LAB ASSIGNMENT 7
Stack and Queue

1. Write a program to implement a stack using two queues.
2. Write a program to implement a queue using two stacks.

Solutions:

1. Write a program to implement a stack
using two queues. #include <stdio.h>

```
#define MAX 100

int q1[MAX], q2[MAX];
int f1 = -1, r1 = -1, f2 = -1, r2 = -1;

void enqueue1(int x) {
  if (r1 == MAX - 1)
  return; if (f1 == -1)
  f1 = 0; q1[++r1] =
  x;
}

int dequeue1() {
  if (f1 == -1 || f1 > r1)
  return -1; return q1[f1++];
}

void enqueue2(int x) {
  if (r2 == MAX - 1)
  return; if (f2 == -1)
  f2 = 0; q2[++r2] =
  x;
}

int dequeue2() {
  if (f2 == -1 || f2 > r2)
  return -1; return q2[f2++];
}
```

```
void push(int x) {
  enqueue2(x);
  while (f1 != -1 && f1 <= r1)
  enqueue2(dequeue1()); int *tf = &f1, *tr
  = &r1, *tf2 = &f2, *tr2 = &r2;
  int *tempf = tf,
  *tempr = tr; f1 =
  *tf2; r1 = *tr2;
  f2 = *tempf; r2 = *tempr;
}

int pop() {
  if (f1 == -1 || f1 > r1)
  return -1; return
  dequeue1();
}

int main() {
  int n, x,
  choice; while
  (1) {
    scanf("%d", &choice);
    if (choice == 1) { scanf("%d", &x); push(x); }
    else if (choice == 2) { x = pop(); if (x != -1) printf("%d\n", x);
else printf("Stack Empty\n"); }
    else break;
  }
  return 0;
}
```

2. Write a program to implement a queue
using two stacks. #include <stdio.h>

```
#define MAX 100

int s1[MAX], s2[MAX];
int top1 = -1, top2 = -1;

void push1(int x) {
  if (top1 == MAX - 1) return;
```

```
   s1[++top1] = x;
}

int pop1() {
   if (top1 == -1)
   return -1; return
   s1[top1--];
}

void push2(int x) {
   if (top2 == MAX - 1) return;
   s2[++top2] = x;
}

int pop2() {
   if (top2 == -1)
   return -1; return
   s2[top2--];
}

void enqueue(int x) {
   push1(x);
}

int dequeue()
   { int x;
   if (top1 == -1 && top2 == -1)
   return -1; if (top2 == -1) {
      while (top1 != -1) push2(pop1());
   }
   return pop2();
}

int main() {
   int x, choice;
   while (1) {
      scanf("%d", &choice);
      if (choice == 1) { scanf("%d", &x); enqueue(x); }
```

```
    else if (choice == 2) { x = dequeue(); if (x != -1) printf("%d\n", x); else printf("Queue Empty\n"); }
    else break;
  }
  return 0;
}
```