

# Rajalakshmi Engineering College

Name: Kaniya Fathima J  
Email: 240701226@rajalakshmi.edu.in  
Roll no: 240701226  
Phone: 7904195258  
Branch: REC  
Department: I CSE AH  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 3\_MCQ\_Updated

Attempt : 1  
Total Mark : 20  
Marks Obtained : 20

#### Section 1 : MCQ

1. Pushing an element into the stack already has five elements. The stack size is 5, then the stack becomes

**Answer**

Overflow

**Status : Correct**

**Marks : 1/1**

2. Consider a linked list implementation of stack data structure with three operations:

push(value): Pushes an element value onto the stack.  
pop(): Pops the top element from the stack.  
top(): Returns the item stored at the top of the stack.

Given the following sequence of operations:

push(10);pop();push(5);top();

What will be the result of the stack after performing these operations?

**Answer**

The top element in the stack is 5

**Status : Correct**

**Marks : 1/1**

3. Here is an Infix Expression:  $4+3*(6*3-12)$ . Convert the expression from Infix to Postfix notation. The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?

**Answer**

4

**Status : Correct**

**Marks : 1/1**

4. What will be the output of the following code?

```
#include <stdio.h>
#define MAX_SIZE 5
int stack[MAX_SIZE];
int top = -1;
void display() {
    if (top == -1) {
        printf("Stack is empty\n");
    } else {
        printf("Stack elements: ");
        for (int i = top; i >= 0; i--) {
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
}
void push(int value) {
    if (top == MAX_SIZE - 1) {
        printf("Stack Overflow\n");
    }
}
```

```
    } else {  
        stack[++top] = value;  
    }  
}
```

```
int main() {  
    display();  
    push(10);  
    push(20);  
    push(30);  
    display();  
    push(40);  
    push(50);  
    push(60);  
    display();  
    return 0;  
}
```

**Answer**

Stack is empty  
Stack elements: 30 20 10  
Stack Overflow  
Stack elements: 50 40 30 20 10

**Status : Correct**

**Marks : 1/1**

5. Which of the following Applications may use a Stack?

**Answer**

All of the mentioned options

**Status : Correct**

**Marks : 1/1**

6. Which of the following operations allows you to examine the top element of a stack without removing it?

**Answer**

Peek

**Status : Correct**

**Marks : 1/1**

7. Consider the linked list implementation of a stack.

Which of the following nodes is considered as Top of the stack?

**Answer**

First node

**Status :** Correct

**Marks :** 1/1

8. What will be the output of the following code?

```
#include <stdio.h>
#define MAX_SIZE 5
void push(int* stack, int* top, int item) {
    if (*top == MAX_SIZE - 1) {
        printf("Stack Overflow\n");
        return;
    }
    stack[++(*top)] = item;
}
int pop(int* stack, int* top) {
    if (*top == -1) {
        printf("Stack Underflow\n");
        return -1;
    }
    return stack[(*top)--];
}
```

```
int main() {
    int stack[MAX_SIZE];
    int top = -1;
    push(stack, &top, 10);
    push(stack, &top, 20);
    push(stack, &top, 30);
    printf("%d\n", pop(stack, &top));
    printf("%d\n", pop(stack, &top));
    printf("%d\n", pop(stack, &top));
    printf("%d\n", pop(stack, &top));
    return 0;
}
```

}

**Answer**

302010Stack Underflow-1

**Status :** Correct

**Marks :** 1/1

9. In a stack data structure, what is the fundamental rule that is followed for performing operations?

**Answer**

Last In First Out

**Status :** Correct

**Marks :** 1/1

10. When you push an element onto a linked list-based stack, where does the new element get added?

**Answer**

At the beginning of the list

**Status :** Correct

**Marks :** 1/1

11. Elements are Added on \_\_\_\_\_ of the Stack.

**Answer**

Top

**Status :** Correct

**Marks :** 1/1

12. What is the primary advantage of using an array-based stack with a fixed size?

**Answer**

Efficient memory usage

**Status :** Correct

**Marks :** 1/1

13. In an array-based stack, which of the following operations can result in a Stack underflow?

**Answer**

Popping an element from an empty stack

**Status :** Correct

**Marks :** 1/1

14. What will be the output of the following code?

```
#include <stdio.h>
#define MAX_SIZE 5
int stack[MAX_SIZE];
int top = -1;
int isEmpty() {
    return (top == -1);
}
int isFull() {
    return (top == MAX_SIZE - 1);
}
void push(int item) {
    if (isFull())
        printf("Stack Overflow\n");
    else
        stack[++top] = item;
}
int main() {
    printf("%d\n", isEmpty());
    push(10);
    push(20);
    push(30);
    printf("%d\n", isFull());
    return 0;
}
```

**Answer**

10

**Status :** Correct

**Marks :** 1/1

15. The user performs the following operations on the stack of size 5 then at the end of the last operation, the total number of elements present in the stack is

```
push(1);  
pop();  
push(2);  
push(3);  
pop();  
push(4);  
pop();  
pop();  
push(5);
```

**Answer**

1

**Status :** Correct

**Marks :** 1/1

16. The result after evaluating the postfix expression  $10\ 5 + 60\ 6 / * 8 -$  is

**Answer**

142

**Status :** Correct

**Marks :** 1/1

17. In the linked list implementation of the stack, which of the following operations removes an element from the top?

**Answer**

Pop

**Status :** Correct

**Marks :** 1/1

18. What is the advantage of using a linked list over an array for implementing a stack?

**Answer**

Linked lists can dynamically resize

**Status :** Correct

**Marks :** 1/1

19. A user performs the following operations on stack of size 5 then which of the following is correct statement for Stack?

```
push(1);  
pop();  
push(2);  
push(3);  
pop();  
push(2);  
pop();  
pop();  
push(4);  
pop();  
pop();  
push(5);
```

**Answer**

Underflow Occurs

**Status :** Correct

**Marks :** 1/1

20. What is the value of the postfix expression 6 3 2 4 + - \*?

**Answer**

-18

**Status :** Correct

**Marks :** 1/1