LAB PROGRAM - 4.

```
#include <stdio.h>
#include <stdlib.h>
#define s 3

int front =-1;
int rear =-1;

int queue[s];

void enque(int, int);
int ~~void~~ deque(int);
void display(int);
int main()
{
    int choice, size;
    int item;
    printf("Enter the SIZE of the queue : \n");
    scanf("%d", &size);
   do
   {
      printf("\n-----MENU------\n");
      printf("1. INSERT to queue \n");
printf("2. Delete From Queue\n");
printf("3. DISPLAY the Queue \n");
printf("4. EXIT \n");
printf("Enter your choice correctly :\n");
scanf("%d", &choice);
switch(choice)
{
    case 1:
      if(((front==0 && rear == SIZE-1)) ||
              (front == rear+1))
      {
         printf("Queue is FULL !\n");
         break;
      }
      printf("Enter the element to insert :\n");
      scanf("%s", &item);
      enque(size, item);
      break;
    case 2:
       item = deque(size);
       if(item==-999)
         printf("Queue is EMPTY \n");
       else
         printf("Removed Element %d\n",item);
       break;
  case 3: display(size); break;
  case 4: printf("EXITING ....\n"); exit(0);
 default: printf("Invalid Choice !\n");
          break;
   }
}
while(choice !=4);
return0;
}
```

```
void enque(int size, int ele)
{
   if(((front==0 && rear == size-1)) || (front==
                                      rear+1)))
   {
      printf("Queue is full");
      return;
   }
   else
   {
      rear = (rear+1)% size;
      queue[rear] = ele;
      if(front ==-1)
       front =0;
   }
}

int deque(int size)
{
   int item;
   if(front ==-1 && rear==-1)
      return -999;
   else
   {
      item = queue[front];
      if(front== rear)
      {
         front=-1;
         rear=-1;
      }
      else
       front=(front+1)% size;
     return item;
   }
}
```

store
67

```
void display(int size)
{
    int i;
    if(front==-1 && rear==-1)
    {
      printf("Queue is Empty \n");
      return;
    }
    else
    {
      printf ("Queue Contents: \n");
      for(i=front; i!=rear; i=(i+1)%size)
         printf(" %d\t ", queue[i]);
      printf("%d\t", queue [i]);
    }
}
```