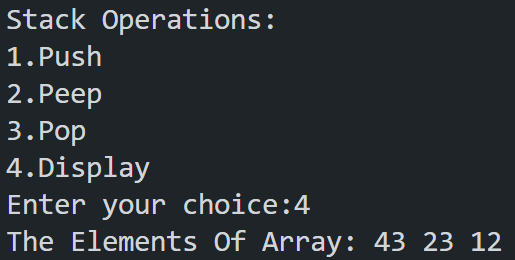
**Input :**

****

****

****

**Output :**

****