//Ankit Kumar 2K20/CO/72 //MenuDriven\_StackImplementation.c

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 1000

struct stack {

    int \*a;

    int size;

    int top;

};

int isfull(struct stack \*st) { return (st->top == st->size - 1); }

int isempty(struct stack \*st) { return st->top == -1; }

void display(struct stack \*st) {

    printf("Stack elements : ");

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

        printf("%d\t", st->a[i]);

    printf("\n");

}

void push(struct stack \*st, int x) {

    if (isfull(st)) {

        printf("Stack is Full!\n");

        return;

    }

    st->top++;

    st->a[st->top] = x;

}

void pop(struct stack \*st) {

    if (isempty(st)) {

        printf("Stack is empty!\n");

        return;

    }

    int x = st->a[st->top];

    st->top--;

    printf("%d was popped!\n", x);

}

int main() {

    struct stack \*st = (struct stack \*)malloc(sizeof(struct stack));

    st->a = (int \*)malloc(sizeof(int) \* MAX\_SIZE);

    st->top = -1;

    st->size = MAX\_SIZE;

    int choice;

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

    do {

        int x;

        printf("Enter Your Choice : ");

        scanf("%d", &choice);

        switch (choice) {

        case 1:

            printf("Enter element to be pushed : ");

            scanf("%d", &x);

            push(st, x);

            printf("%d was pushed!\n", x);

            break;

        case 2:

            pop(st);

            break;

        case 3:

            display(st);

            break;

        default:

            break;

        }

    } while (choice != 4);

    return 0;

}

