```cpp
#include <iostream>
using namespace std;

class PriorityQueue
{
public:
	struct Node
	{
	public:
		int priority;
		int value;
		Node *next;

		Node(int priority,int value):priority(priority),value(value),next(nullptr)
		{}

	};
	Node* head;
	int count;

	PriorityQueue():head(nullptr),count(0)
	{}

	void EnQueue(int pri,int val)
	{
		count++;
		Node * newnode=new Node(pri,val);
		Node *back=nullptr;
		Node *current=head;
		for(;current!=nullptr; current=current->next)
		{
			if(pri < current->priority)
				break;
			back=current;
		}

		//if head is null
		//newly created queue
		//

		if(head == nullptr)
		{
			head=newnode;

			return;
		}

		//when last priority is given
		//and u have to add last
		if(current==nullptr)
		{
			back->next=newnode;

			return;
		}

		if(back == nullptr)
		{
			newnode->next = head;
			head = newnode;
			return;
		}


		newnode->next=current;
```

```
                back->next=newnode;


        }

        void DeQueue()
        {
                Node *node=head;
                head=head->next;
                cout<<node->priority<<" "<<node->value<<endl;
                delete node;
        }

};

int main()
{
        PriorityQueue b1;

        int num,pri;
        while(cout<<"Enter priority & value to store in  (press 0to STOP)  ",
                cin>>pri>>num,
                pri)
        {
                b1.EnQueue(pri,num);
        }

        int count=b1.count;
        for(int i=0; i<count ; i++)
        {
                b1.DeQueue();
        }
}
```

OUTPUT: