21286 ①

Date :	Page No :
/ /20	

Assignment- 12

Shrikrushna S Tirape
21286

Problem Statement:-

Write a program to impliment a priority queue in c++ using an inorder list to store the item in queue. Create a class that includes a data item which should be template and the priority which should be an integer the inorder list should contain these items with operator <= overloaded so that the item with highest priority appear at the start of the list.

learning objectives:-

① To write & execute simple program in c++
② To learn the data structure priority queue.
③ To learn class template & operator overloading.

learning outcomes:-

① After execution of this assignment we will able to
- use class template & operator overloading
- priority queue along with it's operation.
- c++ program with various editors

21286 ②

: Date : | Page No :
/ /20

* Software requirement :-
  Eclipse IDE
  S.I.S Hingwo 64-bit.
  Windows 10OS

* H/W requirement.
  64 bit OS x64 based processor.
  intel Core is processor.

Theory :-

Priority queue :-
  It is a Collection of elements where elements are sorted according to their priority.
- Rules for priority queue :-
  ① The element with a higher priority is processed befor any element of lower priority.
  ② If the elements are with same priority then elements are added first in the queue could get processed.

Pseudocode :-
  Creating classes.
  template <class T>
  class node {
  Public :
  T data;
  int priority;
  node *next;
  node *privious;
  node()

Date :	Page No :
/ /20	

bool operator<=(&obj);
}

Algorithm Node(){
priority = 0
next = Null;
previous = Null;
}

Algorithm operator<=(&obj){
return. (this -> priority <= obj.priority);

Algorithm Queue(){
Front = Null rear = Null;
}

Algorithm enqueue(d,prior){
1 newnode -> data := d;
2 newnode -> priority := prior;
3 if (rear = Null) then rear := newnode,
front := newnode;
4 else.
4.1 if ((*newnode) <= (*rear)) then
4.1.1. newnode -> prev := rear.;
4.1.2 newnode -> next := Null;
4.1.3. rear -> next := newNode;
4.1.4 rear := newnode;
4.2 else if ((*newnode) <= (*rear)) then
4.2.1 := rear q := Null;
4.2.2. while ((P != NULL) && (*P) <= *newnode))
do.
4.2.2.1. q := p
4.2.2.2 p := p -> prev;
4.2.3 newnode -> prev := p;
4.2.4 newnode -> next := q;

21086

: Date :	Page No :
/ /20	

4.2.5 q → prev != newNode;

4.3 temp != rear;

4.4. while (temp → prev != Null) do temp != temp→pre

4.5 : front != temp;

Test Cases :-

Test Case	Test i/p	exep o/p	Act. o/p	Status
①	c=1 ch=1 d=10 p=2 choice=1 d=20 p=1	front 10 20 rear	front 10 20 rear	pass.
②	ch=1 choice=3	front 20 Rear.	front 20 rear	pass.

Complexity :-

O(n)

Conclusion :-

Successfy implimenta priority queue using linked list in c++