***Comsats University Islamabad***

***Vehari Campus***

**DS Lab Assignment no. 2**

Submitted by:

Anum Batool

Roll no.:

Sp22-BCS-112

Section:

B

Subject:

Data Structure

Submitted to:

Mam Yasmeen Jana

**Activity 1:**

#include <iostream>

using namespace std;

class Node {

public

int data;

Node\* next;

Node(){

data=data;

next=NULL;

}

void displayLinkedList(Node\* head) {

cout << "The linked list is: ";

Node\* ptr = head;

while (ptr != NULL) {

cout << ptr->data << " ";

ptr = ptr->next;

}

cout << endl << "\*\*head address: " << head << endl;

cout << "--------------------------" << endl;

cout << "head content: " << head<< endl;

cout << "--------------------------" << endl;

cout << "\*ptr address:\* @" << &head << endl;

cout << "--------------------------" << endl;

cout << "ptr content: " << head << endl;

cout << "----------------------" << endl;

ptr = head;

while (ptr != NULL) {

cout << "ptr->data: " << ptr->data << endl

cout << "----------------------" << endl;

cout << "ptr: " << ptr << endl;

cout << "ptr->next: " << ptr->next << endl;

ptr = ptr->next;

}

}

};

int main() {

Node n;

Node\* head = new Node();

Node\* second = new Node();

Node\* third = new Node();

Node\* fourth = new Node();

head->data = 1;

head->next = second;

second->data = 2;

second->next = third;

third->data = 20;

third->next = fourth;

fourth->data = 30;

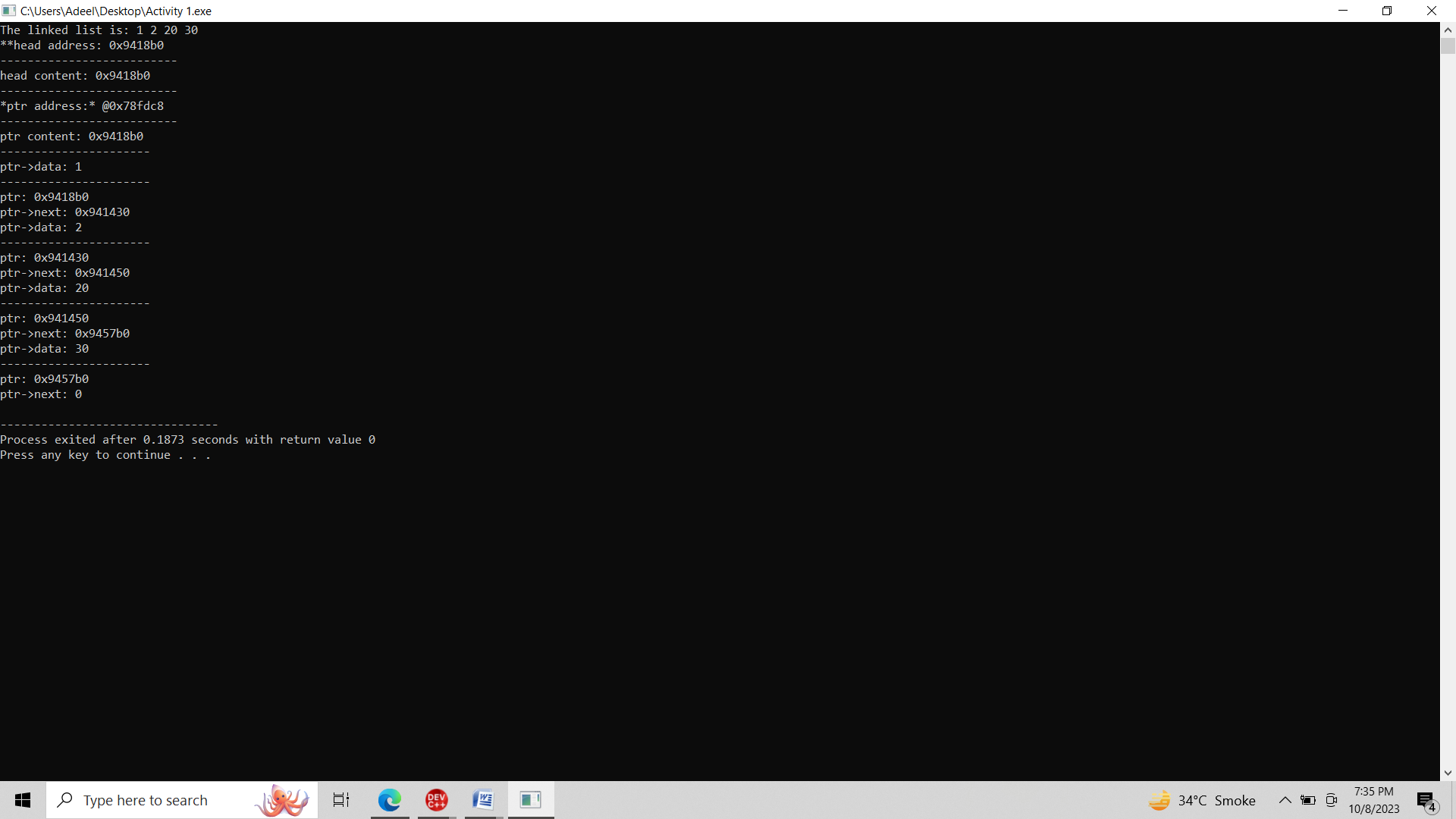
fourth->next = NULL;

n. displayLinkedList(head);

return 0;

}

**Output:**



**Activity 2:**

**Singly:**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

class Node{

private:

int data;

Node \*next;

public:

Node \*head;

Node ()

{

head=NULL;

}

void insert(int n)

{

if (head==NULL){

head= new Node();

head->data=n;

head-> next=NULL;

}

else

{

Node \*p,\*ptr;

ptr=head;

while(ptr->next!=NULL){

ptr=ptr->next;

}

p=new Node();

p->data=n;

p->next=NULL;

ptr->next=p;

}

}

void insert\_beg(int n1)

{

if (head==NULL){

head= new Node();

head->data=n1;

head-> next=NULL;

}

else

{

Node \*p;

p=new Node();

p->data=n1;

p->next=head;

head=p;

}

}

void insert\_at\_value( int pos, int n2){

if (head==NULL){

head= new Node();

head->data=n2;

head-> next=NULL;

}

else

{

Node \*ptr;

ptr=head;

while(ptr->data!=pos){

ptr=ptr->next;

}

Node \*p;

p=new Node();

p->data=n2;

p->next=ptr->next;

ptr->next=p;

} }

void del\_beg(int n1){

if(head==NULL){

cout<<"Empty";

}

else{

Node \*ptr;

ptr=head;

head=ptr->next;

delete ptr;

ptr=NULL;

}

}

void del\_end(int n1){

if(head==NULL){

head= new Node();

head->data=n1;

head-> next=NULL;

}

else{

Node \*ptr,\*p;

ptr=head;

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

ptr=ptr->next;

}

p=ptr->next;

ptr->next=NULL;

delete p;

p=NULL;}

}

void del\_at\_value(int pos){

if(head==NULL){

cout<<"Empty";

}

else{

Node \*ptr;

ptr=head;

while(ptr->next->data!=pos){

ptr=ptr->next;

}

Node \*p;

p=ptr->next;

ptr->next=ptr->next->next;

p->next=NULL;

delete p;

p=NULL;

}

}

void display()

{

Node \*ptr;

ptr=head;

while(ptr!=NULL){

cout<<ptr->data;

ptr=ptr->next;

}

}

};

int main(){

Node n;

int ch;

do{

cout<<endl;

cout<<"Press 1 for insert a nodes"<<endl;

cout<<"Press 2 for insert node at begining "<<endl;

cout<<"Press 3 for insert at specific value "<<endl;

cout<<"Press 4 for display"<<endl;

cout<<"Press 5 for deletion at begining"<<endl;

cout<<"Press 6 for deletion at end"<<endl;

cout<<"Press 7 for deletion at specific value"<<endl;

cout<<"Press 8 for exit"<<endl;

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

cin>>ch;

if(ch==1){

cout<<"insert a nodes"<<endl;

n.insert(1);

n.insert(2);

n.insert(3);

n.insert(4);

}

else if(ch==2){

cout<<"insert a nodes at begining"<<endl;

n.insert\_beg(5);

}

else if(ch==3){

cout<<"insert at specific value "<<endl;

n.insert\_at\_value(3,7);

}

else if(ch==4){

cout<<"For display "<<endl;

n.display();

}

else if(ch==5){

cout<<"Press 5 for deletion at begining"<<endl;

n.del\_beg(5);

}

else if(ch==6){

cout<<"Press 6 for deletion at end"<<endl;

n.del\_end(4);

}

else if(ch==7){

cout<<"Press 7 for deletion at specific value"<<endl;

n.del\_at\_value(3);

}

else{

cout<<"invalid choice";

}

}

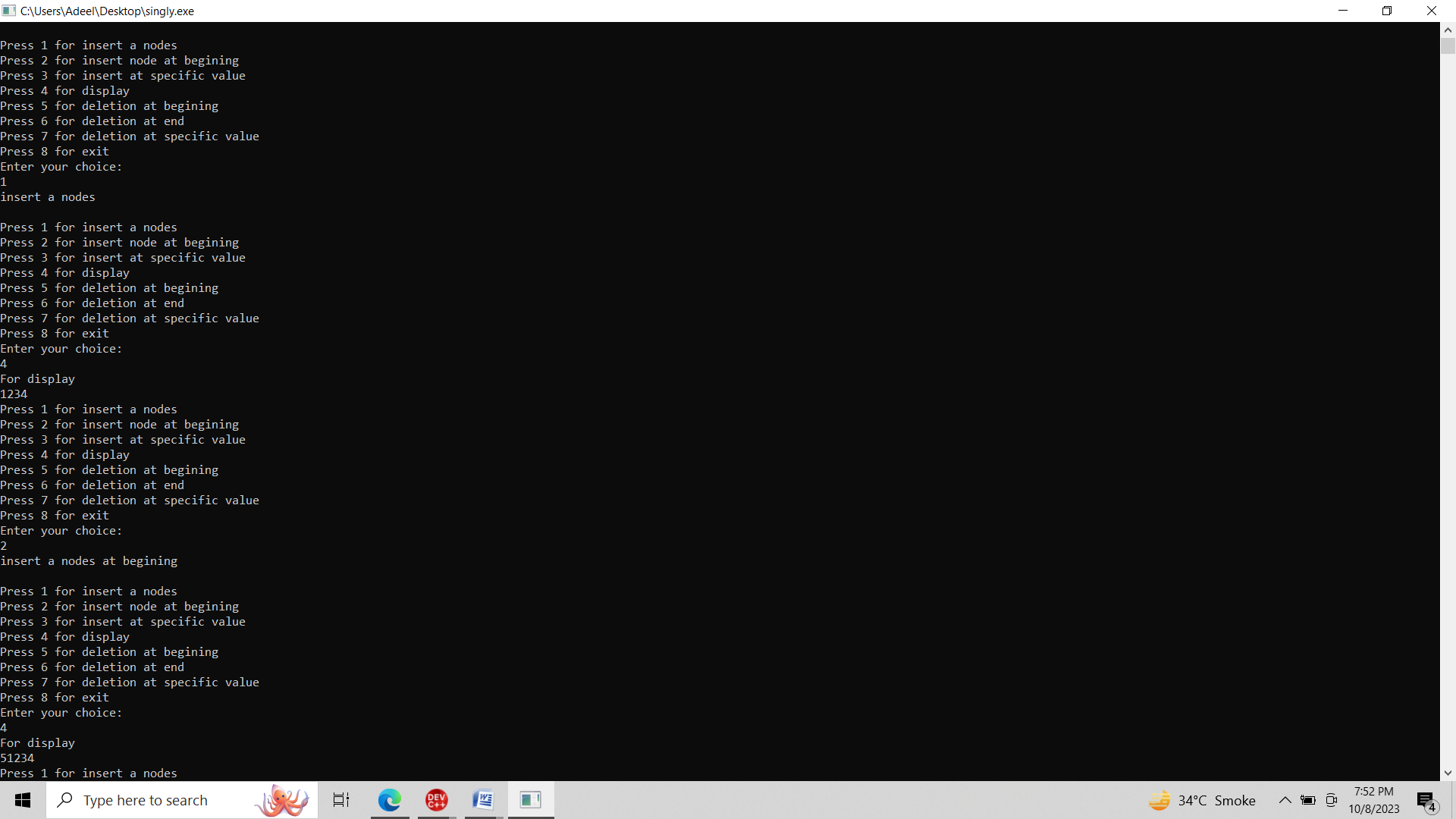
while(ch!=8);

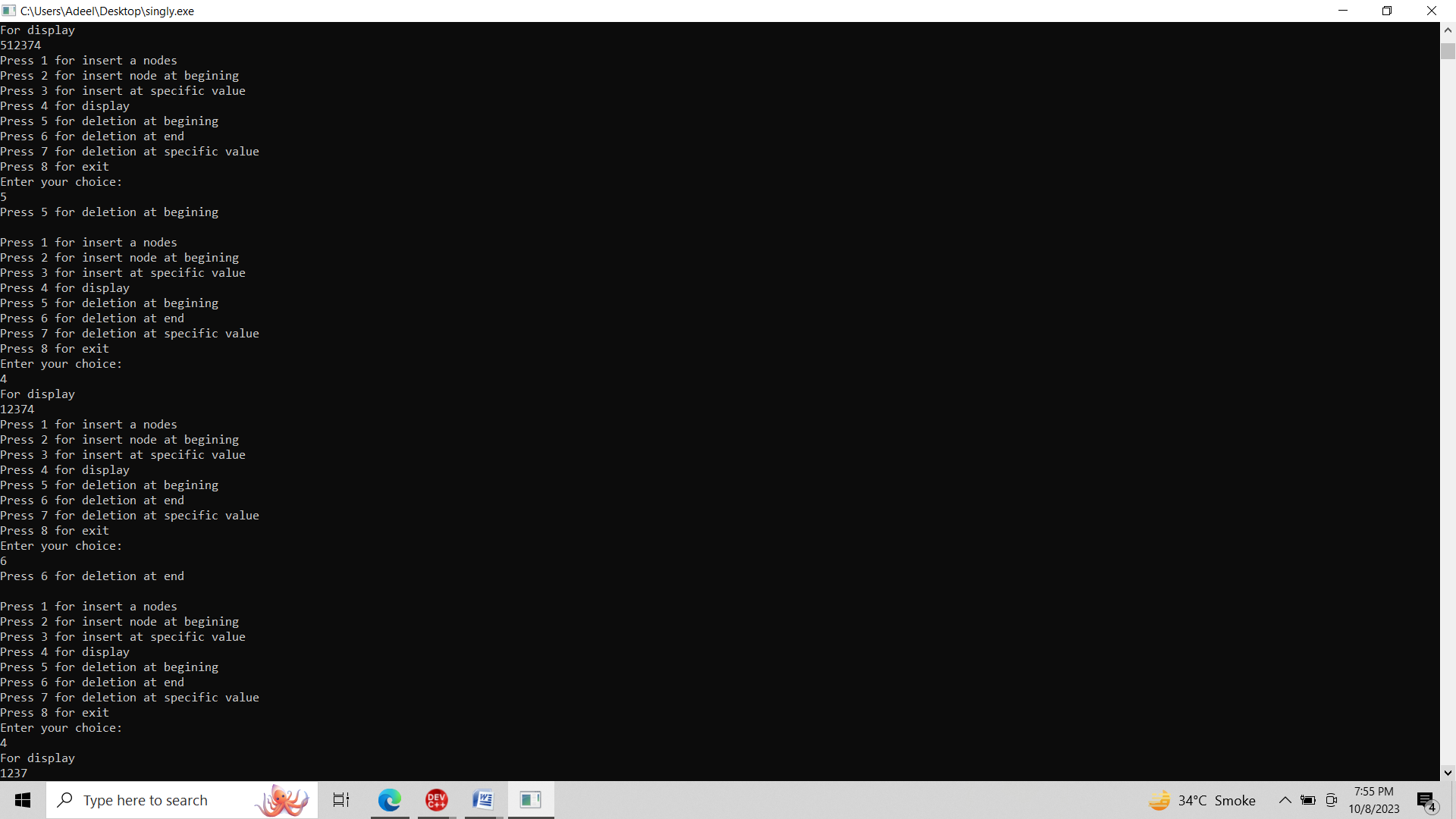
cout<<endl;

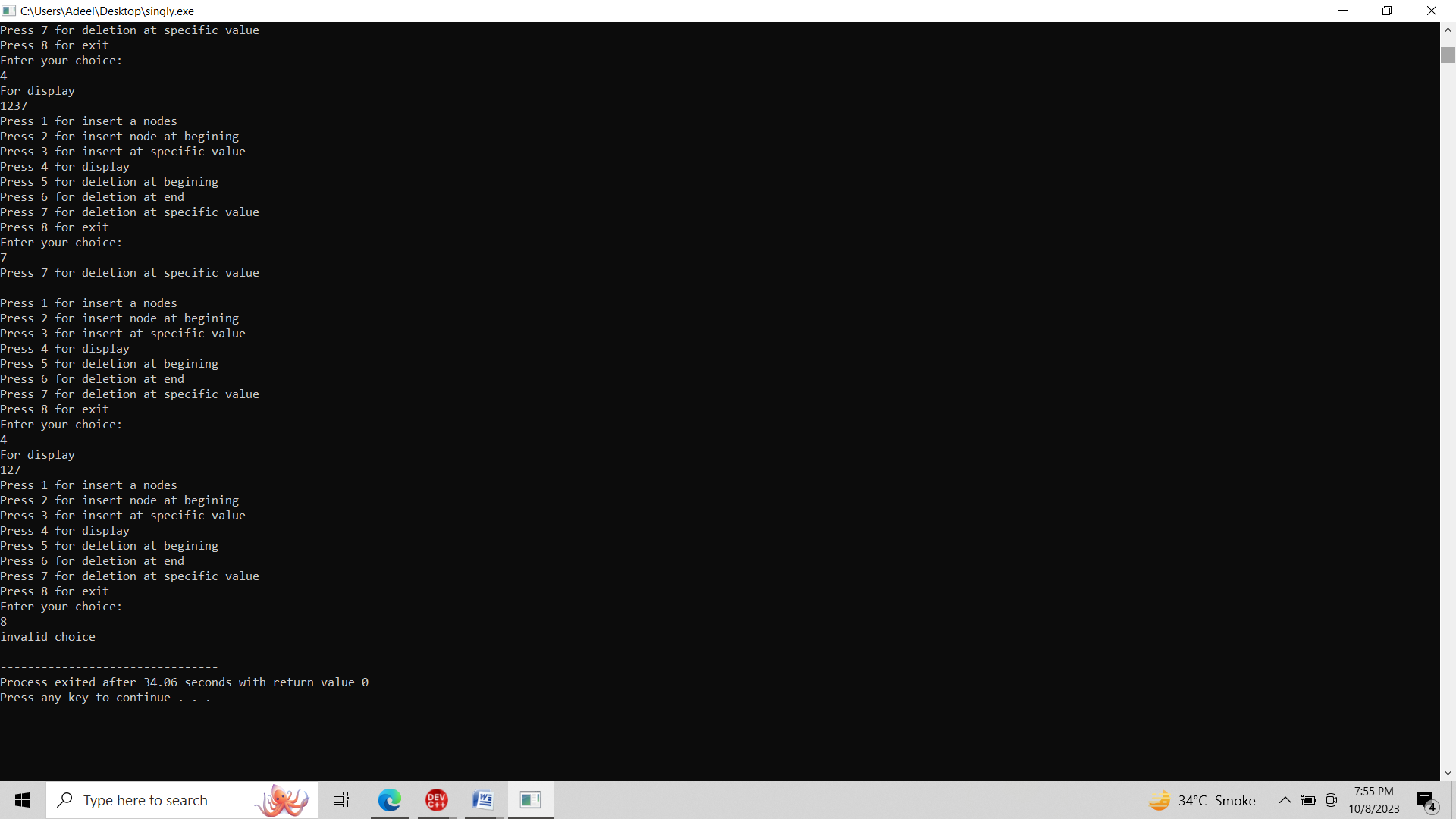
return 0;

}

**Output:**







**Doubly:**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

class Node{

private:

int data;

Node \*next,\*prev;

public:

Node \*head;

Node ()

{

head=NULL;

}

void insert(int n)

{

if (head==NULL){

head= new Node();

head->data=n;

head-> next=NULL;

head-> prev=NULL;

}

else

{

Node \*p,\*ptr;

ptr=head;

while(ptr->next!=NULL){

ptr=ptr->next;

}

p=new Node();

p->data=n;

p->next=NULL;

p->prev=ptr->next;

ptr->next=p;

}

}

void insert\_beg(int n1)

{

if (head==NULL){

head= new Node();

head->data=n1;

head-> next=NULL;

head-> prev=NULL;

}

else

{

Node \*p,\*ptr;

p=new Node();

p->data=n1;

p->next=head;

ptr=head;

ptr->prev=p->next;

head=p;

}

}

void del\_beg(int n1){

if(head==NULL){

cout<<"Empty";

}

else{

Node \*ptr;

ptr=head;

head=ptr->next;

ptr->next->next->prev=NULL;

delete ptr;

ptr=NULL;

}

}

void del\_end(int n1){

if(head==NULL){

head= new Node();

head->data=n1;

head-> next=NULL;

head-> prev=NULL;

}

else{

Node \*ptr,\*p;

ptr=head;

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

ptr=ptr->next;

}

p=ptr->next;

ptr->next=NULL;

delete p;

p=NULL;}

}

void insert\_at\_value( int pos, int n2){

if (head==NULL)

{

head= new Node();

head->data=n2;

head-> next=NULL;

head-> prev=NULL;

}

else

{

Node \*ptr;

ptr=head;

while(ptr->data!=pos){

ptr=ptr->next;

}

Node \*p;

p=new Node();

p->data=n2;

p->next=ptr->next;

ptr->next=p->prev;

ptr->next=p;

}

}

void del\_at\_value(int pos){

if(head==NULL){

cout<<"Empty";

}

else{

Node \*ptr;

ptr=head;

while(ptr->next->data!=pos){

ptr=ptr->next;

}

Node \*p;

p=ptr->next;

ptr->next=NULL;

ptr->next=ptr->next->next;

p->next=NULL;

delete p;

p=NULL; }

}

void display()

{

Node \*ptr;

ptr=head;

while(ptr!=NULL){

cout<<ptr->data;

ptr=ptr->next;

}

}

};

int main(){

Node n;

int ch;

do{

cout<<endl;

cout<<"Press 1 for insert a nodes"<<endl;

cout<<"Press 2 for insert node at begining "<<endl;

cout<<"Press 3 for insert at specific value "<<endl;

cout<<"Press 4 for display"<<endl;

cout<<"Press 5 for deletion at begining"<<endl;

cout<<"Press 6 for deletion at end"<<endl;

cout<<"Press 7 for deletion at specific value"<<endl;

cout<<"Press 8 for exit"<<endl;

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

cin>>ch;

if(ch==1){

cout<<"insert a nodes"<<endl;

n.insert(1);

n.insert(2);

n.insert(3);

n.insert(4);

}

else if(ch==2){

cout<<"insert a nodes at begining"<<endl;

n.insert\_beg(5);

}

else if(ch==3){

cout<<"insert at specific value "<<endl;

n.insert\_at\_value(3,7);

}

else if(ch==4){

cout<<"For display "<<endl;

n.display();

}

else if(ch==5){

cout<<"Press 5 for deletion at begining"<<endl;

n.del\_beg(5);

}

else if(ch==6){

cout<<"Press 6 for deletion at end"<<endl;

n.del\_end(4);

}

else if(ch==7){

cout<<"Press 7 for deletion at specific value"<<endl;

n.del\_at\_value(2);

}

else{

cout<<"invalid choice";

}

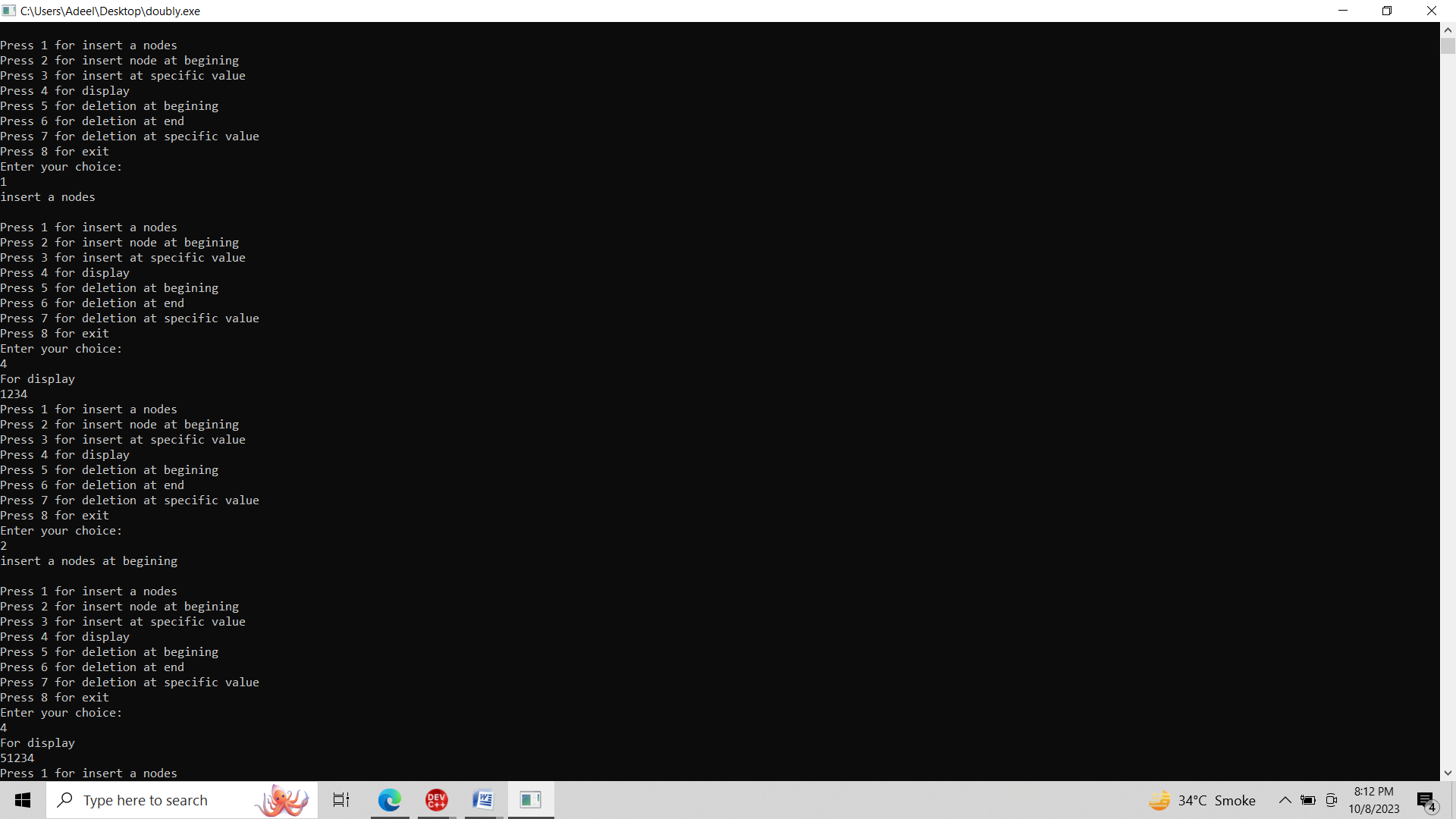
}

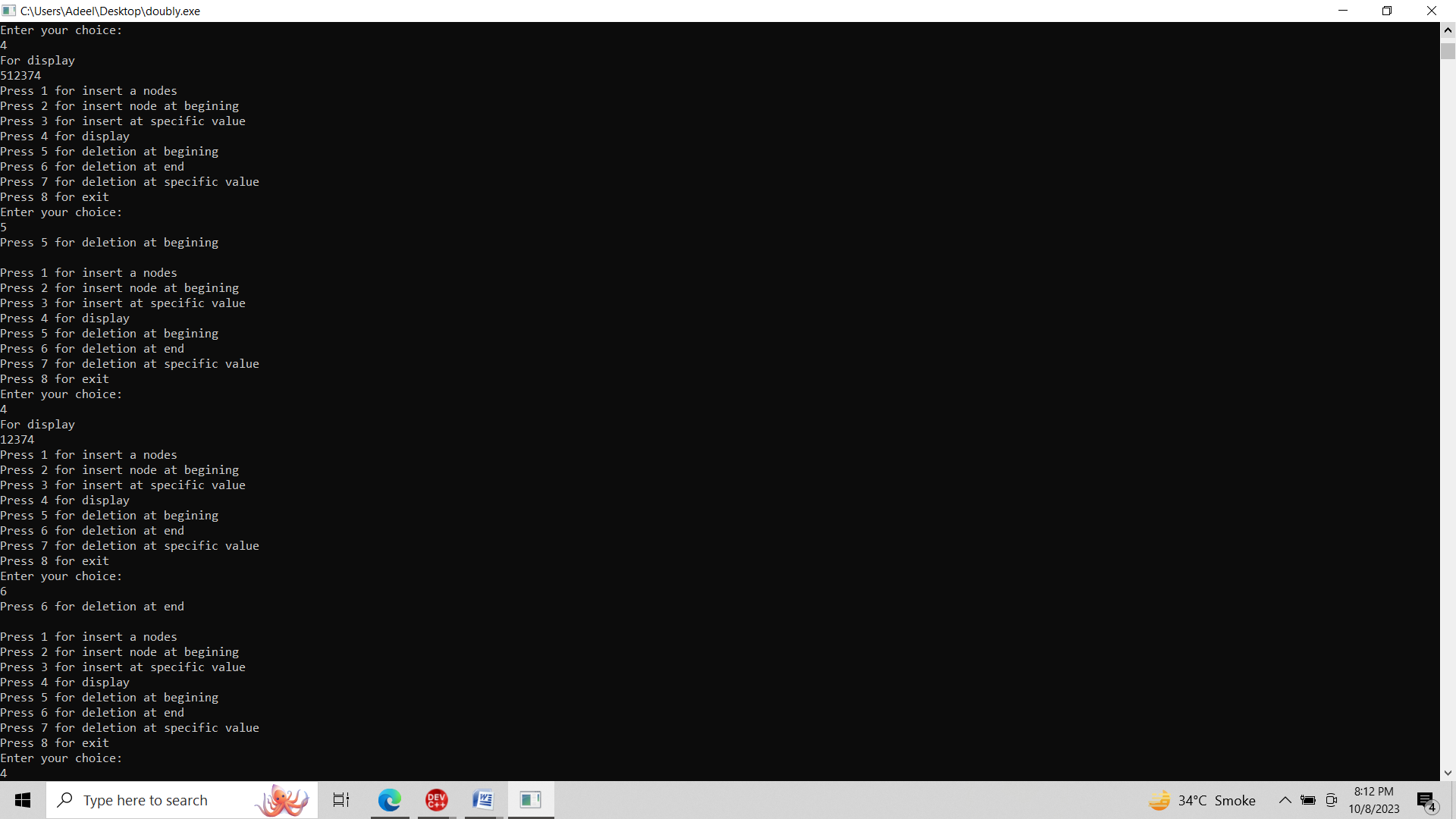
while(ch!=8);

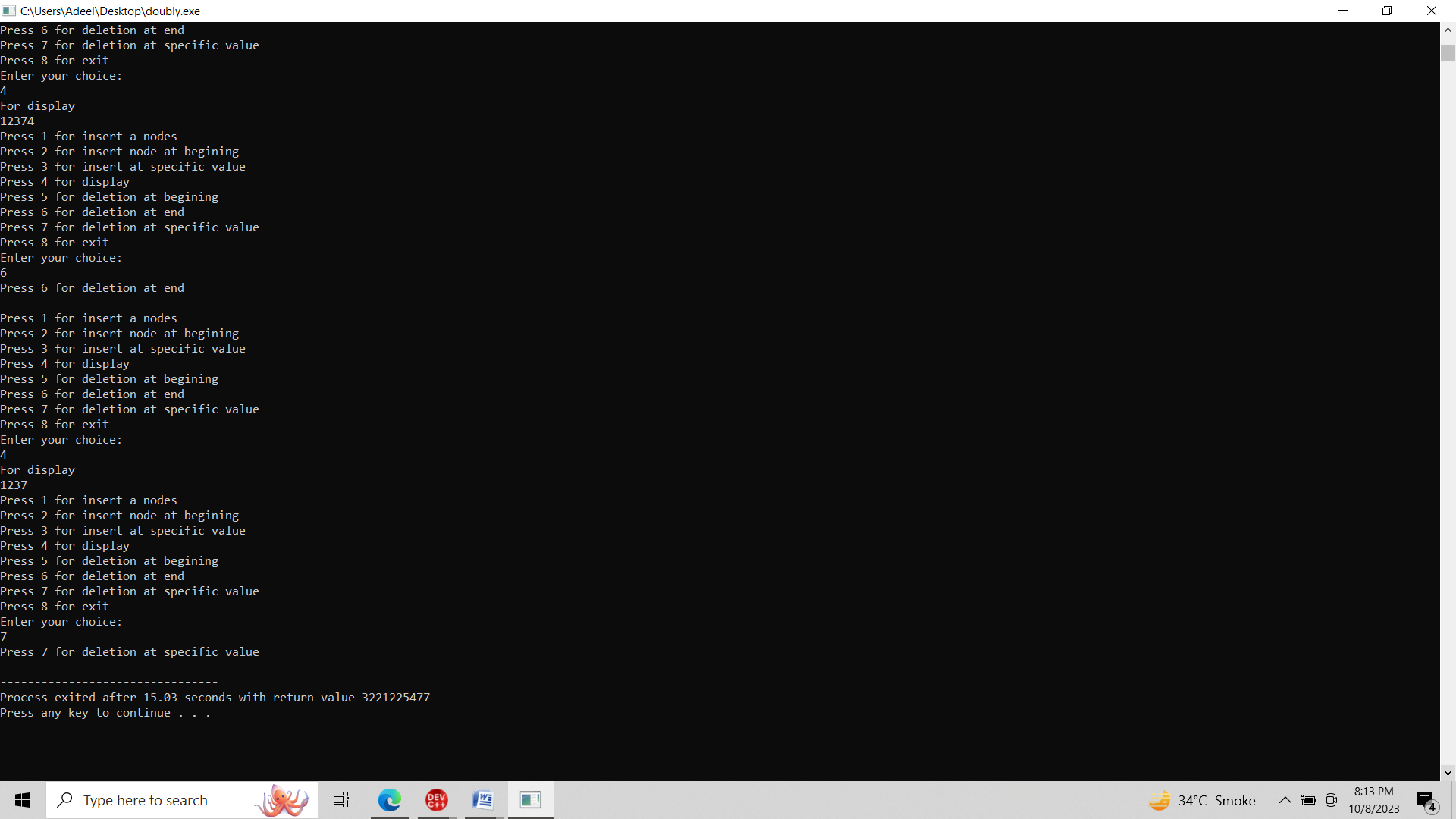
return 0;

}

**Output:**







**Circular:**

#include<iostream>

#include<stdlib.h>

using namespace std;

class Node {

public:

int data;

Node\* next;

Node(int n) {

data=n;

next=nullptr;

}

};

class CircularLinkedList {

private:

Node\* head;

public:

CircularLinkedList() : head(nullptr) {}

void insert(int n) {

Node \*p;

p=new Node(n);

if (head == nullptr) {

head =p;

head->next = head;

}

else {

Node \*ptr = head;

while (ptr->next != head) {

ptr = ptr->next;

}

ptr->next = p;

p->next = head;

head = p;

}

}

void insert\_end(int n) {

Node \*p = new Node(n);

if (head == nullptr) {

head = p;

head->next = head;

}

else {

Node \*ptr = head;

while (ptr->next != head) {

ptr= ptr->next;

}

ptr->next = p;

p->next = head;

}

}

void insert\_at\_value(int pos, int n) {

Node \*p= new Node(n);

if (head == nullptr) {

cout << "List is empty. Cannot insert at value." << endl;

return;

}

Node \*ptr = head;

while (ptr->data != pos) {

ptr = ptr->next;

if (ptr == head) {

cout << "Value " << pos << " not found in the list." << endl;

return;

}

}

p->next = ptr->next;

ptr->next = p;

}

void display() {

if (head == nullptr) {

cout << "List is empty." << endl;

return;

}

Node \*ptr = head;

do {

cout << ptr->data << " ";

ptr = ptr->next;

} while (ptr != head);

cout << endl;

}

};

int main() {

CircularLinkedList cll;

int ch;

do{

cout<<endl;

cout<<"Press 1 for insert a nodes"<<endl;

cout<<"Press 2 for insert node at end "<<endl;

cout<<"Press 3 for insert at specific value "<<endl;

cout<<"Press 4 for display"<<endl;

cout<<"Press 5 for exit"<<endl;

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

cin>>ch;

if(ch==1){

cout<<"insert a nodes"<<endl;

cll.insert(1);

cll.insert(2);

cll.insert(3);

cll.insert(4);

}

else if(ch==2){

cout<<"insert a nodes at end"<<endl;

cll.insert\_end(5);

}

else if(ch==3){

cout<<"insert at specific value "<<endl;

cll.insert\_at\_value(3, 7);

}

else if(ch==4){

cout<<"For display "<<endl;

cll.display(); }

else{

cout<<"invalid choice";

}

}

while(ch!=5);

return 0;

}

**Output:**

