Program to implement Stack Using Two Queues

```
#include <stdio.h>
#define size 100
int f,r;
f = -1;
r= -1;
int q1[size];
int q2[size];

void printStack(){
   int i;
   for( i = f ; i<= r ; ++i){
      printf("%d ",q1[i]);
   }
}


void push(int x){
   int i;
   if(r== -1){
      q1[++r] = x;
      f++;


   }

   else{

      for(i = f ; i <= r ; ++i){
         q2[i] = q1[i];
      }
      q1[f] = x;
      for(i = f+1 ; i <= r+1 ; ++i){
         q1[i] = q2[i-1];
      }
      r++;
      printf("\n");
      printStack();
   }

}


void pop(){
   if(f==-1){
      printf("Error Popping");
      exit(0);
   }
   else{

      printf("Popped Element : %d\nRemaining Elements : ",q1[f++]);
```

```
        int i;
        printStack();
        if(f > r){
            f = r = -1;
        }

    }
}


int main()
{

    int n;
    //pop();
    printf("Enter No of operations  : ");
    scanf("%d",&n);
    for(int i = 0 ; i < n ; ++i){
        int k;
        printf("\n\n1 to push \n2 to pop\n3 to view Stack : ");
        scanf("%d",&k);
        if(k == 1){
            int l;
            printf("\nEnter the element to be pushed : ");
            scanf("%d",&l);
            push(l);
        }
        else if(k == 2){
            pop();
        }
        else if(k == 3){
            printStack();
        }
        else{
            printf("\nInvalid Option\n");
            i = i -1;
        }
    }

    printf("\nFinal Contents Of The Stack:\n");
    printStack();


    return 0;
}
```

Output :

```
Enter No of operations  : 7


1 to push
```

```
2 to pop
3 to view Stack : 1

Enter the element to be pushed : 1


1 to push
2 to pop
3 to view Stack : 1

Enter the element to be pushed : 2



1 to push
2 to pop
3 to view Stack : 1

Enter the element to be pushed : 3



1 to push
2 to pop
3 to view Stack : 1

Enter the element to be pushed : 4



1 to push
2 to pop
3 to view Stack : 3
4 3 2 1

1 to push
2 to pop
3 to view Stack : 2
Popped Element : 4
Remaining Elements : 3 2 1

1 to push
2 to pop
3 to view Stack : 1

Enter the element to be pushed : 7


Final Contents Of The Stack:
7 3 2 1
```

FLOWCHART FOR STACK USING TWO QUEUES

Here q1 and q2 are two queues used for performing stack operations.

push (E element)

- if q1 is empty, enqueue E to q1
- if q1 is not empty, enqueue all elements from q1 to q2, then enqueue E to q1and enqueue all elements from q2 back to q1

pop()

- dequeue an element from q1

Sequence of operations
Enqueue(1)
Enqueue(2)
Enqueue(3)
Enqueue(4)
pop()
Enqueue(5)

Stack Initially :

Stack				

Enqueue(1):

Directly Enqueue new element – 1

Q1	1			

Q2				

Stack	1			

Enqueue(2) :

Copy the elements from q1 to q2

Q1				

Q2	1			

Enqueue new element - 2

Q1	2			

Q2	1			

Enqueue the Q2 elements

Q1	2	1		

Q2				

Stack :

Stack	2	1		

Enqueue(3) :

Copy the elements from q1 to q2

Q1				

Q2	2	1		

Enqueue new element - 3

Q1	3			

Q2	2	1		

Enqueue the Q2 elements

Q1	3	2	1	

Q2				

Stack :

Stack	3	2	1	

Enqueue(4) :

Copy the elements from q1 to q2

Q1				

Q2	3	2	1	

Enqueue new element - 4

Q1	4			

Q2	3	2	1	

Enqueue the Q2 elements

Q1	4	3	2	1

Q2				

Stack :

Stack	4	3	2	1

----

pop()

Stack		3	2	1

Enqueue(5)

Q1		3	2	1

Copy the elements from q1 to q2

Q1				

Q2	3	2	1	

Enqueue new element – 5

Q1	5			

Enqueue the Q2 elements

Q1	5	3	2	1

Q2				

Stack :

Stack	5	3	2	1