DATA STRUCTURE PRACTICAL NO. :-05

Aim :- [a] Implement a Stack and perform the stack operations: Push, Pop and Print using Menu Driver Program such as 1.Push, 2.Pop and 3. Print and 4. Exit. [Using array]

[b] Implement a Stack and perform the stack operations: Push, Pop and Print using Menu Driver Program such as 1.Push, 2.Pop and 3. Print and 4. Exit. [using linklist]

[a] PROGRAM :-

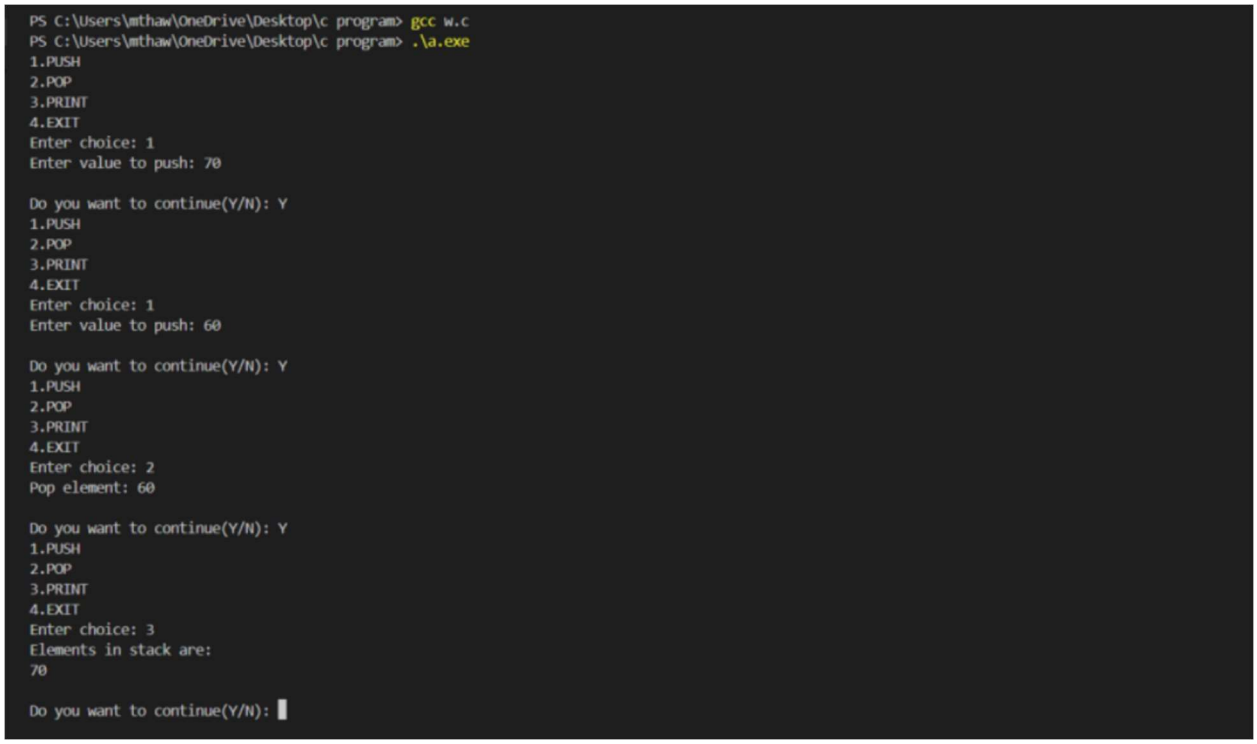
// Implement a Stack and perform the stack operations: Push, Pop and Print using Menu Driver Program such as 1.Push, 2.Pop and 3. Print and 4. Exit. [Using array]

```
#include <stdio.h>


#define MAX 100

int stack[MAX];

int top = -1;


void menu()

{

   printf("1.PUSH\n2.POP\n3.PRINT\n4.EXIT\n");

}


void PUSH()

{

   if(top > MAX)

   {
```

```
        printf("Stack Overflow\n");
        return;
    }
    top += 1;
    printf("Enter value to push: ");
    int a;
    scanf("%d", &a);
    stack[top] = a;

}

void POP()
{
    if(top < 0)
    {
        printf("Stack Underflow\n");
        return;
    }
    printf("Pop element: %d\n", stack[top]);
    top -= 1;

}

void PRINT()
```

```
{
    if(top == -1)
    {
        printf("No Element in Stack\n");
        return;
    }
    printf("Elements in stack are:\n");
    for(int i = top;i >= 0; i--){
        printf("%d \n", stack[i]);
    }
}

int main()
{
    char ch;
    do
    {

    menu();
    int choice;
    printf("Enter choice: ");
    scanf("%d", &choice);

    switch (choice)
```

```
    {
    case 1:
        PUSH();
        break;
    case 2:
        POP();
        break;
    case 3:
        PRINT();
        break;
    case 4:
        return 0;
    default:
        printf("Invalid Choice\n");
        break;
    }

    printf("\nDo you want to continue(Y/N): ");
    scanf(" %c", &ch);

    } while (ch == 'y' || ch == 'Y');
    return 0;
}
```



[b] PROGRAM :-

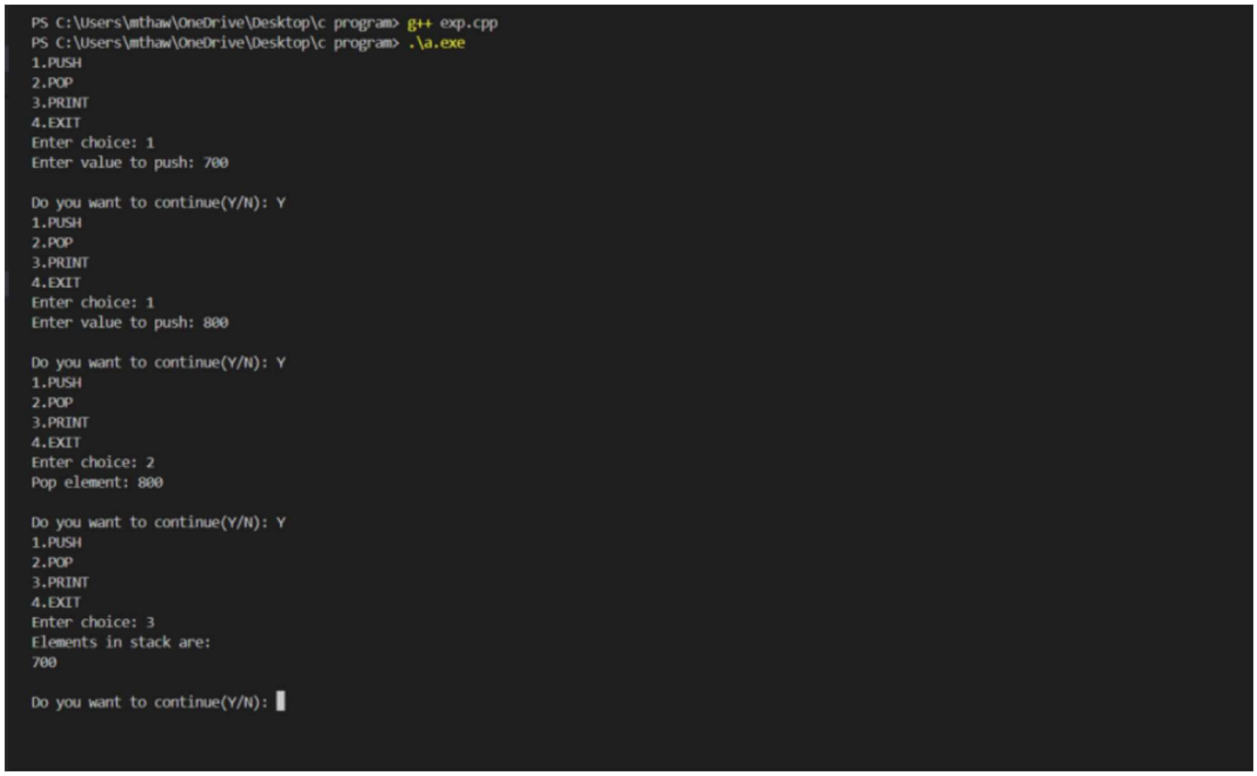
//Implement a Stack and perform the stack operations: Push, Pop and Print using Menu Driver Program such as 1.Push, 2.Pop and 3. Print and 4. Exit. [using linklist]

```
#include <stdio.h>

#include <stdlib.h>


struct Node {

   int data;

   struct Node* next;

};
```

```
struct Node* top = NULL;

void menu() {
    printf("1.PUSH\n2.POP\n3.PRINT\n4.EXIT\n");
}

void PUSH() {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    if (!newNode) {
        printf("Stack Overflow\n");
        return;
    }
    printf("Enter value to push: ");
    scanf("%d", &newNode->data);
    newNode->next = top;
    top = newNode;
}

void POP() {
    if (top == NULL) {
        printf("Stack Underflow\n");
        return;
    }
    struct Node* temp = top;
```

```
    printf("Pop element: %d\n", top->data);
    top = top->next;
    free(temp);
}

void PRINT() {
    if (top == NULL) {
        printf("No Element in Stack\n");
        return;
    }
    struct Node* temp = top;
    printf("Elements in stack are:\n");
    while (temp != NULL) {
        printf("%d \n", temp->data);
        temp = temp->next;
    }
}

int main() {
    char ch;
    do {
        menu();
        int choice;
        printf("Enter choice: ");
```

```
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                PUSH();
                break;
            case 2:
                POP();
                break;
            case 3:
                PRINT();
                break;
            case 4:
                return 0;
            default:
                printf("Invalid Choice\n");
                break;
        }

        printf("\nDo you want to continue(Y/N): ");
        scanf(" %c", &ch);
    } while (ch == 'y' || ch == 'Y');
    return 0;
}
```



GITHUB LINK OF PRACTICAL No. 05 :-
https://github.com/sidheshwar2005/Data_strucutre_practical.git