CS8381 DATA STRUCTURES LABORATORY

EX NO: 1 (A)	ARRAY IMPLEMENTATION OF STACK AND QUEUE ADT

AIM:

To write a program for stack using array implementation.

ALGORITHM:

Step1 :Define a array which stores stack elements..
Step 2 : The operations on the stack area)PUSH data into the stack b)POP data out of stack
Step 3 : PUSH DATA INTO STACK
3a :Enter the data to be inserted into stack.
3b :If TOP is NULLthe input data is the first node in stack.the link of the node is NULL.TOP points tothat node.
3c :If TOP is NOT NULLthe link of TOP points to the new node.TOP points to that node.
Step 4:POP DATA FROM STACK 4a.If TOP is NULLthe stack is empty
4b:If TOP is NOT NULLthe link of TOP is the current TOP.the pervious TOP is popped from stack.
Step 5: The stack represented by linked list is traversed to display its content.

PROGRAM:

```
#include<stdio.h>
int stack[100],choice,n,top,x,i;
void push(void);
void pop(void);
void
display(void); int
main()
{
  clrscr();
   top=-1;
   printf("\n Enter the size of STACK[MAX=100]:");
   scanf("%d",&n);
   printf("\n\t STACK OPERATIONS USING ARRAY");
   printf("\n\t----------");
   printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t
   4.EXIT"); do
   {
      printf("\n Enter theChoice:");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
         {
            push();
```

```
            break;
          }
        case 2:
        {
            pop();
            break;
        }
        case 3:
        {
            display();
            break;
        }
        case 4:
        {
            printf("\n\t EXIT POINT ");
            break;
        }
        default:
        {
        printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
        }
      }
   }
   while(choice!=4);
   return 0;
}
void push()
{
   if(top>=n-1)
   {
      printf("\n\tSTACK is over flow");
   }
   else
   {
      printf(" Enter a value to be pushed:");
      scanf("%d",&x);
      top++;
      stack[top]=x;
   }
}
void pop()
{
   if(top<=-1)
   {
      printf("\n\t Stack is under flow");
   }
   else
   {
      printf("\n\t The popped elements is %d",stack[top]);
```

```
        top—;
    }
    }
    void display()
    {
       if(top>=0)
       {
          printf("\n The elements in STACK \n");
          for(i=top; i>=0; i—)
                printf("\n%d",stack[i]);
          printf("\n Press Next Choice");
       }
        else
       {
          printf("\n The STACK is empty");
       }
    }
```

Output

```
Enter the size of STACK[MAX=100]:10
STACK OPERATIONS USING ARRAY
        1.PUSH
        2.POP
        3.DISPLAY
        4.EXIT
Enter the Choice:1
Enter a value to be pushed:12
Enter the Choice:1
Enter a value to be pushed:24
Enter the Choice:1
Enter a value to be pushed:98
Enter the Choice:3
The elements in STACK
982412
 Enter the Choice:4
          EXIT POINT
```

Result:

Thus the program for stack using array implementation is executed sucessfully & verified.

EX NO: 1 (B)	ARRAY IMPLEMENTATION OF QUEUE ADT

Aim:

To write a program for Queue using array implementation.

Algorithm :

Step1:Define a array which stores queue elements.

Step 2: The operations on the queue are

a.INSERT data into the queue

b.DELETE data out of queue

Step 3: INSERT DATA INTO queue

a.Enter the data to be inserted into queue.

b.If TOP is NULLthe input data is the first node in queue.the link of the node is NULL.TOP points to

that node.

c.If TOP is NOT NULLthe link of TOP points to the new node.TOP points to that node.

Step 4: DELETE DATA FROM queue

a.If TOP is NULLthe queue is empty

b.If TOP is NOT NULLthe link of TOP is the current TOP.the pervious TOP is popped from queue.Step

Step 5. The queue represented by linked list is traversed to display its content.

Program:

```
#include<stdio.h>
#include<conio.h>
#define n 5
void main()
{
   int queue[n],ch=1,front=0,rear=0,i,j=1,x=n;
   //clrscr();
   printf("Queue using Array");
   printf("\n1.Insertion \n2.Deletion \n3.Display \n4.Exit");
   while(ch)
   {
      printf("\nEnter the Choice:");
      scanf("%d",&ch);
      switch(ch)
      {
      case 1:
if(rear==x)
         printf("\n Queue is Full");
      else
      {
         printf("\n Enter no %d:",j++);
         scanf("%d",&queue[rear++]);
      }
```

```
            break;
        case 2:
            if(front==rear)
            {
                printf("\n Queue is empty");
            }
            else
            {
                printf("\n Deleted Element is %d",queue[front++]);
                x++;
            }
            break;
        case 3:
            printf("\n Queue Elements are:\n ");
            if(front==rear)
                printf("\n Queue is Empty");
            else
            {
                for(i=front; i<rear; i++)
                {
                    printf("%d",queue[i]);
                    printf("\n");
                }
                break;
            case 4:
                exit(0);
            default:
                printf("Wrong Choice: please see the options");
            }
        }
    }
    getch();
}
```

Output

```
Queue using Array
        1.Insertion
        2.Deletion
        3.Display
        4.Exit
        Enter the Choice:1
        Enter no 1:10
        Enter the Choice:1
        Enter no 2:54
        Enter the Choice:1
        Enter no 3:98
        Enter the Choice:1
        Enter no 4:234
        Enter the Choice:3
        Queue Elements are:
        10
        54
        98
        234
        Enter the Choice:2
        Deleted Element is 10
        Enter the Choice:3
        Queue Elements are:
        54
        98
        234
        Enter the Choice:4
```

Result:

Thus the program forQueue using array implementation is executed sucessfully and verified.