DSA LAB TASK

**Question 1**

#include <iostream>

using namespace std;

int main() {

int var = 10;

int\* ptr = &var;

cout << "Initial values:" << endl;

cout << "Variable value: " << var << endl;

cout << "Pointer value: " << \*ptr << endl;

\*ptr = 20;

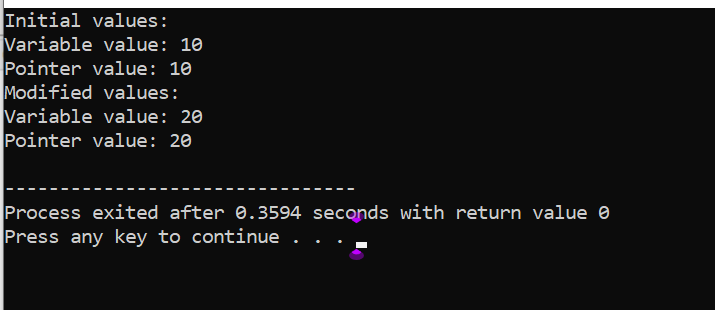
cout << "Modified values:" << endl;

cout << "Variable value: " << var << endl;

cout << "Pointer value: " << \*ptr << endl;

return 0;

}



**Question 2**

#include <iostream>

using namespace std;

int findMax(int arr[], int n) {

int maxVal = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] > maxVal) maxVal = arr[i];

}

return maxVal;

}

int main() {

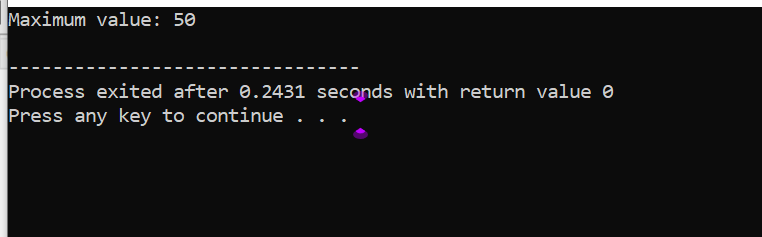
int arr[] = {10, 20, 30, 40, 50};

int n = sizeof(arr) / sizeof(arr[0]);

cout << "Maximum value: " << findMax(arr, n) << endl;

return 0;

}



**Question 3**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head = NULL;

void insert(int data, bool atStart) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = NULL;

if (atStart) {

newNode->next = head;

head = newNode;

} else {

if (head == NULL) head = newNode;

else {

Node\* temp = head;

while (temp->next != NULL) temp = temp->next;

temp->next = newNode;

}

}

}

void display() {

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

int main() {

insert(10, true);

cout << "List after inserting 10 at start: ";

display();

insert(20, true);

cout << "List after inserting 20 at start: ";

display();

insert(30, false);

cout << "List after inserting 30 at end: ";

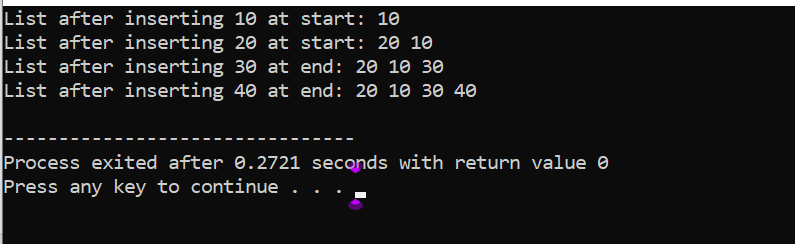
display();

insert(40, false);

cout << "List after inserting 40 at end: ";

display();

return 0;

}

**Question 4**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head = NULL;

void insert(int data, int pos) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = NULL;

if (pos == 0) {

newNode->next = head;

head = newNode;

} else {

Node\* temp = head;

for (int i = 0; i < pos - 1; i++) temp = temp->next;

newNode->next = temp->next;

temp->next = newNode;

}

}

void display() {

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

int main() {

insert(10, 0);

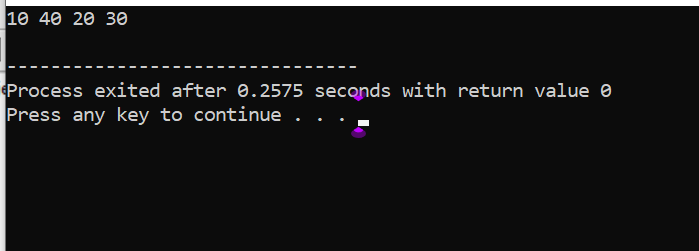
insert(20, 1);

insert(30, 2);

insert(40, 1);

display();

return 0;

}

**Question 5**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head;

void insert(int data, int pos) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = NULL;

if (pos == 0) {

newNode->next = head;

head = newNode;

} else {

Node\* temp = head;

for (int i = 0; i < pos - 1; i++) temp = temp->next;

newNode->next = temp->next;

temp->next = newNode;

}

}

void display() {

Node\* temp = head;

while (temp) {

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

temp = temp->next;

}

}

int main() {

head = NULL;

insert(10, 0);

insert(20, 1);

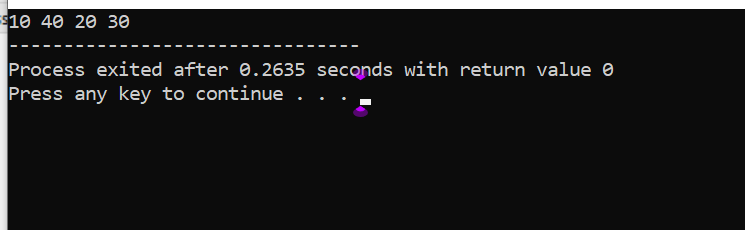
insert(30, 2);

insert(40, 1);

display();

return 0;

}



**Question 6**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head = NULL;

void insert(int data) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = head;

head = newNode;

}

void displayFirst() {

cout << head->data << endl;

}

void displayLast() {

Node\* temp = head;

while (temp->next) temp = temp->next;

cout << temp->data << endl;

}

int main() {

insert(10);

insert(20);

insert(30);

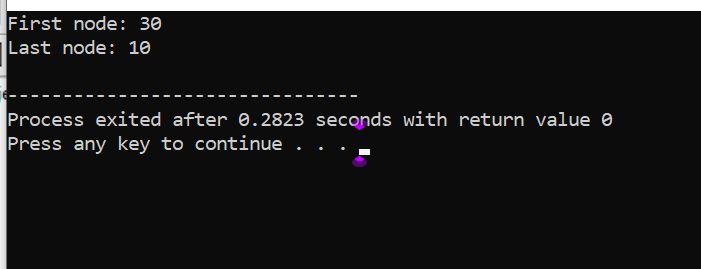
cout << "First node: ";

displayFirst();

cout << "Last node: ";

displayLast();

return 0;

}

**Question 7**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head = NULL;

void insert(int data) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = head;

head = newNode;

}

void displayFirst() {

cout << head->data << endl;

}

void displayLast() {

Node\* temp = head;

while (temp->next) temp = temp->next;

cout << temp->data << endl;

}

int main() {

insert(10);

insert(20);

insert(30);

cout << "First node: ";

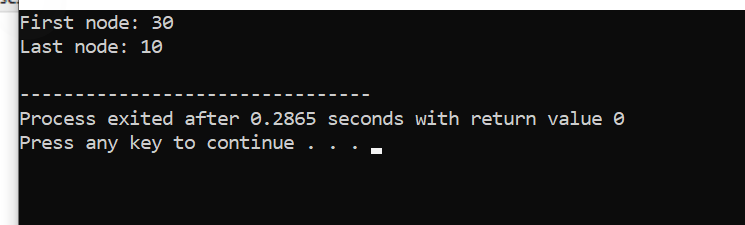
displayFirst();

cout << "Last node: ";

displayLast();

return 0;

}



**Question 8**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

Node\* prev;

};

Node\* head = NULL;

void insert(int data) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = NULL;

newNode->prev = NULL;

if (head == NULL) {

head = newNode;

} else {

Node\* temp = head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

newNode->prev = temp;

}

}

void display() {

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

int main() {

insert(10);

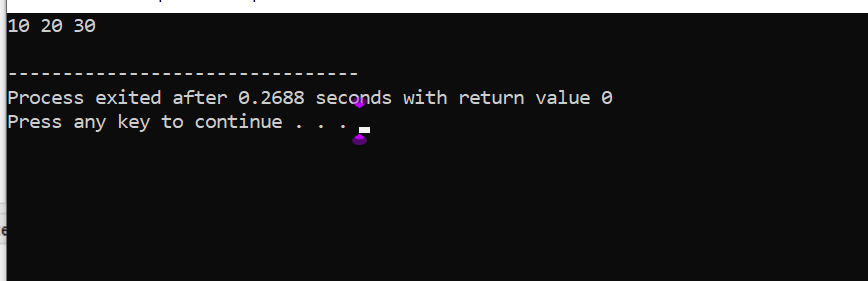
insert(20);

insert(30);

display();

return 0;

}



**Question 8 (b)**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head1 = NULL;

Node\* head2 = NULL;

void insert(Node\* &head, int data) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = head;

head = newNode;

}

void display(Node\* head) {

Node\* temp = head;

while (temp) {

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

temp = temp->next;

}

cout << endl;

}

void merge(Node\* &head1, Node\* &head2) {

Node\* temp = head1;

while (temp->next) temp = temp->next;

temp->next = head2;

}

int main() {

insert(head1, 10);

insert(head1, 20);

insert(head1, 30);

insert(head2, 5);

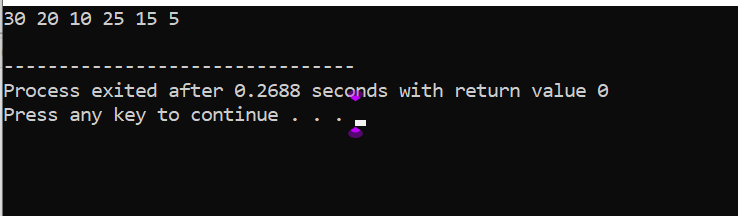
insert(head2, 15);

insert(head2, 25);

merge(head1, head2);

display(head1);

return 0;

}

**Question 9**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* head1 = NULL;

Node\* head2 = NULL;

void insert(Node\* &head, int data) {

Node\* newNode = new Node;

newNode->data = data;

newNode->next = head;

head = newNode;

}

void display(Node\* head) {

while (head) {

cout << head->data << " ";

head = head->next;

}

cout << endl;

}

void merge() {

Node\* temp = head1;

while (temp->next) temp = temp->next;

temp->next = head2;

}

int main() {

insert(head1, 10);

insert(head1, 20);

insert(head1, 30);

insert(head2, 5);

insert(head2, 15);

insert(head2, 25);

merge();

display(head1);

return 0;

}

