**G. H. RAISONI COLLEGE OF ENGG., NAGPUR**

**(An Autonomous Institute)**

**Department of Computer Science & Engg.**



**Date: 31-08-2021**

**Practical Subject: Data Structures and Algorithms in C.**

**Session: 2021-22**

**Student Details:**

| **Roll Number** | 01 |
| --- | --- |
| **Name** | Anand Suralkar |
| **Semester** | 9th |
| **Section** | A |
| **Batch** | CSE |

**Practical Details: Practical Number-4;**

| Practical Aim | Write a program in C/C++ to implement the following operations on QUEUE of Integers a) INSERT b) DELETE c)DISPLAY |
| --- | --- |
| Theory & Syntax | A queue in C is basically a linear data structure to store and manipulate the data elements. It follows the order of First In First Out (FIFO).  In queues, the first element entered into the array is the first element to be removed from the array.  For example, let’s consider the scenario of a bus-ticket booking stall. Here, the fashion of a C programming queue is followed. The tickets are distributed on the first-come-first-serve basis i.e. the first one to enter is the first one to be served with the tickets.  A queue is open at both ends. One end is provided for the insertion of data and the other end for the deletion of data.  A queue can be implemented with any programming language such as C, Java, Python, etc.  Operations Associated with a Queue in C  A queue being an Abstract Data Structure provides the following operations for manipulation on the data elements:  isEmpty(): To check if the queue is empty  isFull(): To check whether the queue is full or not  dequeue(): Removes the element from the frontal side of the queue  enqueue(): It inserts elements to the end of the queue  Front: Pointer element responsible for fetching the first element from the queue  Rear: Pointer element responsible for fetching the last element from the queue  Working of Queue Data Structure  Queue follows the First-In-First-Out pattern. The first element is the first to be pulled out from the list of elements.  Front and Rear pointers keep the record of the first and last element in the queue.  At first, we need to initialize the queue by setting Front = -1 and Rear = -1  In order to insert the element (enqueue), we need to check whether the queue is already full i.e. check the condition for Overflow. If the queue is not full, we’ll have to increment the value of the Rear index by 1 and place the element at the position of the Rear pointer variable. When we get to insert the first element in the queue, we need to set the value of Front to 0.  In order to remove the element (dequeue) from the queue, we need to check whether the queue is already empty i.e. check the condition for Underflow. If the queue is not empty, we’ll have to remove and return the element at the position of the Front pointer, and then increment the Front index value by 1. When we get to remove the last element from the queue, we will have to set the values of the Front and Rear index to -1. |
| Program | #include<iostream>  #include<conio.h>  using namespace std;  class queue{  public:  int size=10;  int data[10];  int front,rear;  queue(){  front=-1;rear=-1;}  void insert();  void deletion();  void display();  };  void queue::insert(){  if(rear==size-1){  cout<<"\n queue is full";  return;  }else if(rear==-1){  rear++;  front++;  }else{  rear++;  }  cout<<"Enter Data : ";  cin>>data[rear];  }  void queue::deletion(){  if(front==-1){  cout<<"\n Queue is empty";  return;  }  cout<<data[front]<<"deleted"<<endl;  if(front==rear){  front=-1;rear=-1;  }else{front++;  }  }  void queue::display(){  cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_{";  for(int i=front;i<=rear;i++){  cout<<data[i]<<",";  }  cout<<"}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  }  int main(){  queue q;  int ch;  do{cout<<"\n1.Insert\n2.Delete\n3.Display\n4.Quit\nEnter Choice ";  cin>>ch;  switch(ch){  case 1:q.insert();break;  case 2:q.deletion();break;  case 3:q.display();  }  }while(ch!=4);;  return 0;  } |
| Output |  |
| Conclusion | In C++ implemented the following operations on QUEUE of Integers a) INSERT b) DELETE c)DISPLAY |