

**Lab No: 2 Date: 2081/**

**Title: Write a menu based program to show the basic operation of Stack.**

A Stack is a linear data structure that holds a linear, ordered sequence of elements. It is an abstract data type. A Stack works on the LIFO process (Last In First Out), i.e., the element that was inserted last will be removed first. To implement the Stack, it is required to maintain a pointer to the top of the Stack, which is the last element to be inserted because we can access the elements only on the top of the Stack.

**Application of the Stack**

1. A Stack can be used for evaluating expressions consisting of operands and operators.
2. Stacks can be used for Backtracking, i.e., to check parenthesis matching in an expression.
3. It can also be used to convert one form of expression to another form.

**Advantages of Stack**

1. A Stack helps to manage the data in the ‘Last in First out’ method.
2. When the variable is not used outside the function in any program, the Stack can be used.
3. It allows you to control and handle memory allocation and deallocation.

**Disadvantages of Stack**

1. It is difficult in Stack to create many objects as it increases the risk of the Stack overflow.
2. It has very limited memory.

**IDE: Visual Studio Code**

**Language: C**

**Source code:**

#include <stdio.h>

#include <stdlib.h>

#define MAX 5 // array size

struct stack

{

    int arr[MAX];

    int top;

};

void push(struct stack \*sq, int item)

{

    if (sq->top == MAX - 1)

    {

        printf("\n The stack is overflow...\n");

    }

    else

    {

        sq->top++;

        sq->arr[sq->top] = item;

        printf("%d is push in stack\n", item);

    }

}

void pop(struct stack \*sq)

{

    if (sq->top == -1)

    {

        printf("\n The stack is underflow...\n");

    }

    else

    {

        printf("%d is pop from stack\n", sq->arr[sq->top]);

        sq->top--;

    }

}

void display(struct stack \*sq)

{

    if (sq->top == -1)

    {

        printf("Stack is empty...\n");

    }

    else

    {

        printf("The data of stack is:\n");

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

        {

            printf("%d,", sq->arr[i]);

        }

    }

}

int main()

{

    struct stack sq;

    sq.top = -1;

    int c, choose, item;

    while (1)

    {

        printf("\n\n1.For push\n");

        printf("2. For pop\n");

        printf("3. For display\n");

        printf("4.For exit\n");

        printf("Choose number(1-4):- ");

        scanf("%d", &choose);

        switch (choose)

        {

        case 1:

            printf("Enter a number to push:-");

            scanf("%d", &item);

            push(&sq, item);

            break;

        case 2:

            pop(&sq);

            break;

        case 3:

            display(&sq);

            break;

        case 4:

            printf("\n\nProgram END...\n");

            exit(0);

            break;

        default:

            printf("Wrong Entry!!!\n Please choose (1-4) only\n");

            break;

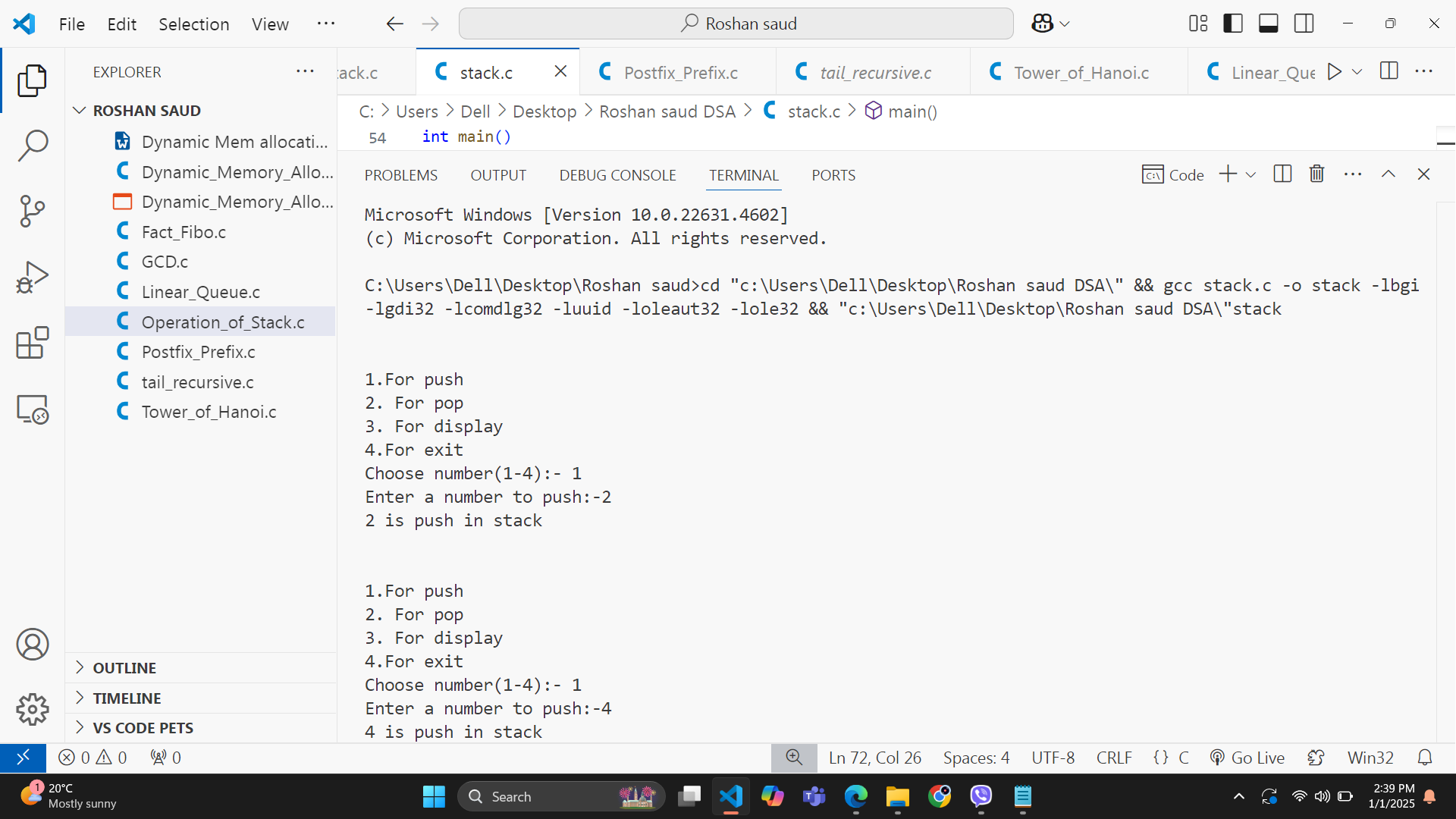
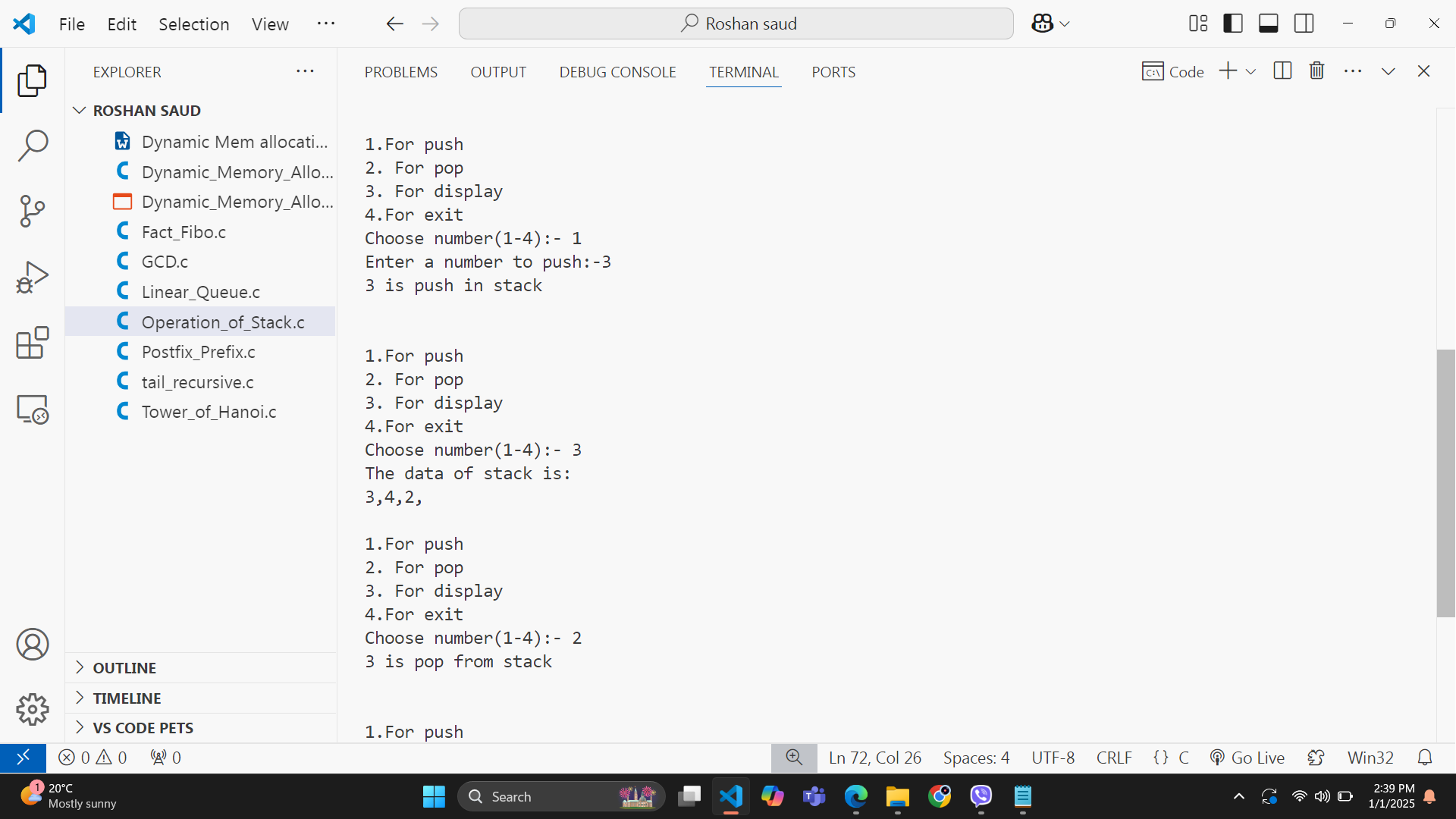
        }

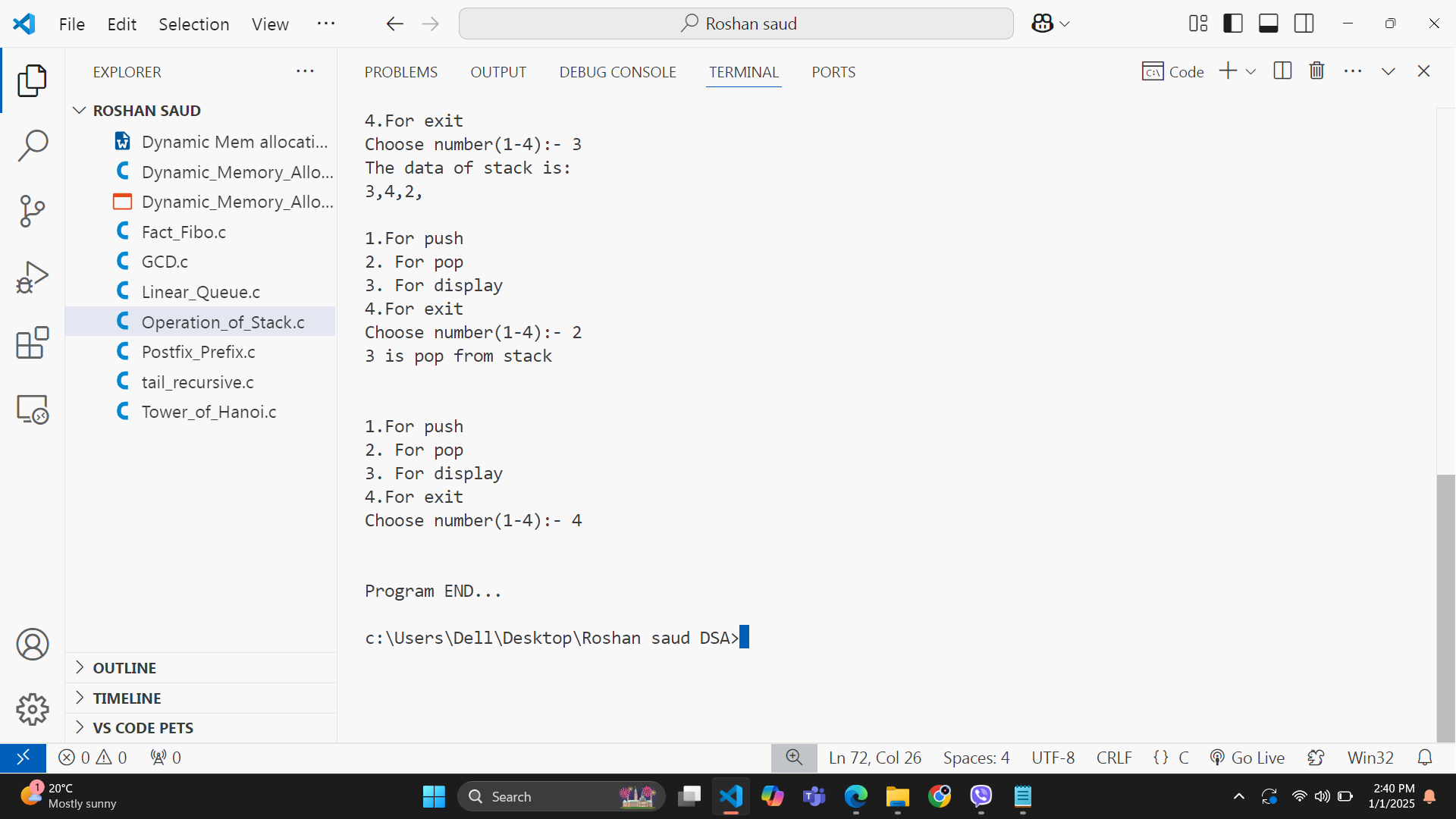
    }

    return 0;

}

**Output:**

****

****