**Koneru Lakshmaiah Education Foundation**

**(Deemed to be University)**

FRESHMAN ENGINEERING DEPARTMENT

**A Project Based Lab Report**

**On**

**HOSPITAL MANAGEMENT**

**SUBMITTED BY:**

I.D NUMBER NAME

180030136 S.Rakesh Kumar

180030167 S.Sandeep

180030171 G.Jaya Sravani

180030185 P.Sindhu Gayathri

**UNDER THE GUIDANCE OF**

**Mr.Azmira Krishna**

**ASSISTANT PROFESSOR**



**KL UNIVERSITY**

Green fields, Vaddeswaram – 522 502

Guntur Dist., AP, India.

**DEPARTMENT OF BASIC ENGINEERING SCIENCES-1**

****

**CERTIFICATE**

This is to certify that the project based laboratory report entitled “**HOSPITAL MANAGEMENT**” submitted by Y.N.S.Sruthi bearing Regd.No-180030930; to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in “TECHNICAL SKILLS-II(CODING)”course in I B Tech II Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2018 – 2 019.

PROJECT SUPERVISORHEAD OF THE DEPARTMENT

**Mr.Azmira Krishna** T. Vamsidhar

**Assistant Professor**

**ACKNOWLEDGEMENTS**

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Prof.Dr. N. Venkataram** for his administration towards our academic growth.

I express sincere gratitude to HOD-BES-1 **T. Vamshidar** successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor  **Mr.Azmira Krishna** for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Regd.No: Name

180030136 S.Rakesh Kumar

180030167 S.Sandeep

180030171 G.Jaya Sravani

180030185 P.Sindhu Gayathri

**ABSTRACT**

A queue is a collection in which the entities in the collection are kept in order and the principal operations on the collection are the addition of entities to the rear terminal position, known as enqueue, and removal of entities from the front terminal position, known as dequeue.

Queue is also an abstract data type or a linear data structure in which the first element is inserted from one end called the REAR and the removal of existing element takes place from the other end called as FRONT.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | Introduction | 1 |
| 2 | Aim of the Project | 2 |
| 2.1 | Advantages & Disadvantages | 2 |
| 2.2 | Future Implementation | 2 |
| 3 | Software & Hardware Details | 3 |
| 4 | Algorithm | 4 |
| 5 | Flowchart | 5-8 |
| 6 | Implementation | 9-12 |
| 7 | Integration and System Testing | 13-14 |
| 8 | Conclusion | 15 |

**1**

1. **INTRODUCTION**

**CIRCULAR QUEUE:**

Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. It is also called 'Ring Buffer'. In a normal Queue, we can insert elements until queue becomes full.

**Operations on Circular Queue**:

* **Front:** Get the front item from queue.
* **Rear:** Get the last item from queue.
* **En-Queue(value)**This function is used to insert an element into the circular queue. In a circular queue, the new element is always inserted at Rear position.

**Steps:**

* 1. Check whether queue is Full – Check ((rear == SIZE-1 && front == 0) || (rear == front-1)).
  2. If it is full then display Queue is full. If queue is not full then, check if (rear == SIZE – 1 && front != 0) if it is true then set rear=0 and insert element.
* **De-Queue()** This function is used to delete an element from the circular queue. In a circular queue, the element is always deleted from front position.

**Steps:**

* 1. Check whether queue is Empty means check (front==-1).
  2. If it is empty then display Queue is empty. If queue is not empty then step 3
  3. Check if (front==rear) if it is true then set front=rear= -1 else check if (front==size-1), if it is true then set front=0 and return the element.

2

**AIM**

To implement the patient details by using circular queue.

**Advantages:-**

1. It takes up less memory than the linear queue.

2. A new item can be inserted in the location from where a previous item is deleted.

3. Infinitenumber of elements can be added continuously but deletion must be used.

**Disadvantages:-**

1. A major disadvantag**e** of a classical queue is that a new element can only be inserted when all of the elements are deleted from the queue.

2. This is the major limitation of a classical queue, i.e. even if there is space available at the front of thequeue we can’t use it.

3. As elements are removed from the queue, the portion of the array which is effectively used shifts to the right.

4.A major disadvantage of a classical queue is that a new element can only be inserted when *all* of the elements are deleted from the queue.

**Future enhancements:-**

Recent emerging applications, such as fraud detection in secure trading and banking, network traffic monitoring, sensor network data analysis, Web click stream mining, power consumption measurement, and stock fluctuations’ tracking generate streams of data

**3**

**SYSTEM REQUIREMENTS**

* **SOFTWARE REQUIREMENTS:**

The major software requirements of the project are as follows:

Language : Turbo-C

Operating System**:**Windows XP or later.

* **HARDWARE REQUIREMENTS:**

The hardware requirements that map towards the software are as follows:

RAM : 8 GB

Processor:Intel®core™i5-7200U

4

**ALGORITHM**

**Step-1 :**start

**Step-2 :**read the values

**Step-3:**inserting the elements(enqueue)

F==-1&&r==-1

a[0]=s[0].id

f++,r++

r==10&&f!=0

r==10&&f!=0 || r==f+1

creating the details of the patient by using an array.

**Step-4 :**searching the elements

C!=0&&f==0

I=0,f>0

A[i]==p,j++,i++

Finding the full details of patient

**Step-5** : deleting the details

C==0,c— (there are no patients)

R==0,r--,r=10

Deleted details of the patient

**Step-6:** created the details of the patient

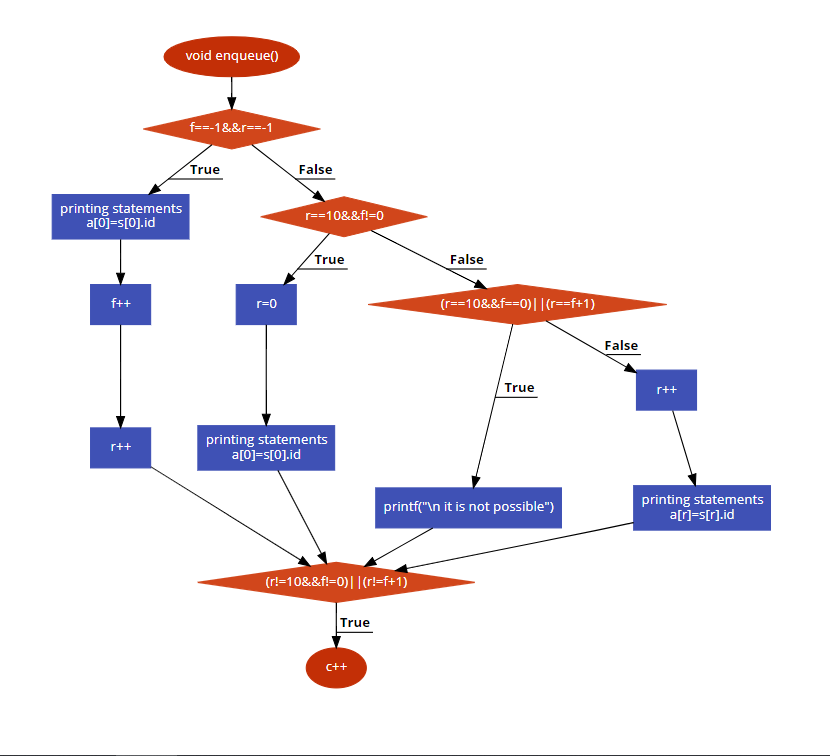
**Step-7 :** finding the details of the patient

**Step-8 :** deleting the details

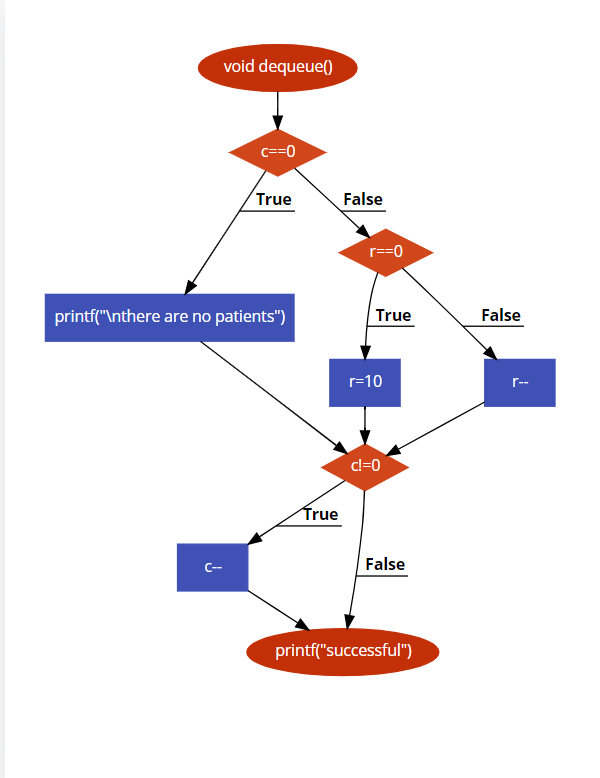
**Step-9 :** Stop

**5**

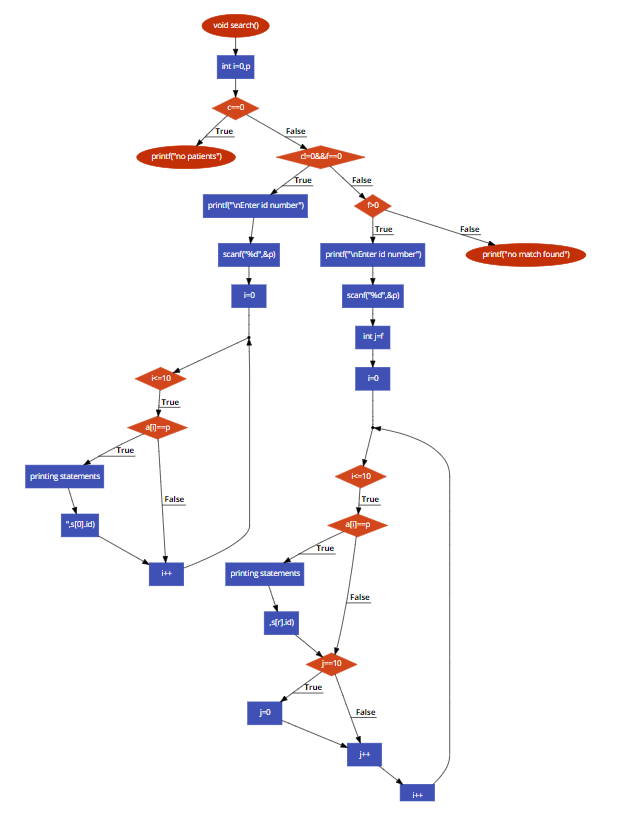
**Flow Chart**

****

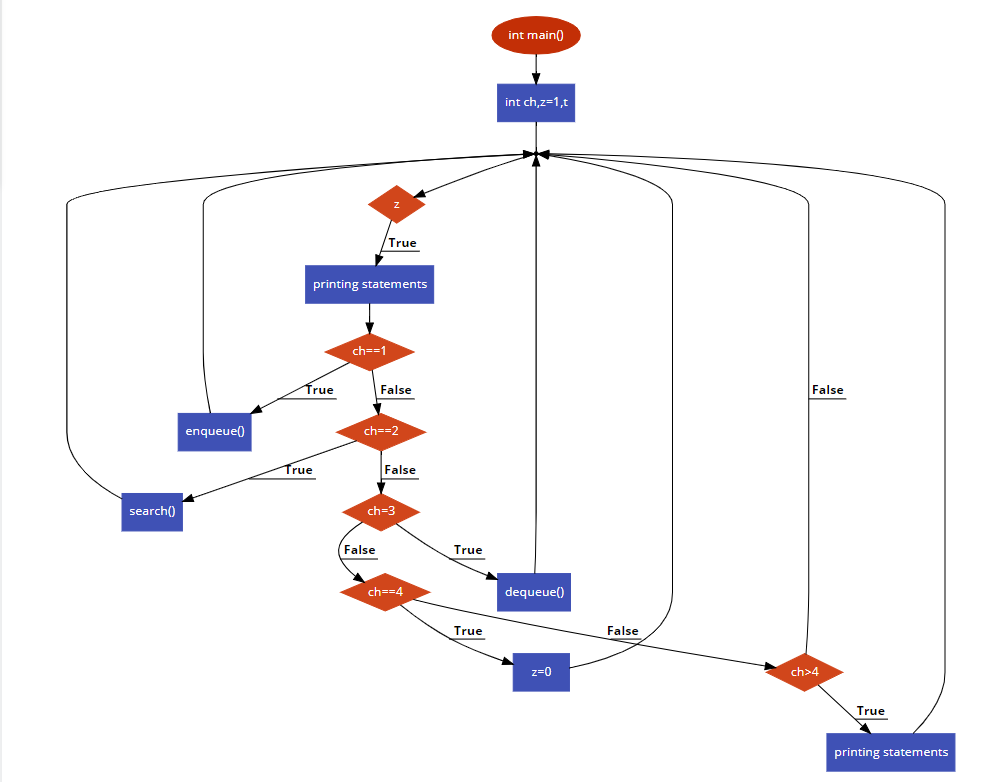
**6**

****

**7**

****

**8**

****

**9**

**IMPLEMENTATION**

#include<stdio.h>

#include<stdlib.h>

struct ram

{

char fn[100],ln[100],ad[100],dos[100],di[100];

int id;

}s[10];

int a[10],f=-1,r=-1;

int c=0;

void enqueue()

{

if(f==-1&&r==-1)

{

printf("\nEnter first name:");

scanf("%s",&s[0].fn);

printf("\nEnter last name:");

scanf("%s",&s[0].ln);

printf("\nEnter address:");

scanf("%s",&s[0].ad);

printf("\nEnter date of admission:");

scanf("%s",&s[0].dos);

printf("\nEnter disease:");

scanf("%s",&s[0].di);

printf("\nEnter register id:");

scanf("%d",&s[0].id);

a[0]=s[0].id;

f++;

r++;

}

else if(r==10&&f!=0)

{

r=0;

printf("\nEnter first name:");

scanf("%s",&s[0].fn);

printf("\nEnter last name:");

10

scanf("%s",&s[0].ln);

printf("\nEnter address:");

scanf("%s",&s[0].ad);

printf("\nEnter date of admission:");

scanf("%s",&s[0].dos);

printf("\nEnter disease:");

scanf("%s",&s[0].di);

printf("\nEnter register id:");

scanf("%d",&s[0].id);

a[0]=s[0].id;

}

else if((r==10&&f==0)||(r==f+1))

printf("\n it is not possible");

else

{

r++;

printf("\nEnter first name:");

scanf("%s",&s[r].fn);

printf("\nEnter last name:");

scanf("%s",&s[r].ln);

printf("\nEnter address:");

scanf("%s",&s[r].ad);

printf("\nEnter date of admission:");

scanf("%s",&s[r].dos);

printf("\nEnter disease:");

scanf("%s",&s[r].di);

printf("\nEnter register id:");

scanf("%d",&s[r].id);

a[r]=s[r].id;

}

if((r!=10&&f!=0)||(r!=f+1))

c++;

}

void dequeue()

{

if(c==0)

printf("\nthere are no patients");

else if(r==0)

11

r=10;

else

r--;

if(c!=0)

c--;

printf("successful");

}

void search()

{

int i=0,p;

if(c==0)

printf("no patients");

else if(c!=0&&f==0)

{

printf("\nEnter id number");

scanf("%d",&p);

for(i=0;i<=10;i++)

{

if(a[i]==p)

{

printf("\nEnter first name:");

printf("%s",s[0].fn);

printf("\nEnter last name:");

printf("%s",s[0].ln);

printf("\nEnter address:");

printf("%s",s[0].ad);

printf("\nEnter date of admission:");

printf("%s",s[0].dos);

printf("\nEnter disease:");

printf("%s",s[0].di);

printf("\nEnter register id:");

printf("%d",s[0].id);

}

}

}

else if(f>0)

{printf("\nEnter id number");

scanf("%d",&p);

12

int j=f;

for(i=0;i<=10;i++)

{

if(a[i]==p)

{

printf("\nEnter first name:");

printf("%s",s[r].fn);

printf("\nEnter last name:");

printf("%s",s[r].ln);

printf("\nEnter address:");

printf("%s",s[r].ad);

printf("\nEnter date of admission:");

printf("%s",s[r].dos);

printf("\nEnter disease:");

printf("%s",s[r].di);

printf("\nEnter register id:");

printf("%d",s[r].id);

}

if(j==10)

j=0;

j++;

}

}

else

printf("no match found");

}

int main()

{

int ch,z=1,t;

while(z)

{

printf("\n.................\n1.creating account\n2.finding account\n3.deleting account\n4.exit\n.........................");

printf("\nEnter your choice");

scanf("%d",&ch);

if(ch==1)

enqueue();

else if(ch==2)

search();

else if(ch=3)

13

dequeue();

else if(ch==4)

z=0;

else if(ch>4)

printf("\n please enter correct choice");

}

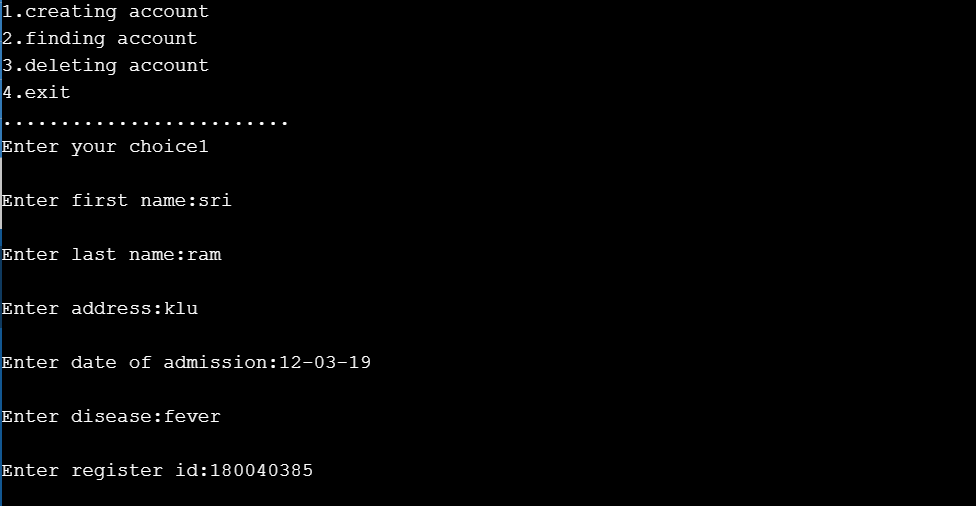
}

**14**

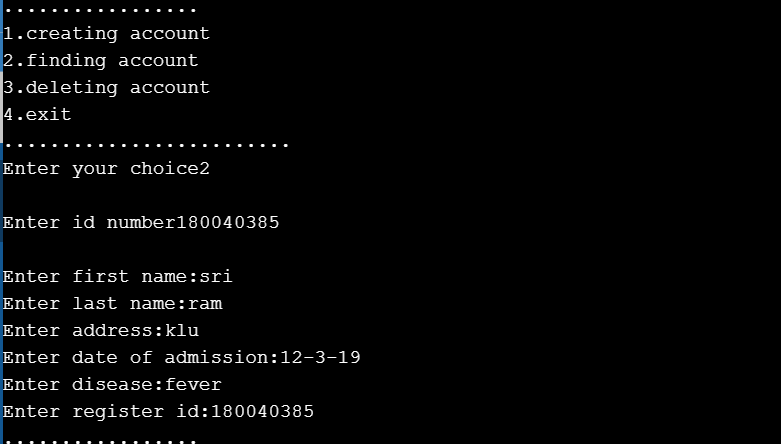
**INTEGRATION AND SYSTEM TESTING**

OUTPUTS

Screen Shots:

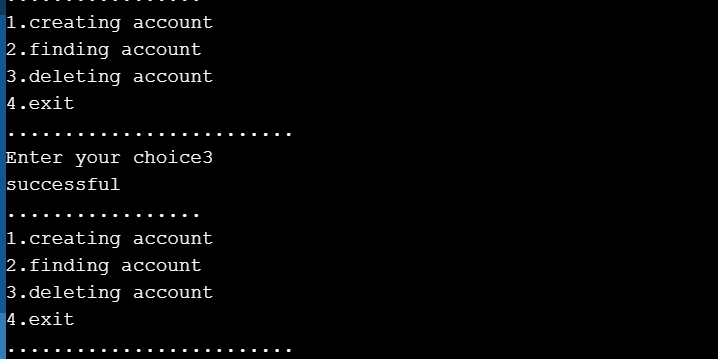


Creating the details of the patient (i.e, Name, Address, Date of submission,Disease,Register number)



Finding the details of the patient

15



Deleting the details of the patient

**16**

**CONCLUSION**

By using this circular queue we can coclude that the principle and the last position is connected back to the first position to make a circle.

An astounding and ever expanding amount of information is pouring in every day and they are used for organizing the economy and the society at large. This has paved the way for the collection of such information with the aid of sophisticated technologies and tools such as computers, satellites, remote sensors, and many more. At the outset, much more importance was given to the collection and storage of information as it is an established fact that information leads to power and power in turn leads to success