**Practical No: 04**

**Implement the following for Queue**

1. **Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.**

**Code:**

#include <iostream>

#include <cstdlib>

using namespace std;

struct queue {

int data;

queue \*next;

};

struct queue \*top = NULL;

void insertion(int value) {

struct queue \*n=(struct queue\*)malloc(sizeof(struct queue));

n->data = value;

n->next = NULL;

if (top == NULL) {

top = n;

return;

}

else{

struct queue \*temp = top;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = n;

}

}

void display() {

struct queue \*temp = top;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

void del(){

struct queue \*t=top;

top=top->next;

free(t);

}

int main(){

int ch,c;

do{

cout<<"Enter\n1.Push\n2.Pop\n3.show\n4.Exit"<<endl;

cin>>ch;

switch(ch){

case 1:

int size,nm;

cout<<"Enter size of queue: ";

cin>>size;

for(int i=0;i<size;i++){

cout<<"Enter element: ";

cin>>nm;

insertion(nm);

}

cout<<"pushed element are: ";

display();

cout<<endl;

break;

case 2:

cout<<"After pop: ";

del();

display();

cout<<endl;

break;

case 3:

cout<<"Your final queue: ";

display();

break;

case 4:

exit(0);

break;

}

cout<<"Do you want to continue?(0/1)";

cin>>c;

}

while(c!=0);

}

**Output:**

**……………………………………………………………………………………….**

Enter

1.Push

2.Pop

3.show

4.Exit

1

Enter size of queue: 5

Enter element: 10

Enter element: 20

Enter element: 30

Enter element: 40

Enter element: 50

pushed element are: 10 20 30 40 50

Do you want to continue?(0/1)1

Enter

1.Push

2.Pop

3.show

4.Exit

2

After pop: 20 30 40 50

Do you want to continue?(0/1)1

Enter

1.Push

2.Pop

3.show

4.Exit

3

Your final queue: 20 30 40 50

Do you want to continue?(0/1)1

Enter

1.Push

2.Pop

3.show

4.Exit

4

**……………………………………………………………………………………….**

1. **Write a program to implement the concept of Circular Queue.**

**Code:**

#include<iostream>

#include<cstdlib>

using namespace std;

struct node{

int data;

node \*next;

};

struct node \*head=NULL;

void inse(int v){

struct node \*n=(struct node \*)malloc(sizeof(struct node));

n->data=v;

n->next=NULL;

if(head==NULL){

head=n;

head->next=n;

}

else{

struct node \*t=head;

while(t->next!=head){

t=t->next;

}

t->next=n;

n->next=head;

}

}

void display(){

struct node \*s=head;

if(head!=NULL){

while(s->next!=head){

cout<<s->data<<"("<<&s->data<<")"<<"----->"<<s->next<<endl;

s=s->next;

}

cout<<s->data<<"("<<&s->data<<")"<<"----->"<<s->next<<endl;

}

else{

cout<<"List is empty";

}

}

int main(){

inse(10);

inse(20);

inse(30);

inse(40);

cout<<"Element in list"<<endl;

display();

}

**Output:**

**….………………………………………………………………………………….**

Element in list

10(0x181510)----->0x181530

20(0x181530)----->0x181550

30(0x181550)----->0x181a10

40(0x181a10)----->0x181510

**.……………………………………………………………………………………**

1. **Write a program to implement the concept of Deque.**
2. **Input restricted**

**Code:**

#include <iostream>

#include <cstdlib>

using namespace std;

struct queue {

int data;

queue \*next;

};

struct queue \*top = NULL;

void End(int value) {

struct queue \*n=(struct queue\*)malloc(sizeof(struct queue));

n->data = value;

n->next = NULL;

if (top == NULL) {

top = n;

return;

}

struct queue \*temp = top;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = n;

}

void display() {

struct queue \*temp = top;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

void delf(){

struct queue \*t=top;

top=top->next;

free(t);

}

void dele(){

struct queue \*t=top;

while(t->next->next!=NULL){

t=t->next;

}

free(t->next);

t->next=NULL;

}

int main({

End(10);

End(20);

End(30);

End(40);

End(50);

cout<<"Element in Queue"<<endl;

display();

cout<<endl;

int ch;

int c;

do{

cout<<"Enter\n1.Deletion from begining\n2.Deletion at end\n3.Exit"<<endl;

cin>>ch;

switch(ch){

case 1:

cout<<"Element after deleting first elemennt"<<endl;

delf();

display();

cout<<endl;

break;

case 2:

cout<<"Element after deleting last elemennt"<<endl;

dele();

display();

cout<<endl;

break;

case 3:

exit(0);

break;

}

cout<<"Do you want to continue?(0/1): ";

cin>>c;

}

while(c!=0);

}

**Output:**

**….……………………………………………………………………………**

Element in Queue

10 20 30 40 50

Enter

1.Deletion from begining

2.Deletion at end

3.Exit

1

Element after deleting first elemennt

20 30 40 50

Do you want to continue?(0/1): 1

Enter

1.Deletion from begining

2.Deletion at end

3.Exit

2

Element after deleting last elemennt

20 30 40

Do you want to continue?(0/1): 1

Enter

1.Deletion from begining

2.Deletion at end

3.Exit

3

**….……………………………………………………………………………**

1. **Output restricted**

**Code:**

#include<iostream>

#include<cstdlib>

using namespace std;

struct queue{

int data;

struct queue \*next;

};

struct queue \*top=NULL;

void insertionf(int v)

{

struct queue \*new\_n=(struct queue\*)malloc(sizeof(struct queue));

new\_n->data=v;

new\_n->next=NULL;

if(top==NULL){

top=new\_n;

}

else{

new\_n->next=top;

top=new\_n;

}

}

void insertione(int value) {

struct queue \*n=(struct queue\*)malloc(sizeof(struct queue));

n->data = value;

n->next = NULL;

if (top == NULL) {

top= n;

return;

}

struct queue \*temp = top;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = n;

}

void del(){

struct queue \*t=top;

top=top->next;

free(t);

}

void display(){

struct queue \*temp=top;

while(temp!=NULL){

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

temp=temp->next;

}

}

int main(){

int size,nm;

cout<<"Enter size of Queue: ";

cin>>size;

for(int i=0;i<size;i++){

cout<<"Enter element: ";

cin>>nm;

insertione(nm);

}

cout<<"Your Queue is: ";

display();

int ch;

int c;

do{

cout<<"\nEnter\n1.Insert at begining\n2.Insert at end\n3.Delete element\n4.exit"<<endl;

cin>>ch;

switch(ch){

case 1:

int num;

cout<<"Enter no: ";

cin>>num;

insertionf(num);

cout<<endl;

cout<<"Element in Queue: ";

display();

cout<<endl<<endl;

break;

case 2:

int nm;

cout<<"Enter no: ";

cin>>nm;

insertione(nm);

cout<<endl;

cout<<"Element in Queue: ";

display();

cout<<endl<<endl;

break;

case 3:

if(top==NULL){

cout<<"Queue is in underflow";

}

else{

del();

cout<<"Queue after delete: "<<endl;

display();

}

cout<<endl;

break;

case 4:

exit(0);

break;

}

cout<<"Do you want to continue?(0/1)";

cin>>c;

}

while(c!=0);

}

**Output:**

**…………………………………………………………………………………**

Enter size of Queue: 5

Enter element: 10

Enter element: 20

Enter element: 30

Enter element: 40

Enter element: 50

Your Queue is: 10 20 30 40 50

Enter

1.Insert at begining

2.Insert at end

3.Delete element

4.exit

1

Enter no: 5

Element in Queue: 5 10 20 30 40 50

Do you want to continue?(0/1)1

Enter

1.Insert at begining

2.Insert at end

3.Delete element

4.exit

2

Enter no: 60

Element in Queue: 5 10 20 30 40 50 60

Do you want to continue?(0/1)1

Enter

1.Insert at begining

2.Insert at end

3.Delete element

4.exit

3

Queue after delete:

10 20 30 40 50 60

Do you want to continue?(0/1)1

Enter

1.Insert at begining

2.Insert at end

3.Delete element

4.exit

4

**……………………………………………………………………………………….**