```
/* DS Lab 3:
Develop a menu driven Program in C for the
following operations on STACK of Integers
(Array Implementation of Stack with maximum size
MAX)
a. Push an Element on to Stack
b. Pop an Element from Stack
C. Demonstrate how Stack can be used to check
Palindrome
d. Demonstrate Overflow and Underflow situations on
Stack
e. Display the status of Stack
f. Exit
Support the program with appropriate functions for
each of the above operations
*/


// Write C Program to implement Stack Operation
Dynamically by Passing Parameters

#include<stdio.h>
#include<stdlib.h>
#include <string.h>

int STACK_SIZE = 1; // Set a Stack size


// Function to insert an item into the Stack at the
top
void push(int item, int stack[], int *top)
{
    //Check for over flow of stack
    if(*top == STACK_SIZE - 1)
    {
        printf("Stack overflow, Reallocating Memory
        to Stack to store an Item...\n");
        STACK_SIZE++;
```

```
            stack = (int *)realloc(stack,STACK_SIZE*
            sizeof(int));
        }

        //insert an item into  the Stack
        stack [++(*top)] = item;
}

// Function to delete an item from the top of the
Stack
void pop(int stack[], int *top)
{
        //Check for Stack under flow
        if (*top == -1)
        {
            printf("Stack underflow\n");
            return; // Indicating the Stack is empty
        }

        printf("Item deleted = %d\n",stack[(*top)--]);

}

// Function to display the elements of the Stack
void display(int stack[], int top)
{

        //Check for empty Stack
        if(top == -1)
        {
            printf("Stack is empty\n");
            return;
        }

        // display contents in an stack
        printf("Stack elements: ");
        for(int i=0;i <= top;i++)
            printf("%d ", stack[i]);
```

```
    printf("\n");
}

// Function to check if a given string is a
palindrome using a stack
void palindrome(char str[])
{
    int len = strlen(str);
    int *stack = (int *)malloc(len * sizeof(int));
    // Separate stack for palindrome check
    int top = -1;

    // Push each character onto the stack
    for (int i = 0; i < len; i++)
        stack[++top] = str[i];

    // Display stack content for palindrome check
    printf("Stack content is: ");
    for (int i = top; i >= 0; i--)
        printf("%c", stack[i]);

    printf("\n");

    // Compare each character of string with stack
    content
    for (int i = 0; i < len; i++)
    {
        if (str[i] != stack[top--])
        {
            printf("%s is not a palindrome\n", str);
            free(stack); // Free allocated memory
            for stack
            return;
        }
    }

    printf("%s is a palindrome\n", str);
```

```
    free(stack); // Free allocated memory for stack
}

void main()
{
    int choice, item, *stack, top = -1;
    char str[25]; // Increased size to accommodate
    larger strings

    stack = (int *)malloc(STACK_SIZE * sizeof(int));

    for (;;)
    {
        printf("1: Push 2: Pop 3: Display 4:
        Palindrome 5: Exit : ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter the item: ");
            scanf("%d", &item);

            push(item, stack, &top);
            break;

        case 2:
            pop(stack, &top);
            break;

        case 3:
            display(stack, top);
            break;

        case 4:
            printf("Enter the text string: ");
            scanf(" %[^\n]", str);
```

```
                palindrome(str);
                break;

            case 5:
                free(stack); // Free allocated memory
                exit(0);

            default:
                printf("Invalid choice. Please try
                again.\n");
            }
        }
}


/*

Output :

1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 8
Invalid choice. Please try again.
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Stack underflow
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack is empty
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 1
Enter the item: 10
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 1
Enter the item: 20
Stack overflow, Reallocating Memory to Stack to
store an Item...
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10 20
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 1
Enter the item: 30
Stack overflow, Reallocating Memory to Stack to
```

```
store an Item...
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10 20 30
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 1
Enter the item: 40
Stack overflow, Reallocating Memory to Stack to
store an Item...
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10 20 30 40
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Item deleted = 40
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10 20 30
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Item deleted = 30
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10 20
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Item deleted = 20
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack elements: 10
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Item deleted = 10
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 3
Stack is empty
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 2
Stack underflow

1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 4
Enter the text string: Gadag
Gadag : is Not a Palindrome
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 4
Enter the text string: gadag
gadag : is Palindrome
1: Push 2: Pop 3: Display 4: Palindrome 5: Exit : 4
Enter the text string: Jayaprada S Hiremath
Stack content is: htameriH S adarpayaJ
Jayaprada S Hiremath is not a palindrome
```

```
1:Push 2:Pop 3:Display 4:Palindrome 5: Exit : 5

...Program finished with exit code 0
Press ENTER to exit console.

*/
```