Tushar Monga - 35214803119

# EXPERIMENT 4

## Question 1 : Program to perform push, pop, peek and display functions of a stack made on an array.

```
#include<stdio.h>

int top=-1;
int arr[1000];

void Push(){
   if(top==999)
   {
      //if stack is full
      printf("Stack Overflow");
      return;
   }

   int ele;
   printf("Enter element : ");
   scanf("%d",&ele);
   //increasing top
   top=top+1;
   //setting element at top
   arr[top]=ele;
}
```

```
int Pop() {
   //checked conditions before calling
   //removing the top element and return it
   int rele=arr[top];
   top--;
   return rele;
}

int Peek() {
   //conditions checked before calling
   //top element
   return arr[top];
}

void Display() {
   printf("[");

//if empty.. it would show empty brackets
   for(int i=0;i<=top;i++) {
      if(i!=top)
         printf("%d, ",arr[i]);
      else
         printf("%d",arr[i]);
   }

   printf("]\n");
}
```

```
void main() {

   printf("1. Push\n");
   printf("2. Pop\n");
   printf("3. Peek\n");
   printf("4. Display\n");
   printf("5. Exit\n");

   int choice=0;

   while(choice!=5)
   {
      //taking user choice input
      printf("\n\nEnter the choice : ");
      scanf("%d",&choice);
      int ele;

      switch(choice) {

      case 1 : Push();
            break;
      case 2 : if(top==-1)
            {
               //if stack is empty
              printf("Stack Underflow");
            }
            else
```

```
        {
           ele=Pop();
           printf("Removed Element : %d",ele);
        }

        break;
   case 3 : if(top==-1)
        {
           //is stack is empty
          printf("Stack Underflow");
        }
        else
        {
           ele=Peek();
           printf("Removed Element : %d",ele);
        }
        break;

   case 4 : Display();
        break;

   case 5 : //it will exit out of while loop with choice=5
        break;
   default : printf("Invalid\n");
   }
}
```

**OUTPUT**

```
1. Push
2. Pop
3. Peek
4. Display
5. Exit


Enter the choice : 1
Enter element : 7


Enter the choice : 1
Enter element : 14


Enter the choice : 3
Element on top : 14

Enter the choice : 1
Enter element : 77


Enter the choice : 4
[7, 14, 77]


Enter the choice : 1
Enter element : 74


Enter the choice : 2
Removed Element : 74

Enter the choice : 4
[7, 14, 77]


Enter the choice : 5

Process returned 5 (0x5)   execution time : 45.904 s
Press any key to continue.
```

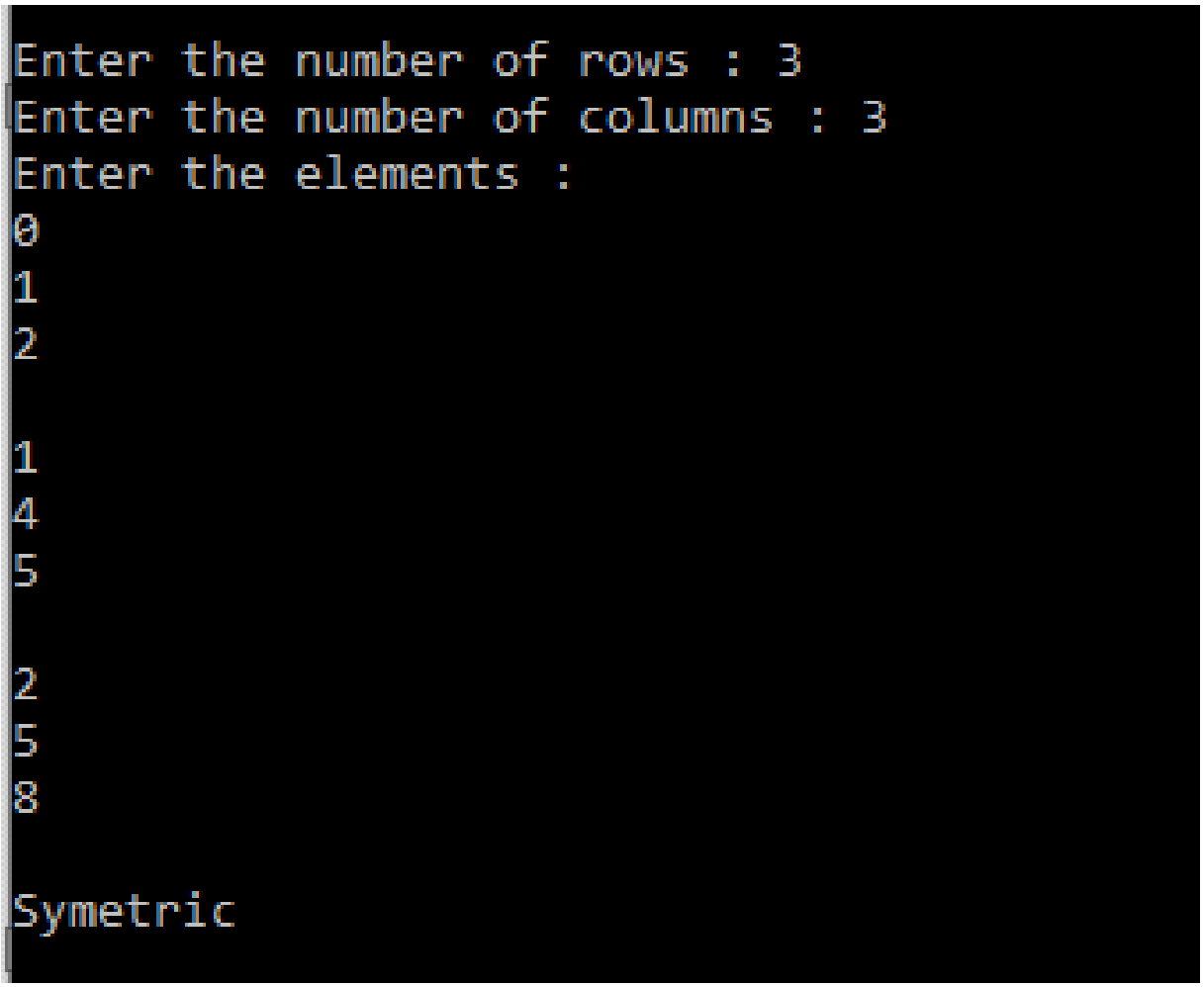
Tushar Monga - 35214803119

# Question 2 : Program to check whether a given matrix is symmetric or not.

```
#include<stdio.h>

void main()
{
   int arr[100][100],i,j,m,n,flag=0;

   //taking no of rows as input
   printf("Enter the number of rows : ");
   scanf("%d",&n);

   //taking no of columns as input
   printf("Enter the number of columns : ");
   scanf("%d",&m);

   //taking no of non zero elements as input
   printf("Enter the elements : \n");
   for(i=0;i<n;i++)
   {
      for(j=0;j<m;j++)
         scanf("%d",&arr[i][j]);
   }
```

```
    if(m!=n)
        //if it is not of same order
        printf("\nNot Symetric\n");
    else
    {
        for(i=0;i<n;i++)
        {
            for(j=0;j<m;j++)
            {
                if(i+j<n)
                {
                    if(arr[i][j]!=arr[j][i])
                    {
                        flag=1;
                        break;
                    }
                }
            }
        }
    }
    if(flag==1)
        printf("\nNot Symetric\n");
    else
        printf("\nSymetric\n");
}
```

**OUTPUT**


Enter the number of rows : 3
Enter the number of columns : 3
Enter the elements :
0
1
2
1
4
5
2
5
8
Symetric