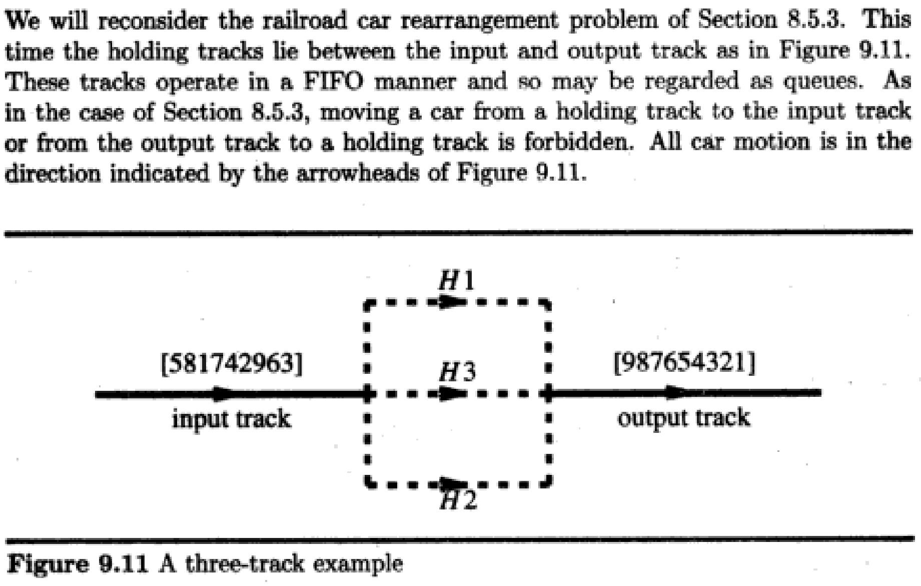
Name	Komal Tarachandani
UID no.	2021600065
Experiment No.	2

AIM:	Implement queue operations basics and also the application of queues in real life.
Program2	
PROBLEM STATEMENT :	 We will reconsider the railroad car rearrangement problem of Section 8.5.3. This time the holding tracks lie between the input and output track as in Figure 9.11. These tracks operate in a FIFO manner and so may be regarded as queues. As in the case of Section 8.5.3, moving a car from a holding track to the input track or from the output track to a holding track is forbidden. All car motion is in the direction indicated by the arrowheads of Figure 9.11. H1 [581742963] H3 [987654321] input track output track H2 **Figure 9.11** A three-track example Task to be done : 1- basic queue operations implementation 2- Implement the above scenario

PROGRAM:

BASIC QUEUE OPERATIONS:

```
#include <iostream>
#include <stdlib.h>
using namespace std;
typedef struct{
    int front,rear;
    int arr[10];
}queue;
queue *q;
int n;
int isFull()
{
    if(q->rear==n-1) return 1;
    else return 0;
}
int isEmpty()
{
    if(q->front>q->rear) return 1;
    else return 0;
}
void enqueue(int ele)
{
    if(q->front==-1)
    {
        cout<<"\nStarting the queue";
        q->front++;
        q->rear++;
        q->arr[q->front]=ele;
    }
    else if(isFull()==1)
    {
        cout<<"STACK OVERFLOW..cannot enque element";
    }
    else
    {
        q->rear++;
        q->arr[q->rear]=ele;
    }
}
int dequeue()
```

```
{
  if(q->front==0)
  {
    cout<<"Dequeing first element";
    int temp=q->arr[q->front];
    q->front++;
    return temp;
  }
  else if(isEmpty()==1)
  {
    cout<<"QUEUE UNDERFLOW....cannot dequeue further";
    return 0;
  }
  else
  {
     int temp=q->arr[q->front];
     q->front++;
     return temp;
  }
}
int peek()
{
  return q->arr[q->front];
}
int main()
{
  q=(queue*)malloc(sizeof(queue));
  q->front=-1;
  q->rear=-1;
  int ch=1,ele,choice;
  cout<<"Enter size of queue: ";
  cin>>n;
  while(ch==1)
  {
    cout<<"MAIN MENU\n1.ENQUEUE\n2.DEQUEUE\n3.PEEK";
    cout<<"\nEnter your choice: ";
    cin>>choice;
    switch(choice)
    {
      case 1:
            {
                cout<<"Enter data to enqueue: ";
                cin>>ele;
                enqueue(ele);
            }
            break;
            case 2:
            {
                ele=dequeue();
                cout<<"\n the dequeued element is: "<<ele;
            }
            break;
            case 3:
            {
                ele=peek();
                cout<<"\n the peek element is: "<<ele;
            }
        }
        cout<<"Enter 1 to continue: ";
        cin>>ch;
    }
    return 0;
}
```

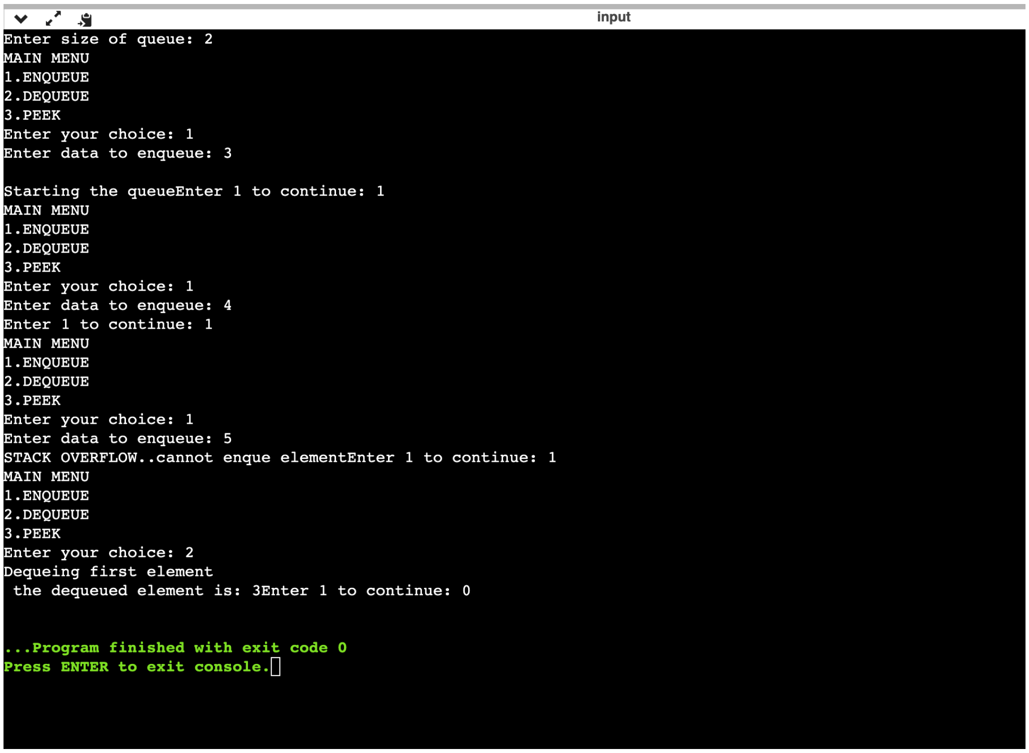
APPLICATION OF QUEUE:

```
#include <iostream>
#include <stdlib.h>
using namespace std;
typedef struct{
    int front,rear;
    int arr[9];
}queue;
queue *q1,*q2,*q3;
void enqueue(queue *q,int num)
{
    if(q->front==-1)
    {
        q->front++;
        q->rear++;
        q->arr[q->front]=num;
    }
    else
    {
        q->rear++;
        q->arr[q->rear]=num;
    }
}
int dequeue(queue *q)
{
    int temp=q->arr[q->front];
    q->front++;
    return temp;
}
void display(queue *q)
{
    q->front=0;
    while(q->front<=q->rear)
    {
        cout<<"enqued element: "<<q->arr[q->front]<<endl;
        q->front++;
    }
}
int otpt[9];
int main()
{
    q1=(queue*)malloc(sizeof(queue));
    q1->front=-1;
    q1->rear=-1;
    q2=(queue*)malloc(sizeof(queue));
    q2->front=-1;
    q2->rear=-1;
    q3=(queue*)malloc(sizeof(queue));
    q3->front=-1;
    q3->rear=-1;
    int inpt[]={3,6,9,2,4,7,1,8,5};
    for(int i=0;i<9;i++)
    {
        enqueue(q1,inpt[i]);
    }
    display(q1);
    q1->front=0;
    int num=1,j=0;
    while(j<=8)
    {
        if(num==q1->arr[q1->front])
        {
            cout<<"\nREMOVED "<<num<<" FROM q1 and added to output
array\n";
            num=dequeue(q1);
            otpt[j]=num;
            j++;
            num++;
        }

        else if(num==q2->arr[q2->front])
        {
            cout<<"\nREMOVED "<<num<<" FROM q2 and added to output
array\n";
            num=dequeue(q2);
            otpt[j]=num;
            j++;
            num++;
        }
        else if(num==q3->arr[q3->front])
        {
            cout<<"\nREMOVED "<<num<<" FROM q3 and added to output
array\n";
            num=dequeue(q3);
            otpt[j]=num;
            j++;
            num++;
        }
        else
        {
            int no=dequeue(q1);
            if(q2->front==-1 || no>q2->arr[q2->rear])
            {
                cout<<"\n"<<no<<" added to q2";
                enqueue(q2,no);
            }
            else
            {
                cout<<"\n"<<no<<" added to q3";
                enqueue(q3,no);
            }
        }

    }
    cout<<"\nOUTPUT ARRAY \n";
    for(int i=0;i<9;i++)
    {
        cout<<otpt[i]<<" ";
    }
    return 0;
}
```

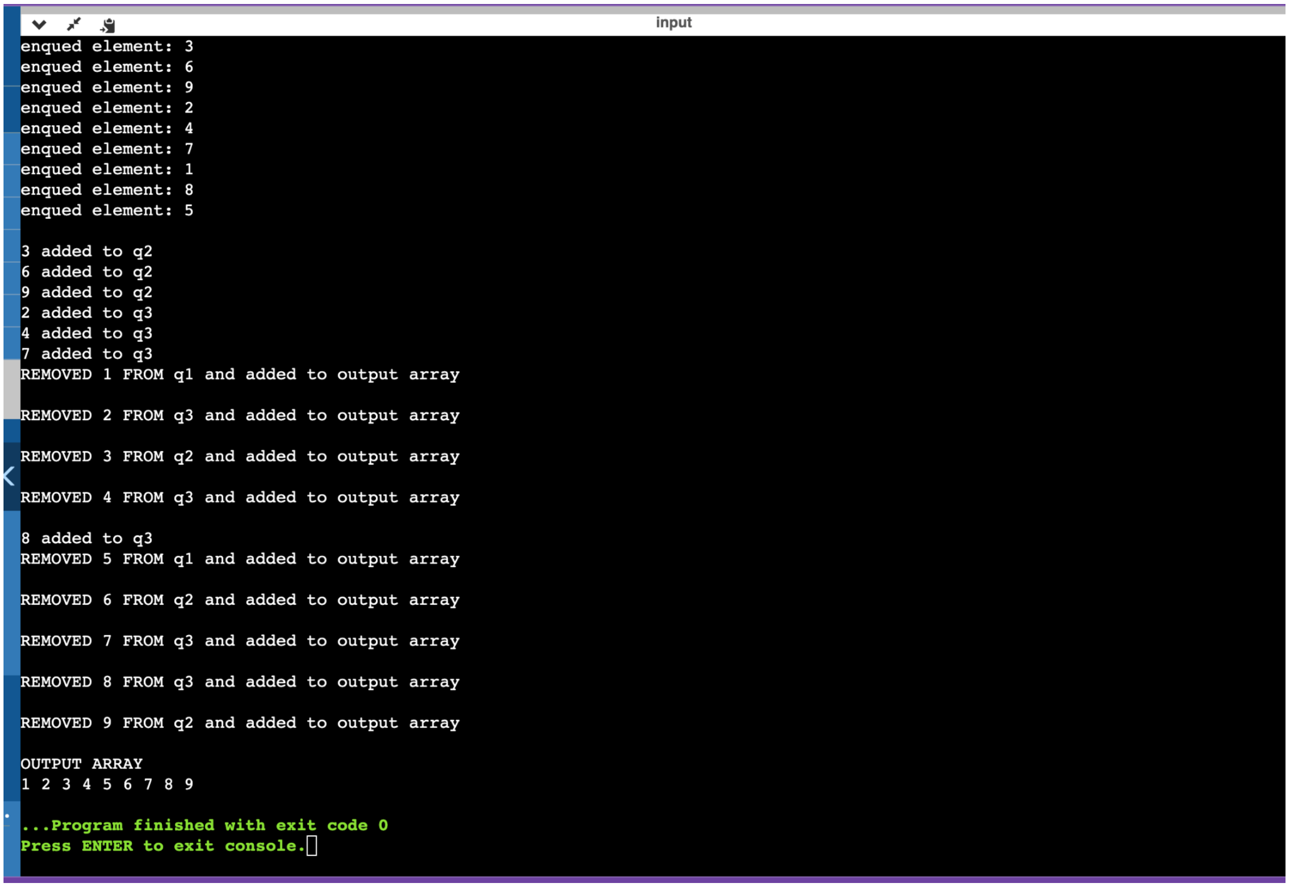
RESULT:



```
Enter size of queue: 2
MAIN MENU
1.ENQUEUE
2.DEQUEUE
3.PEEK
Enter your choice: 1
Enter data to enqueue: 3

Starting the queueEnter 1 to continue: 1
MAIN MENU
1.ENQUEUE
2.DEQUEUE
3.PEEK
Enter your choice: 1
Enter data to enqueue: 4
Enter 1 to continue: 1
MAIN MENU
1.ENQUEUE
2.DEQUEUE
3.PEEK
Enter your choice: 1
Enter data to enqueue: 5
STACK OVERFLOW..cannot enque elementEnter 1 to continue: 1
MAIN MENU
1.ENQUEUE
2.DEQUEUE
3.PEEK
Enter your choice: 2
Dequeing first element
 the dequeued element is: 3Enter 1 to continue: 0


...Program finished with exit code 0
Press ENTER to exit console.
```



```
enqued element: 3
enqued element: 6
enqued element: 9
enqued element: 2
enqued element: 4
enqued element: 7
enqued element: 1
enqued element: 8
enqued element: 5

3 added to q2
6 added to q2
9 added to q2
2 added to q3
4 added to q3
7 added to q3
REMOVED 1 FROM q1 and added to output array

REMOVED 2 FROM q3 and added to output array

REMOVED 3 FROM q2 and added to output array

REMOVED 4 FROM q3 and added to output array

8 added to q3
REMOVED 5 FROM q1 and added to output array

REMOVED 6 FROM q2 and added to output array

REMOVED 7 FROM q3 and added to output array

REMOVED 8 FROM q3 and added to output array

REMOVED 9 FROM q2 and added to output array

OUTPUT ARRAY
1 2 3 4 5 6 7 8 9

...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION:	Hence I was able to learn the proper implementation and application of queues.