**Implementation of Stack using Array**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 10

void push(int);

void pop();

void display();

int stack[SIZE], top = -1;

int main() {

int value, choice;

while(1) {

printf("\n\n\*\*\*\*\* MENU \*\*\*\*\*\n");

printf("1. Push\n2. Pop\n3. Display\n4. Exit");

printf("\nEnter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

printf("Enter the value to insert: ");

scanf("%d", &value);

push(value);

break;

case 2:

pop();

break;

case 3:

display();

break;

case 4:

printf("Exiting program.\n");

exit(0);

default:

printf("\nWrong selection! Try again!");

}

}

return 0;

}

void push(int value) {

if(top == SIZE - 1)

printf("\nStack is Full! Insertion not possible.");

else {

top++;

stack[top] = value;

printf("\nInsertion success!");

}

}

void pop() {

if(top == -1)

printf("\nStack is Empty! Deletion not possible.");

else {

printf("\nDeleted: %d", stack[top]);

top--;

}

}

void display() {

if(top == -1)

printf("\nStack is Empty!");

else {

int i;

printf("\nStack elements are:\n");

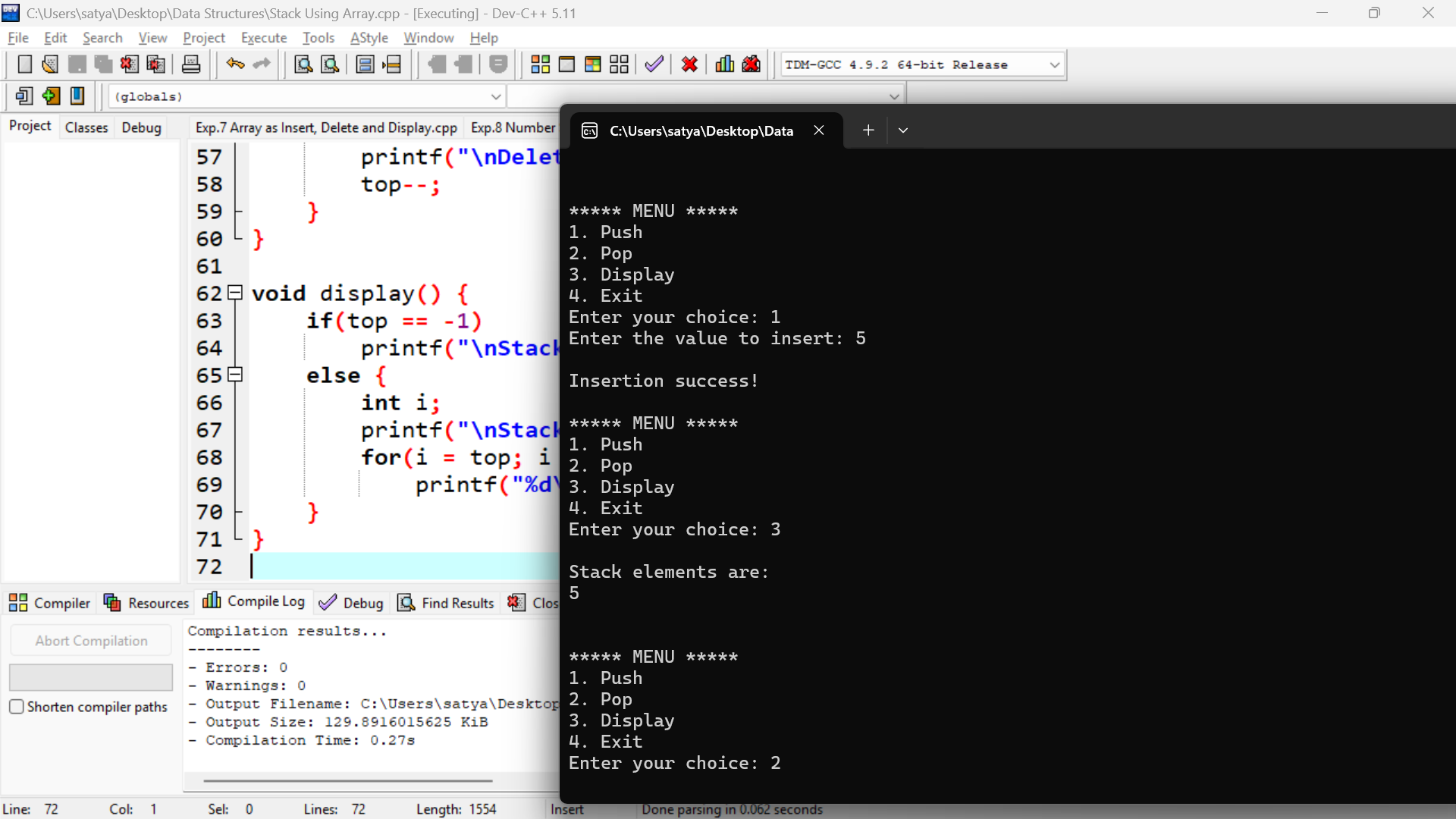
for(i = top; i >= 0; i--)

printf("%d\n", stack[i]);

}

}

**Output**



**Implementation of Stack using Linked List | C Programming**

#include <stdio.h>

#include <stdlib.h>

struct Node { int data; struct Node \*next; } \*top = NULL;

void push(int val) {

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

n->data = val; n->next = top; top = n;

printf("Inserted %d\n", val);

}

void pop() {

if(!top) { printf("Stack Empty\n"); return; }

struct Node \*temp = top; printf("Deleted %d\n", temp->data);

top = top->next; free(temp);

}

void display() {

if(!top) { printf("Stack Empty\n"); return; }

struct Node \*t = top; while(t){ printf("%d ", t->data); t = t->next; }

printf("\n");

}

int main() {

int ch, val;

while(1){

printf("1.Push 2.Pop 3.Display 4.Exit: "); scanf("%d",&ch);

switch(ch){

case 1: scanf("%d",&val); push(val); break;

case 2: pop(); break;

case 3: display(); break;

case 4: exit(0);

default: printf("Invalid choice\n");

}

}

}

**Output**

