

**/\*WAP to implement QUEUE using linked list \*/**

**#include<iostream>**

**#include<cstdlib>**

**using namespace std;**

**struct node**

**{**

**int data;**

**node\* next;**

**};**

**class prqueue**

**{**

**node \*head;**

**public:**

**prqueue()**

**{**

**head = NULL;**

**}**

**void enqueue(int n)**

**{**

**node \*ptr = head;**

**node \*newNode = new node;**

**newNode->data = n;**

**if(head == NULL)**

**{**

**head= newNode;**

**newNode->next = NULL;**

**}**

**else**

**{**

**while(ptr->next != NULL)**

**{**

**ptr = ptr->next;**

**}**

**newNode->next = ptr->next;**

**ptr->next = newNode;**

**}**

**}**

```

void dequeue()
{
    if(head == NULL)
    {
        cout<<endl<<endl<<"Underflow!!"<<endl;
    }
    else
    {
        node *ptr = head;
        cout<<endl<<"The dequeued data is: "<<head->data<<endl;
        head = head->next;
        delete ptr;
    }
}

void display_prqueue()
{
    if(head == NULL)
    {
        cout<<"\nThe list is empty!!"<<endl;
    }
    else
    {
        cout<<endl<<endl;
        node *ptr = head;
        while(ptr != NULL)
        {
            cout<<" "<<ptr->data<<" ";
            ptr = ptr->next;
        }
        cout<<endl<<endl;
    }
}

};

int main()
{
    prqueue queueobj;
    int choose;

```

```

do
{
    fflush(stdin);
    cout<<"1. Enqueue."<<endl;
    cout<<"2. Dequeue"<<endl;
    cout<<"3. Exit"<<endl;
    cout<<"\n\n\tChoose an option: ";
    cin>>choose;
    switch (choose)
    {
    case 1:
    {
        int val;
        cout<<"\nEnter push val: ";
        cin>>val;
        queueobj.enqueue(val);
        break;
    }
    case 2:
    {
        queueobj.dequeue();
        break;
    }
    case 3:
    {
        exit(1);
        break;
    }
    default :
    {
        cout<<"Invalid input";
        break;
    }
    }
    queueobj.display_prqueue();
}
while (choose != 3);

```

```
    return 0;
}
```

```
/*WAP to implement QUEUE using linked list */
#include<iostream>
#include<cstdlib>
using namespace std;
struct node
{
    int info;
    node *next;
};
class Queue
{
    node *Front,*rear;
    bool IsEmpty()
    {
        if(Front==NULL)
            return true;
        else
            return false;
    }
public:
    Queue()
    {
        Front=rear=NULL;
    }
    void enqueue(int num)
    {
        node *temp=new node;
        if(temp==NULL)
            cout<<"\n\nFailed to initialize the new node.\n\n";
        else
        {
            temp->info=num;

```

```

        temp->next=NULL;
        if(Front==NULL)
            Front=temp;
        else
            rear->next=temp;
        rear=temp;
    }
}
void dequeue()
{
    if(IsEmpty())
        cout<<"\n\nQueue Underflow\n\n";
    else
    {
        node *temp;
        temp=Front;
        cout<<"\n\nThe dequeued element is : "<<temp->info<<endl<<endl;
        if(Front==rear)
            Front=rear=NULL;
        else
            Front=Front->next;
        delete temp;
    }
}
void viewfront()
{
    if(IsEmpty())
        cout<<"\n\nQueue Underflow\n\n";
    else
        cout<<"\n\nThe front element of queue is : "<<Front->info<<"\n\n";
}
void displayQueue()
{
    if(IsEmpty())
        cout<<"\n\nQueue Underflow\n\n";
    else
    {

```

```

        node *temp;
        temp=Front;
        cout<<"\n\nElements of queue are : "<<endl;
        while(temp!=NULL)
        {
            cout<<temp->info<<"\t";
            temp=temp->next;
        }
        cout<<"\n\n";
    }
};

int main()
{
    int choice,num;
    Queue q;
    while(1)
    {
        cout<<"1. Enqueue\n2. Dequeue\n3. View front element\n4. View
queue\n5. Exit\n\nEnter your choice : ";
        cin>>choice;
        switch(choice)
        {
            case 1:
            {
                while(1)
                {
                    cout<<"\nEnter -1 to finish enqueue\nEnter the value: ";
                    cin>>num;
                    if(num== -1)
                        break;
                    q.enqueue(num);
                }
                break;
            }
            case 2:
            {

```

```

        q.dequeue();
        break;
    }
    case 3:
    {
        q.viewfront();
        break;
    }
    case 4:
    {
        q.displayQueue();
        break;
    }
    default :
        exit(0);
    }
}
return 0;
}

```

```

/*WAP to implement QUEUE using linked list */
#include<iostream>
using namespace std;
class Queue
{
    struct node
    {
        int data;
        struct node * next;
    };
public:
    struct node * start;
    struct node * newnode,* temp,* ptr,*Front,*Rear;
    void creation()
    {

```

```

Front=Rear=NULL;
newnode = new node;
cout<<"Enter the data for the queue(insert -1 to end the ): ";
cin>>newnode->data;
newnode->next=NULL;
if (start==NULL)
{
    start=newnode;
    temp=newnode;
}
else
{
    temp->next=newnode;
    temp=newnode;
}
do
{
    enqueue();
    Front= start;
}
while (newnode->data!=-1);
}
void enqueue()
{
    newnode=new node;
    cout<<"Enter the data to be stored in the queue: ";
    cin>>newnode->data;
    if (newnode->data!=-1)
    {
        ptr=start;
        while(ptr->next!=NULL)
        {
            ptr=ptr->next;
        }
        ptr->next=newnode;
        newnode->next=NULL;
        Rear=newnode;
    }
}

```



```

    }
}
void dequeue()
{
    ptr=start->next;
    delete start;
    start=ptr;
    Front=start;
}
void display_queue()
{
    ptr=start;
    cout<<"-----"<<endl;
    cout<<"\n\nThe queue is: "<<endl;
    cout<<"\t\t"<<ptr->data;
    while(ptr->next!=NULL)
    {
        ptr=ptr->next;
        cout<<"\t"<<ptr->data;
    }
    cout<<endl;
    cout<<"Front: "<<Front->data<<endl;
    cout<<"Rear: "<<Rear->data<<endl;
    cout<<"-----"<<endl;
}
};
int main()
{
    class Queue q;
    q.start=NULL;
    int choice=0,c=0;
    while(choice!=10)
    {
        c++;
        cout<<"\n\nyour Choice please: "<<endl;
        if (c==1)
        {

```

```

        cout<<"0-Creating a new queue "<<endl;
    }
    cout<<"1-Enqueue "<<endl;
    cout<<"2-Dequeue "<<endl;
    cout<<"10-Exit.\n"<<endl;
    cout<<"\t\t your choice: ";
    cin>>choice;
    switch (choice)
    {
    case 0:
        q.creation();
        break;
    case 1:
        q.enqueue();
        break;
    case 2:
        q.dequeue();
        break;
    }
    q.display_queue();
}
cout<<"THANK YOU";
}

```

```

/*WAP to implement QUEUE using linked list */
#include<iostream>
using namespace std;
class linkList
{
    struct Node
    {
        int data;
        Node *next;
    };
    typedef struct Node* nodeptr;

```

```

    nodeptr head;
public:
    linkList()    //constructor
    {
        head=NULL;
    }
    void enqueue(int new_data)    //insert at the rear
    {
        nodeptr p;
        nodeptr ptr=head;
        if(head==NULL)
        {
            nodeptr p;
            p=new Node;
            p->data= new_data;
            p->next=head;
            head=p;
        }
        else
        {
            while(ptr->next!=NULL)
            {
                ptr=ptr->next;
            }
            p=new Node;
            ptr->next=p;
            p->data =new_data;
            p->next =NULL;
        }
    }
    int dequeue()    // delete from the front
    {
        nodeptr ptr=head;
        if(head!=NULL)
        {
            head=ptr->next;
            cout<<ptr->data<<" is Dequeued\n\n"<<endl;

```

```

        delete ptr;
        return ptr->data;
    }
    else
    {
        cout<<"Empty\n\n"<<endl;
        return -1;
    }
}

void display()    // display the list
{
    nodeptr p=head;
    cout<<"\n\t=====X======"<<endl;
    cout<<"\taddress"<<"\t\tdata"<<"\t\tnext"<<endl;
    while(p!=NULL)
    {
        cout<<"\t"<<p<<"\t"<<p->data<<"\t"<<p->next<<endl;
        p=p->next;
    }
    if(head==NULL)
    {
        cout<<"\tEmpty"<<endl;
    }
    cout<<"\tthat's it"<<endl;
    cout<<"\t=====X=====\\n"<<endl;
}

};

int main()
{
    linkList li;
    int x,a;
    int choice=-1;
    while(choice!=0)
    {
        cout<<"\n\nyour Choice please: "<<endl;
        cout<<"1-Enqueue "<<endl;
        cout<<"2-Dequeue "<<endl;
    }
}

```

```

cout<<"0-Exit\n"<<endl;
cout<<"\t\tyour choice: ";
cin>>choice;
system("CLS");
cout<<"\tBEFORE LIST";
li.display();
switch (choice)
{
case 1:
    cout<<"enter data to insert: ";
    cin>>x;
    li.enqueue(x);
    break;
case 2:
    li.dequeue();
    break;
}
cout<<"\tAFTER LIST";
li.display();
}
cout<<"\n=====X======"<<endl;
cout<<"\t THANK YOU "<<endl;
return 0;
}

```