LAB 6 LINEAR QUEUE IMPLIMENTATION

```
#include <stdio.h>
#include <stdlib.h>

#define SIZE 5

int item, q[10];
int f=0, r=-1;
int count=0;

void Insert_front(){
    if(count == SIZE-1){
        printf("STACK IS FULL");
```

```
else{
    Pf("Enter Element");
    Scanf("%d", &item);

    r = (r+1) % SIZE;
    q[r] = item;
    count++;
}

void delete_rear(){
    if(count==0){
        Pf("queue Empty");
    }
    else{
        item = q[front];
        front = (front+1) % SIZE;
        count--;
    }
    Pf("The Item Deleted is %d", item);
}

void display(){
    if(count==0){
        Pf("queue empty");
    }
    else{ Pf("The Items are:\n");
        for(i=0; i<count; i++){
            Pf("%d\t", q[i]);
        }
}
```

```
int main(){
    int choice;

for(;;){
printf(" Enter the corresponding choice
        1. Insert_Front \n
        2. Delete_Rear \n
        3. Display \n
        4. Exit \n");

switch(choice){

case 1 : insert_front();
         break;
case 2 : delete_rear();
         break;
case 3; display()
        break;
default : exit(0) ;

}

}
```

* Output :-

⊙ Enter the operations
1. Insert
2. Delete
3. Display
→ 1
→ Enter item
25
⊙ Enter the operations
1. Insert
2. Delete
3. Display
→ 1
→ Enter item
55
⊙ Enter the Operations
1. Insert
2. Delete.
3. Display
→ 3
→ Elements are
25 55
⊙ Enter the operations
1. Insert
2. Delete
3. Display
→ 2.
→ The Element deleted is 25.

LAB-5 : CIRCULAR QUEUE :

```
#include <stdlib.h>
#include <stdio.h>
#define SIZE 5

int item, front=0, rear=-1, q[SIZE],
        count=0;

void insertrear(){
    if (count==SIZE) {
        printf("queue overflow");
    }
    printf("Enter the element");
    scanf("%d", &item);
    rear = (rear+1) % SIZE;
    q[rear] = item;
    count++;

void deletfront(){
    if (count==0) {
        printf("queue is Empty");
    }
    else {
    item = q[front];
    printf("\n item deleted is = %d\n",
                item);
```

```
        front = (front+1) % SIZE;
        count--;
    }
}

void display(){
    int i, f;
    if (count == 0){
        pf("Queue is Empty");
    }
    f = front;
    pf("ITEMS of Queue\n");
    for(i=0; i<count; i++){
        pf("%d\t", q[f]);
        f = (f+1) % SIZE;
    }
}

int main(){
    int choice;
    for(;;){
        pf("\nEnter operations\n
            1. Insert\n2. delete\n
        3. display\n4. Exit");
        scanf("%d", &choice);
```

```
switch (choice) {
        case 1: insertrear()
                break;
        case 2: deletefront()
                break;
        case 3: display();
                break;
        default : exit(0);
    }
}
```

Output:

- Enter operations
  1. insert
  2. Delete
  3. Display
- → 1
- → Enter item to be inserted
- → 45
- Enter operations
  1. insert
  2. Delete
  3. Display
- → 1.
- → Enter item
- → 20
- Enter operations
  1. insert → 3.
  2. Delete → Elements are;
  3. Display 45 20