**Queue using Array Implementation**

#include<iostream>

using namespace std;

const int size=10;

template <class t>

class queue

{

t q[size];

int f;

int r;

public:

queue()

{

f=r=-1;

}

void enqueue(t );

t dequeue();

t frontmost();

t rearmost();

int count();

int isEmpty();

int isFull();

void clear();

void display();

};

template <class t>

void queue<t>::enqueue(t i)

{

if(f==-1)

{

f=r=0;

q[r]=i;

}

else

{

r++;

q[r]=i;

}

}

template <class t>

t queue<t>::dequeue()

{

t del;

if(f==r)

{

del=q[f];

f=r=-1;

}

else

{

del=q[f];

f++;

}

return del;

}

template <class t>

t queue<t>::frontmost()

{

t temp;

temp=q[f];

return temp;

}

template <class t>

t queue<t>::rearmost()

{

t temp;

temp=q[r];

return temp;

}

template <class t>

int queue<t>::count()

{

int c=0;

if(f==-1)

return(0);

else

{

for(int i=f;i<=r;i++)

c++;

}

return(c);

}

template <class t>

int queue<t>::isEmpty()

{

if((f==-1)&&(r==-1))

return(1);

else

return(0);

}

template <class t>

int queue<t>::isFull()

{

if(r==size-1)

return(1);

else

return(0);

}

template <class t>

void queue<t>::clear()

{

f=r=-1;

}

template <class t>

void queue<t>::display()

{

if(r==-1)

cout<<"Queue is empty "<<endl;

else

{

cout<<"Contents of Queue are "<<endl;

for(int i=f;i<=r;i++)

cout<<q[i]<<" ";

cout<<endl;

}

}

int main()

{

queue<int> q1;

int choice,empty,full,i,d;

int f,r,c;

char ch='y';

do

{

cout<<"--Main Menu--"<<endl;

cout<<"1.To insert a new element "<<endl;

cout<<"2.To remove the element "<<endl;

cout<<"3.Display the frontmost element "<<endl;

cout<<"4.Display the rearmost element "<<endl;

cout<<"5.Count the no of elements in queue "<<endl;

cout<<"6.Check if queue is empty "<<endl;

cout<<"7.Check if queue is full "<<endl;

cout<<"8.Clear the queue "<<endl;

cout<<"9.Display the contents of the queue "<<endl;

cout<<"Enter your choice "<<endl;

cin>>choice;

switch(choice)

{

case 1: full=q1.isFull();

if(full==1)

cout<<"Insertion not possible.Queue is Full "<<endl;

else

{

cout<<"Enter the element you want to insert "<<endl;

cin>>i;

q1.enqueue(i);

q1.display();

}

break;

case 2: empty=q1.isEmpty();

if(empty==1)

cout<<"Deletion not posssible.Queue is Empty. "<<endl;

else

{

d=q1.dequeue();

cout<<d<<"has been deleted "<<endl;

q1.display();

}

break;

case 3: f=q1.frontmost();

cout<<"The frontmost element in queue is "<<f<<endl;

break;

case 4: r=q1.rearmost();

cout<<"The rearmost element in queue is "<<r<<endl;

break;

case 5: c=q1.count();

cout<<"The total no elements in queue are "<<c<<endl;

break;

case 6: empty=q1.isEmpty();

if(empty==1)

cout<<"Queue is Empty "<<endl;

else

cout<<"Queue is not Empty "<<endl;

break;

case 7: full=q1.isFull();

if(full==1)

cout<<"Queue is Full"<<endl;

else

cout<<"Queue is not Full "<<endl;

break;

case 8: q1.clear();

q1.display();

break;

case 9 : q1.display();

break;

default :cout<<"Error in input ";

}

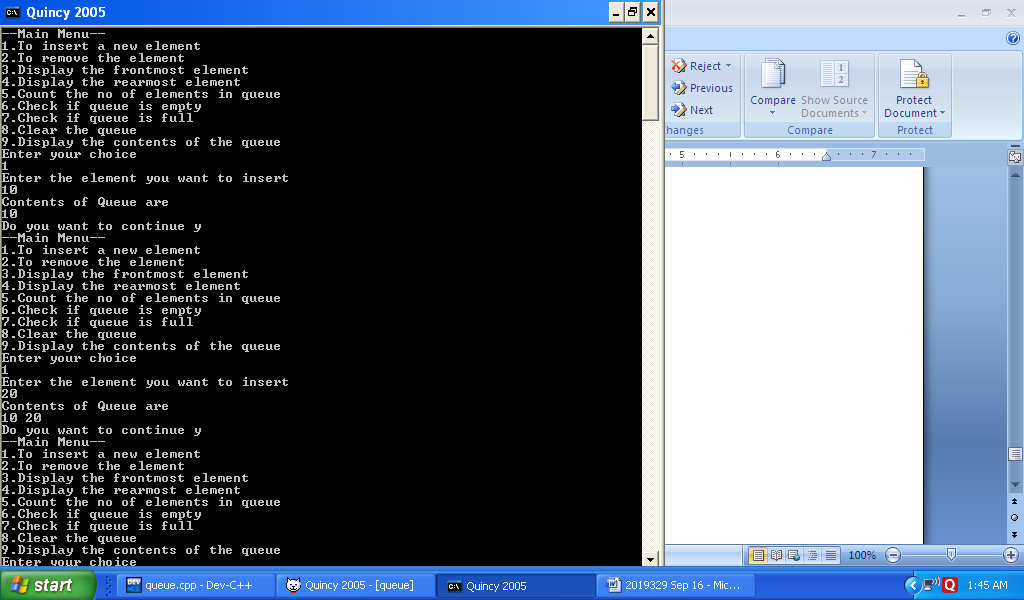
cout<<"Do you want to continue ";

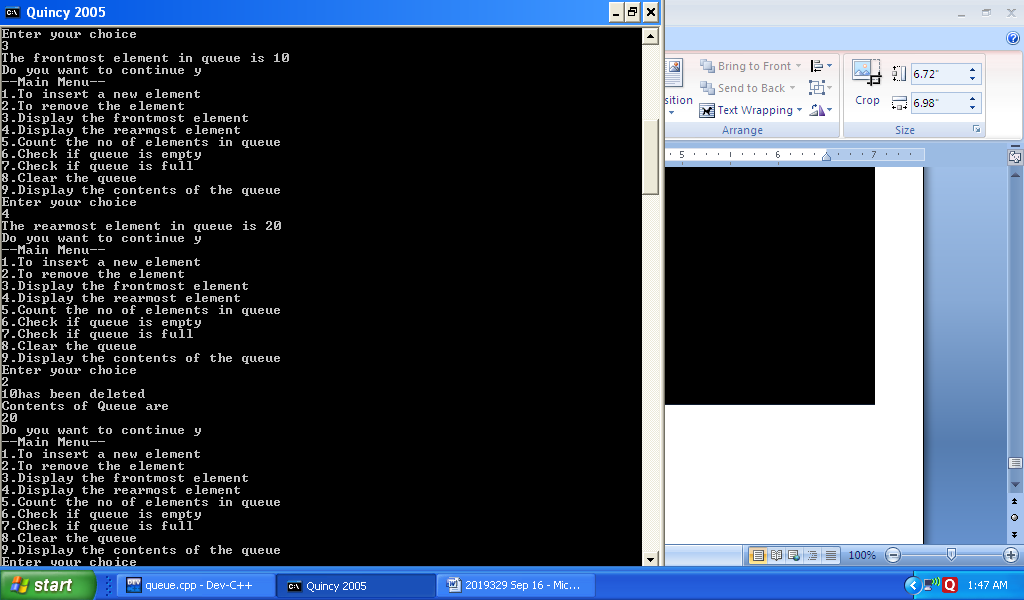
cin>>ch;

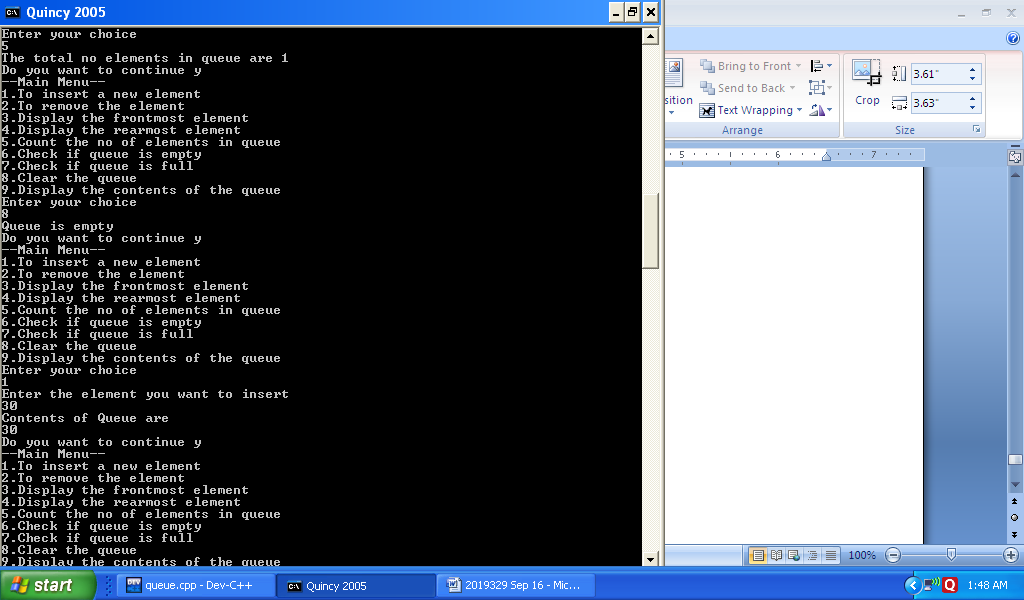
}while(ch=='y');

return 0;

}







**Queue using Linked Implementation**

#include<iostream>

using namespace std;

template<class t>

class node

{

public:

t data;

node \*next;

node(t x,node\* n=NULL)

{

data=x;

next=n;

}

};

template<class t>

class queue

{

node<t> \*f;

node<t> \*r;

public:

queue()

{

f=r=NULL;

}

void enqueue(node<t>\* );

t dequeue();

t frontmost();

t rearmost();

int count();

int isEmpty();

void clear();

void display();

node<t>\* createNode(t);

};

template <class t>

void queue<t>::enqueue(node<t> \*newnode)

{

if(f==NULL)

f=r=newnode;

else

{

r->next=newnode;

r=newnode;

}

}

template <class t>

t queue<t>::dequeue()

{

t d=f->data;

node<t>\*temp=f;

if(f==r)

{

f=r=NULL;

}

else

{

f=f->next;

}

delete temp;

return(d);

}

template <class t>

t queue<t>::frontmost()

{

t temp;

temp=f->data;

return(temp);

}

template <class t>

t queue<t>::rearmost()

{

t temp;

temp=r->data;

return(temp);

}

template <class t>

int queue<t>::count()

{

node<t> \*temp;

int c=0;

for(temp=f;temp!=NULL;temp=temp->next)

c++;

return(c);

}

template <class t>

int queue<t>::isEmpty()

{

if(f==NULL)

return(1);

else

return(0);

}

template <class t>

void queue<t>::clear()

{

node<t> \*temp;

for(temp=f;temp!=NULL;temp=f)

{

f=f->next;

delete temp;

}

}

template <class t>

void queue<t>::display()

{

node<t> \*temp;

if(f==NULL)

cout<<"Queue is Empty "<<endl;

else

{

cout<<"Contents of Queue are "<<endl;

for(temp=f;temp!=NULL;temp=temp->next)

cout<<temp->data<<" ";

cout<<endl;

}

}

template <class t>

node<t>\* queue<t>::createNode(t i)

{

node<t> \*newnode=new node<t>(i);

newnode->data=i;

newnode->next=NULL;

return newnode;

}

int main()

{

queue<int> q1;

node<int> \*newnode;

int choice,i,empty,d;

int f,r,c;

char ch='y';

do

{

cout<<"--Main Menu--"<<endl;

cout<<"1.To insert a new element "<<endl;

cout<<"2.To remove the element "<<endl;

cout<<"3.Display the frontmost element "<<endl;

cout<<"4.Display the rearmost element "<<endl;

cout<<"5.Count the no of elements in queue "<<endl;

cout<<"6.Check if queue is empty "<<endl;

cout<<"7.Clear the queue "<<endl;

cout<<"8.Display the contents of the queue "<<endl;

cout<<"Enter your choice "<<endl;

cin>>choice;

switch(choice)

{

case 1: cout<<"Enter the data you want to insert "<<endl;

cin>>i;

newnode=q1.createNode(i);

q1.enqueue(newnode);

q1.display();

break;

case 2: empty=q1.isEmpty();

if(empty==1)

cout<<"Deletion not posssible.Queue is Empty. "<<endl;

else

{

d=q1.dequeue();

cout<<d<<" has been deleted "<<endl;

q1.display();

}

break;

case 3: f=q1.frontmost();

cout<<"The frontmost element in queue is "<<f<<endl;

break;

case 4: r=q1.rearmost();

cout<<"The rearmost element in queue is "<<r<<endl;

break;

case 5: c=q1.count();

cout<<"The total no elements in queue are "<<c<<endl;

break;

case 6: empty=q1.isEmpty();

if(empty==1)

cout<<"Queue is Empty "<<endl;

else

cout<<"Queue is not Empty "<<endl;

break;

case 7: q1.clear();

q1.display();

break;

case 8 : q1.display();

break;

default :cout<<"Error in input ";

}

cout<<"Do you want to continue ";

cin>>ch;

}while(ch=='y');

return 0;

}

