```
#include <iostream>

using namespace std;

class PriorityQue
{
    class Node
    {
        public:

        int pri,value;
        Node *next;

        Node(int p, int v)
        {
                  next=nullptr;
                  value=v;
                  pri= p;

                  }

     };
       public:

        Node *head;


        PriorityQue()
        {
            head =nullptr;
        }

        void push(int pri,int value)
        {
            Node *newNode = new Node(pri,value);

            if (nullptr == head)
            {
                head = newNode;
                return;
            }
            Node *cur;

            Node *back=nullptr;
            for (cur=head; cur!=nullptr; cur=cur->next)
            {
                if(pri<cur->pri)
                break;

                back=cur;
            }
```

```
            //if priority of new node is at start
            if(nullptr == back)
            {
                newNode->next = head;
                head = newNode;
                return;
            }

            //if priority of newNode is at newNode

            if (nullptr == cur)
            {
                back->next = newNode;
                return;
            }

            // if priority is in between nodes
            back->next = newNode;
            newNode->next = cur;

            return;
        }


        void pop(Node *h)
        {
            if (h == nullptr)
            throw "Que is empty";


            Node *temp = head;
            head = head->next;
            cout << temp -> value<< endl;
            delete temp;
        }

 };



int main()
{
    PriorityQue p1;
    p1.push(3,10);
    p1.push(1,4);
    p1.push (2,30);
      p1.push(1,100);

      p1.pop(p1.head);
    p1.pop(p1.head);
    p1.pop(p1.head);
    p1.pop(p1.head);
```

```
    return 0;
}
```