DSA Experiment - 7

Harshit Saraswat | B – 42 | 500097101

**Question 1**: Sort the List of Elements stored in a Linked List using the Bubble sort algorithm

**CODE:**

#include <stdio.h>

#include <stdlib.h>

struct node{

int data;

struct node \* next;

};

struct node \* head = NULL;

struct node \* new\_node = NULL;

struct node \* ptr = NULL;

int main()

{

while (1 == 1){

int choice = 0;

printf("1 - Insert Nodes\n2 - Bubble Sort\n3 - Display Linked List\n4 - Quit\nChoice : ");

scanf("%d",&choice);

switch(choice){

case 1 : insert\_new\_node();break;

case 2 : bubble\_sort();break;

case 3 : print\_list();break;

case 4 : exit(0);

}

}

return 0;

}

void bubble\_sort(){

struct node \* ptr2;

int flag = 1;

while (flag == 1){

flag = 0;

for (ptr = head; ptr->next!=NULL;ptr = ptr->next){

if(ptr->data > ptr->next->data){

swap\_nodes(ptr,ptr->next);

flag = 1;

}

}

}

}

void swap\_nodes(struct node \* node1,struct node \* node2){

int temp = node1->data;

node1->data = node2->data;

node2->data = temp;

}

void insert\_new\_node(){

int data;

printf("Enter Data To Insert : ");

scanf("%d",&data);

new\_node = (struct node \*)malloc(sizeof(struct node));

new\_node->data = data;

if(head == NULL){

head = new\_node;

head->next = NULL;

return;

}

for (ptr = head; ptr->next!=NULL; ptr = ptr->next);

ptr->next = new\_node;

new\_node->next = NULL;

}

void print\_list(){

printf("\n");

for (ptr = head; ptr !=NULL; ptr = ptr->next){

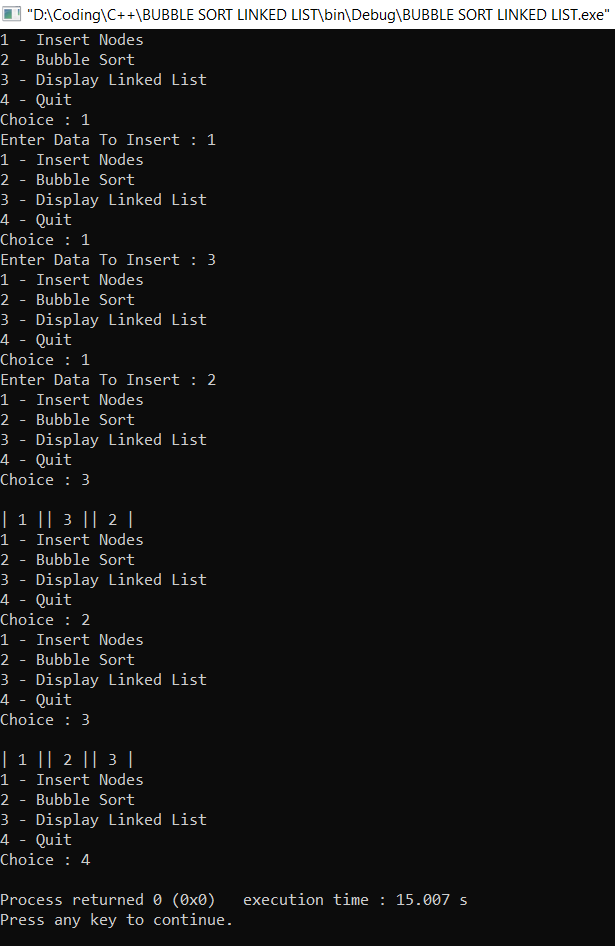
printf("| %d |",ptr->data);

}

printf("\n");

}

**Output :**



**Question 2 :** Sort the List of Elements stored in an Array using the following algorithms: Insertion Sort, Selection Sort, Merge Sort, Quick Sort.

**Code :**

#include <stdio.h>

#include <stdlib.h>

int \* my\_array;

int size;

int main()

{

printf("Enter Size of Array : ");

scanf("%d",&size);

my\_array = (int \*)malloc(size\*sizeof(int));

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

printf("Enter value of Index %d : ",i);

scanf("%d",&my\_array[i]);

}

print\_array();

selection\_sort();

print\_array();

return 0;

}

void print\_array(){

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

printf("| %d |",my\_array[i]);

}

printf("\n");

}

void insertion\_sort(){

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

int old\_index = i;

while(my\_array[i] < my\_array[i-1] && i != 0){

swap\_elements(i,i-1);

i = i-1;

}

i = old\_index;

}

}

void swap\_elements(int i1,int i2){

int temp = my\_array[i1];

my\_array[i1] = my\_array[i2];

my\_array[i2] = temp;

}

void selection\_sort(){ //SWAP WITH LOWEST

int i = 0;

int j = 1;

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

int j = i+1;

int lowest = j;

for (j = i+1;j < size;j++){

if (my\_array[j] < my\_array[lowest])

lowest = j;

}

printf("de %d",lowest);

if (my\_array[lowest] < my\_array[i]){

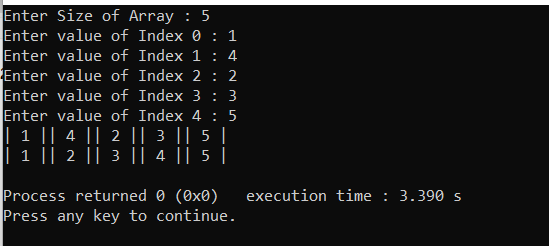
swap\_elements(i,lowest);

}

}

}

**Output :**

****