Assignment no	12
Aim	**Write program to implement a priority queue in C++ using an order list/array to store the items in the queue. Create a class that includes the data items (which should be template) and the priority (which should be int). The order list/array should contain these objects, with operator <= overloaded so that the items with highest priority appear at the beginning of the list/array (which will make it relatively easy to retrieve the highest item.)**
Objective	• To understand concept of Priority queue. • To Understand how this can be used to implement specific application • To understand the concept of operator overloading.
Outcome	• To implement operations on priority queue. • To write functions to use queue for job scheduling and prioritizing jobs. •
OS/Programming tools used	(64-Bit) 64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards Programming Tools (64-Bit) Eclipse with C++ plugin

Theory related to assignment:

Priority Queue

- It is an abstract data type that is similar to a queue, and every element has some priority value associated with it.
- The priority of the elements in a priority queue determines the order in which elements are served (i.e., the order in which they are removed).
- If in any case the elements have same priority, they are served as per their ordering in the queue.

Therefore, all the elements are either arranged in an ascending or descending order.

Properties of Priority Queue

A priority Queue is an extension of the queue with the following properties.

- Every item has a priority associated with it.
- An element with high priority is dequeued before an element with low priority.
- If two elements have the same priority, they are served according to their order in the queue.

Operations of a Priority Queue:

A typical priority queue supports the following operations:

1) Insertion in a Priority Queue

When a new element is inserted in a priority queue, it moves to the empty slot. However, if the element is not in the correct place, then it will be compared with other elements already in queue and are swapped . The swapping process continues until all the elements are placed in the correct position.

2) Deletion in a Priority Queue

It will remove the element which has maximum priority first. Thus, you remove 1st element from the queue as already elements are arranged according to priority so can be removed using FIFO principle

How to Implement Priority Queue?

Priority queue can be implemented using the following data structures:

Arrays

Linked list

Heap data structure

Implement Priority Queue Using Array:

A simple implementation is to use an array of the following structure/class

```
Class item {
    int id;
    int priority;
}
```

Operations on Priority Queue

enqueue(): This function is used to insert new data into the queue.

dequeue(): This function removes the element with the highest priority from the queue.

peek()/top(): This function is used to get the highest priority element in the queue without removing it from the queue.

Pseudocode

Enqueue Operation

Algorithm Enqueue(q[N], f, r, item(process no , priority))

{//q is an array of size N ,item is element to be inserted.

1. if(r==N-1)

1.1 Print “overflow"

2. if(f==-1)

2.1 f=0 , r=0

2.2 Enter process no and priority ,item

2.3 q[r]=item

else

2.1 r=r+1

2.2 Enter process no and priority, item // object of structure

2.3 a[r]=item

2.2 j=r-1

2.4 while((q[j] <= item) && (j >= F))

2.4.1q[j+1] = q[j]

2.4.2 j=j-1

2.5 q[j+1] = item;

}

Dequeue Operation

Algorithm Dequeue(q[N],item,f,r)

{

1. if ((f == – 1) || (f == r+1))

1.1 Print “QUEUE EMPTY”

else

1.1 f = f +1

}

In the above code we have have to use concept of operator overloading as mention in problem statement

int operator <= (item obj1,item ob2) {

if(ob1 . p < =ob2 . p)

return 1;

```
    else
  return 0;
}
```

Time Complexity

enqueue(): O(n)

dequeue():O(1)

peek()/top(): O(1)

operator <= overloaded

ADT :

Class Priority queue

{

Item A[] //a finite ordered list with zero or more elements(Process id and priority associated with them)

Public:

Queue();//create queue with some initial capacity.

Bool isempty();//to check queue is empty or not.

int peek();//return element at front.

void enqueue();//insert item at rear and arrange

void dequeue;//delete front element of the queue.

}

Conclusion:

The various operation of priority queue are implemented successfully using array data structure.

Review Questions:

1. What is priority queue?
2. Explain different operation of priority queue?
3. How priority queue is different from simple queue?
4. How can we use list to implement priority queue?
5. What is the complexity of various operations of queue?
6. What is an ADT? How can we define ADT of priority queue?
7. What are the various application of priority queue?
8. What is the advantage and disadvantage of implementing priority queue using an Array?