Week 5 Lab 1

1. **Write a program that creates a linked list of 10 characters, and then reverse the order of the list. (Creating a Linked List, Then Reversing Its Elements).**

**Code:**

#include <stdio.h>

//Represent a node of linked list

struct node{

int data;

struct node \*next;

};

//Represent the head and tail of the singly linked list

struct node \*head, \*tail = NULL;

//addNode() will add a new node to the list

void addNode(int data) {

//Create a new node

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

newNode->data = data;

newNode->next = NULL;

//Checks if the list is empty

if(head == NULL) {

//If list is empty, both head and tail will point to new node

head = newNode;

tail = newNode;

}

else {

//newNode will be added after tail such that tail's next will point to newNode

tail->next = newNode;

//newNode will become new tail of the list

tail = newNode;

}

}

//reverse() will the reverse the order of the list

void reverse(struct node \*current) {

//Checks if list is empty

if(head == NULL) {

printf("List is empty\n");

return;

}

else{

//Checks if the next node is null, if yes then prints it.

if(current->next == NULL) {

printf("%d ", current->data);

return;

}

//Recursively calls the reverse function

reverse(current->next);

printf("%d ", current->data);

}

}

//display() will display all the nodes present in the list

void display() {

//Node current will point to head

struct node \*current = head;

if(head == NULL) {

printf("List is empty\n");

return;

}

while(current != NULL) {

//Prints each node by incrementing pointer

printf("%d ", current->data);

current = current->next;

}

printf("\n");

}

int main()

{

//Add nodes to the list

addNode(1);

addNode(2);

addNode(3);

addNode(4);

addNode(5);

addNode(6);

addNode(7);

addNode(8);

addNode(9);

addNode(10);

printf("Original List: \n");

display();

printf("Reversed List: \n");

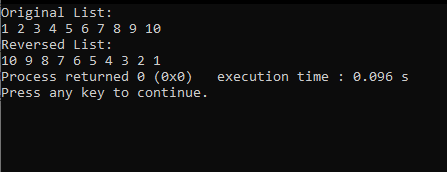
//Print reversed list

reverse(head);

return 0;

}

**Output:**



1. **Write a function printlistBackward that recursively outputs the items in a list in reverse order. Use your function in a test program that creates a sorted list of integers and prints the list in reverse order. (Recursively Print a List Backward)**

**Code:**

#include <bits/stdc++.h>

using namespace std;

// Function to return the reverse

int printlistBackward(int num)

{

int rev\_num = 0;

while (num > 0) {

rev\_num = rev\_num \* 10 + num % 10;

num = num / 10;

}

return rev\_num;

}

// Function to sort the array according to

// the reverse of elements

void sortArr(int arr[], int n)

{

// Vector to store the reverse

vector<pair<int, int> > vp;

// Inserting reverse with elements

// in the vector pair

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

vp.push\_back(

make\_pair(printlistBackward(arr[i]),

arr[i]));

}

// Sort the vector, this will sort the pair

// according to the reverse of elements

sort(vp.begin(), vp.end());

// Print the sorted vector content

for (int i = 0; i < vp.size(); i++)

cout << vp[i].second << " ";

}

// Driver code

int main()

{

int arr[] = { 15,5,78,445,256,412,55,62,45,213 };

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

sortArr(arr, n);

return 0;

}

**Output:**

