Experiment-13 : Applications Of Stack

13.Write a C Program To Implement The Application Of Stack.

Code :

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 100

typedef struct {

int data[MAX\_SIZE];

int top;

} Stack;

void initialize(Stack \*stack) {

stack->top = -1;

}

int isEmpty(Stack \*stack) {

return stack->top == -1;

}

int isFull(Stack \*stack) {

return stack->top == MAX\_SIZE - 1;

}

void push(Stack \*stack, int element) {

if (isFull(stack)) {

printf("Stack Overflow\n");

return;

}

stack->data[++stack->top] = element;

}

int pop(Stack \*stack) {

if (isEmpty(stack)) {

printf("Stack Underflow\n");

return -1;

}

return stack->data[stack->top--];

}

int peek(Stack \*stack) {

if (isEmpty(stack)) {

printf("Stack is empty\n");

return -1;

}

return stack->data[stack->top];

}

void display(Stack \*stack) {

if (isEmpty(stack)) {

printf("Stack is empty\n");

return;

}

printf("Stack elements: ");

for (int i = stack->top; i >= 0; i--) {

printf("%d ", stack->data[i]);

}

printf("\n");

}

int main() {

Stack stack;

initialize(&stack);

push(&stack, 10);

push(&stack, 20);

push(&stack, 30);

push(&stack, 40);

display(&stack);

printf("Top element: %d\n", peek(&stack));

printf("Popped element: %d\n", pop(&stack));

printf("Popped element: %d\n", pop(&stack));

display(&stack);

return 0;

}

Output :

