

**G. H. RAISONI COLLEGE OF ENGG., NAGPUR**  
(An Autonomous Institute under UGC Act 1956)  
**Department of Artificial Intelligence**

---

**Date: 20/07/2020**

**Practical Subject: Data Structures and Algorithms**  
**Session: 2020-21**

---

**Student Details:**

<b>Roll Number</b>	40
<b>Name</b>	Vishal Narnaware
<b>Semester</b>	3
<b>Section</b>	A
<b>Branch</b>	Artificial Intelligence

---

**Practical Details: Practical Number- 2**

<b>Practical Aim</b>	Design, Develop and Implement 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
<b>Theory</b>	<p>A stack is a data structure that holds a list of elements. A stack works based on the LIFO principle-Last In First Out, meaning that the most recently added element is the first one to remove.</p> <p>A stack has two main operations that occur only at the top of the stack: push and pop. The push operation places an element at the top of stack whereas the pop operation removes an element from the top of the stack.</p> <p><b>push ():</b> The push () method allows you to add one or more elements to the end of the array. The push () method returns the value of the length</p>

	<p>property that specifies the number of elements in the array.</p> <p><b>pop ():</b> The pop () method removes the element at the end of the array and returns the element to the caller. If the array is empty, the pop() method returns undefined.</p> <p><b>display ():</b> Display function used to print the values at any time.</p> <p><b>Palindrome number:</b> A palindrome number is a number that is same after reverse.</p> <p><b>Stack overflow:</b> A stack overflow is when you've used up more memory for the stack than your program was supposed to use.</p> <p><b>Stack underflow:</b> An error condition that occurs when an item is called for from the stack, but the stack is empty.</p>
<b>Procedure</b>	<ol style="list-style-type: none"> <li>1. User selects the operation from MENU.</li> <li>2. Push, Pop, Palindrome Check, Display and Exit: MENU</li> <li>3. If the stack gets full while inserting the integer, it gives a message as 'Stack Overflow!' and reset the stack.</li> <li>4. If the stack is empty then it gives a message as 'Stack Underflow!' and asks user to enter an element first.</li> <li>5. Selecting the Palindrome option in the menu, checks if the given input is palindrome or not and displays the appropriate message.</li> <li>6. We can then display the integers in the stack by selecting Display.</li> <li>7. And we can exit the program by selecting the last option.</li> </ol>
<b>Algorithm</b>	<p><b>Step 1:</b> START</p> <p><b>Step 2:</b> Initialize stack size MAX and top as 0.</p> <p><b>Step 3:</b> Display Menu and ask for operation. Do the necessary.</p> <p><b>Step 4:</b> If selected Push, insert integer element on to stack and increment top by 1. If stack is full give a message as</p>

	<p>‘Stack Overflow!’.</p> <p><b>Step 5:</b> If selected Pop, then delete an element from stack and decrement top by 1. If stack is empty give a message as ‘Stack Underflow! Enter an element first...’.</p> <p><b>Step 6:</b> If check, then compare top and bottom elements of stack if they are not equal, update flag with 1 else don’t change.</p> <p><b>Step 7:</b> If flag is 0, then stack contents are palindrome else not.</p> <p><b>Step 8:</b> Display the stack, if required or exit.</p> <p><b>Step 9:</b> STOP</p>
<p><b>Program</b></p>	<pre> stacks.c 1  #include &lt;stdio.h&gt; 2  #include &lt;string.h&gt; 3  #include &lt;stdlib.h&gt; 4  #define MAX 5 // maximum 5 elements can be inserted 5 6  void push(int stack[MAX], int *top, int ele) { 7      // function to insert element into Stack 8      if (*top &gt;= MAX) { 9          printf("\n\tOverflow! Resetting Stack..."); 10         *top = 0; // reset top to 0 11     } 12     else { 13         stack[*top] = ele; 14         *top += 1; // increment top by 1 15     } 16 } 17 int pop(int stack[MAX], int *top) { 18     // function to delete element from Stack 19     if(*top &lt;= 0) { 20         printf("\nUnderflow! Enter elements in Stack..."); 21         return -1; 22     } 23     else { 24         *top -= 1; // decrement top 25         return stack[*top]; 26     } </pre>

```

27 }
28 void check() {
29     // function to check if given number is palindrome or not
30     char s[MAX];
31     int stack[MAX];
32     int i, flag=0, top=0, temp;
33     printf("\nEnter any number: ");
34     scanf("%s", &s);
35     for (i=0; i<strlen(s); i++) {
36         temp = s[i];
37         push(stack, &top, temp);
38     }
39
40     for (i=0; i<strlen(s)/2; i++) {
41         if (stack[i] != pop(stack, &top)) {
42             flag = 1;
43             break;
44         }
45     }
46     if (flag == 1) {
47         printf("\nNumber isn't Palindrome");
48     }
49     else {
50         printf("\nNumber is Palindrome");
51     }
52 }
53
54 void display(int stack[MAX], int *top) {
55     // function to show elements of Stack
56     int i;
57     printf("\nStack : ");
58     for(i=0; i<*top; i++) {
59         printf("%d ", stack[i]);
60     }
61 }
62
63 int main(int argc, char const *argv[]) {
64     int i, top=0, ch, temp;
65     int stack[MAX];
66     printf("\nAuthor: Vishal Narnaware");
67     while(1) { // Repeat again and again
68         printf("\n-----**MENU**-----"); // The driving Menu
69         printf("\n1. Push");
70         printf("\n2. Pop");
71         printf("\n3. Palindrome Check");
72         printf("\n4. Display");
73         printf("\n5. Exit");
74         printf("\nEnter your choice: "); // Ask user for operation
75         scanf("%d", &ch); // Get in ch
76         switch(ch) { // Driving block
77             case 1: printf("\nEnter element: ");
78                     scanf("%d", &temp);
79                     push(stack, &top, temp);

```

	<pre> 79             break; 80         case 2: temp = pop(stack, &amp;top); 81             printf("\nPopped off element: %d", temp);    // display deleted element 82             break; 83         case 3: check(); 84             break; 85         case 4: display(stack, &amp;top); 86             break; 87         case 5: return 0;                                // No need to write break 88         default: printf("\nWrong Choice!!! \t Enter between 1 to 5"); 89             break; 90     } 91 } 92 return 0; 93 } </pre>
Output	<p><b>Push:</b></p> <pre> Author: Uishal Narnaware -----**MENU**----- 1. Push 2. Pop 3. Palindrome Check 4. Display 5. Exit Enter your choice: 1  Enter element: 10  -----**MENU**----- 1. Push 2. Pop 3. Palindrome Check 4. Display 5. Exit Enter your choice: 1  Enter element: 20  -----**MENU**----- 1. Push 2. Pop 3. Palindrome Check 4. Display 5. Exit Enter your choice: 4  Stack : 10 20 </pre> <p><b>Pop:</b></p> <pre> -----**MENU**----- 1. Push 2. Pop 3. Palindrome Check 4. Display 5. Exit Enter your choice: 2  Popped off element: 20 -----**MENU**----- 1. Push 2. Pop 3. Palindrome Check 4. Display 5. Exit Enter your choice: 4  Stack : 10 </pre>

### ***Underflow:***

```
C:\Users\bagde\Desktop\Uishal\C\C-Basics\Practical\Practical2>stacks.exe
Author: Uishal Narnaware
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 2
Underflow! Enter elements in Stack...
Popped off element: -1
```

### ***Overflow:***

```
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 1
Enter element: 5
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 4
Stack : 1 2 30 40 5
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 1
Enter element: 9
Overflow! Resetting Stack...
```

### ***Palindrome Check:***

```
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 3
Enter any number: 303
Number is Palindrome
-----**MENU**-----
1. Push
2. Pop
3. Palindrome Check
4. Display
5. Exit
Enter your choice: 3
Enter any number: 441
Number isn't Palindrome
```

<b>Conclusion</b>	Hence, successfully completed and implemented Stacks using array in a C program and checked whether a given number is palindrome.
-------------------	---