

DAILY ONLINE ACTIVITIES SUMMARY

Date:	23-06-2020	Name:	Anvitha Poojary
Sem & Sec	6A	USN:	4AL17CS008
Online Test Summary			
Subject	PAP assignment test		
Max. Marks	20	Score	10
Certification Course Summary			
Course	JAVA		
Certificate Provider		Duration	
Coding Challenges			
Problem Statement: 1. Write a C Program to Sort a stack using a temporary stack			
Status: completed			
Uploaded the report in Github		yes	
If yes Repository name		https://github.com/anvithapo99/Daily-Report	
Uploaded the report in slack		yes	

Online test details:

Subject:PAP

Largest Tech Community | Hacka... x +

techgig.com/challenge/result/round1/bUVWNzVnVnQrbktWejRhc0s0RFiIz09

Test Completed!

You have successfully participated in PAP Assignment 4.

Rate this Test
Your Rating: ★★★★★ Click to Rate

Results Analytics

✓ Round1

Your Score **10** / 20

This site uses cookies so that we can remember you and understand how you interact with our website. This allows us to improve and customize your browsing experience. To find out more about the cookies we use, see our [Cookies Policy](#). OK

Start a search

03:30 PM 23-06-2020

JAVA Quiz details:

Largest Tech Community | Hacka... x +

techgig.com/challenge/result/mcq/WlxVeTdpQ3I4eGJoeWUrZUVYdn8Kdz09

Test Completed!

You have successfully participated in java-abstract method& interfaces.

Rate this Test
Your Rating: ★★★★★ Click to Rate

Results Analytics

✓ MCQ

Your Score **8** / 10

Start a search

02:57 PM 23-06-2020

Coding Challenges Details:

1. Write a C Program to Sort a stack using a temporary stack

We follow this algorithm.

1. Create a temporary stack say tmpStack.
2. While input stack is NOT empty do this:
 - Pop an element from input stack call it temp
 - while temporary stack is NOT empty and top of temporary stack is greater than temp,
pop from temporary stack and push it to the input stack
 - push temp in temporary stack
3. The sorted numbers are in tmpStack

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct stack
```

```
{
```

```
    int data;
```

```
    struct stack *next;
```

```
};
```

```
void initStack(struct stack **s)
```

```
{
```

```
    *s = NULL;
```

```
}
```

```
int isEmpty(struct stack *s)
```

```
{
```

```
    if (s == NULL)

        return 1;

    return 0;
}
```

```
void push(struct stack **s, int x)
{
    struct stack *p = (struct stack *)malloc(sizeof(*p));
```

```
    if (p == NULL)
    {
        fprintf(stderr, "Memory allocation failed.\n");
        return;
    }
```

```
    p->data = x;
    p->next = *s;
    *s = p;
}
```

```
int pop(struct stack **s)
{
    int x;
    struct stack *temp;
```

```

    x = (*s)->data;

    temp = *s;

    (*s) = (*s)->next;

    free(temp);


    return x;
}


int top(struct stack *s)
{
    return (s->data);
}


void sortedInsert(struct stack **s, int x)
{
    if (isEmpty(*s) || x > top(*s))
    {
        push(s, x);

        return;
    }

    int temp = pop(s);

    sortedInsert(s, x);

```

```
    push(s, temp);  
}
```

```
void sortStack(struct stack **s)  
{  
    if (!isEmpty(*s))  
    {  
        int x = pop(s);  
  
        sortStack(s);  
  
        sortedInsert(s, x);  
    }  
}
```

```
void printStack(struct stack *s)  
{  
    while (s)  
    {  
        printf("%d ", s->data);  
        s = s->next;  
    }  
  
    printf("\n");  
}
```

```
}
```

```
int main(void)
```

```
{
```

```
    struct stack *top;
```

```
    initStack(&top);
```

```
    push(&top, 30);
```

```
    push(&top, -5);
```

```
    push(&top, 18);
```

```
    push(&top, 14);
```

```
    push(&top, -3);
```

```
    printf("Stack elements before sorting:\n");
```

```
    printStack(top);
```

```
    sortStack(&top);
```

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

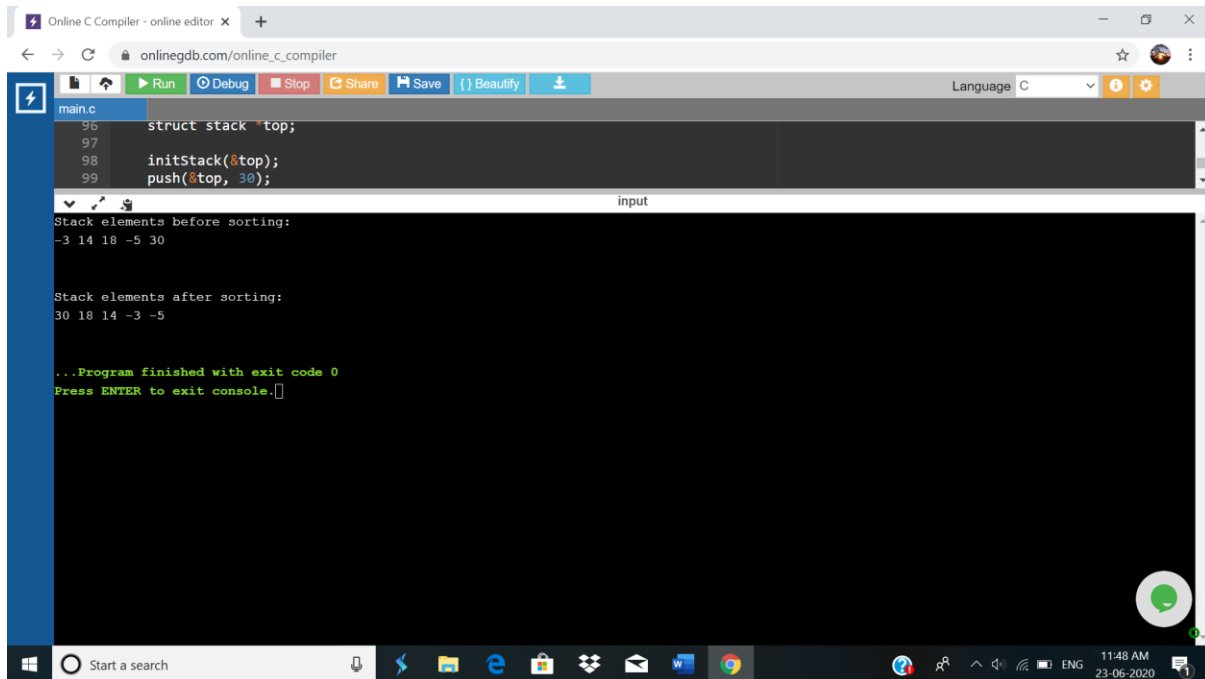
```
    printf("Stack elements after sorting:\n");
```

```
    printStack(top);
```

```
    return 0;
```

```
}
```

Output:



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The browser's address bar and tabs are visible at the top. Below the browser window is a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The main area is divided into two panes. The top pane, titled 'main.c', contains the following C code:

```
96 struct stack *top;
97
98 initStack(&top);
99 push(&top, 30);
```

The bottom pane, titled 'Input', displays the program's output:

```
Stack elements before sorting:
-3 14 18 -5 30

Stack elements after sorting:
30 18 14 -3 -5

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

The Windows taskbar is visible at the bottom of the screen, showing the Start button, a search bar, and several application icons. The system clock in the bottom right corner indicates the time is 11:48 AM on 23-06-2020.