

# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

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

Attempt : 1  
Total Mark : 20  
Marks Obtained : 16

#### Section 1 : MCQ

1. 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**

2. 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**

3. 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**

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

**Answer**

All of the mentioned options

**Status : Correct**

**Marks : 1/1**

5. 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**

6. 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**

7. 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

8. 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

9. 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

10. 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

11. 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 emptyStack elements: 10 20 30Stack elements: 30 20 10

**Status : Wrong**

**Marks : 0/1**

12. 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

**Status : Wrong**

**Marks : 0/1**

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

**Answer**

-18

**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. 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

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

**Answer**

71

**Status :** Wrong

**Marks :** 0/1

17. 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**

3

**Status :** Wrong

**Marks :** 0/1

18. 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

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

**Answer**

Efficient memory usage

**Status :** Correct

**Marks :** 1/1

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

**Answer**

Top

**Status :** Correct

**Marks :** 1/1



# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 3\_COD\_Question 1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 0

#### Section 1 : Coding

##### 1. Problem Statement

In a coding competition, you are assigned a task to create a program that simulates a stack using a linked list.

The program should feature a menu-driven interface for pushing an integer to stack, popping, and displaying stack elements, with robust error handling for stack underflow situations. This challenge tests your data structure skills.

##### ***Input Format***

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the integer value onto the stack. If the choice is 1, the following input is a space-separated integer, representing the element to be pushed onto

the stack.

Choice 2: Pop the integer from the stack.

Choice 3: Display the elements in the stack.

Choice 4: Exit the program.

### ***Output Format***

The output displays messages according to the choice and the status of the stack:

If the choice is 1, push the given integer to the stack and display the following:  
"Pushed element: " followed by the value pushed.

If the choice is 2, pop the integer from the stack and display the following:  
"Popped element: " followed by the value popped.

If the choice is 2, and if the stack is empty without any elements, print "Stack is empty. Cannot pop."

If the choice is 3, print the elements in the stack: "Stack elements (top to bottom): " followed by the space-separated values.

If the choice is 3, and there are no elements in the stack, print "Stack is empty".

If the choice is 4, exit the program and display the following: "Exiting program".

If any other choice is entered, print "Invalid choice".

Refer to the sample input and output for the exact format.

**Sample Test Case**

Input: 1 3

1 4

3

2

3

4

Output: Pushed element: 3

Pushed element: 4

Stack elements (top to bottom): 4 3

Popped element: 4

Stack elements (top to bottom): 3

Exiting program

**Answer**

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Divya Dharshini K  
Email: 241501053@rajalakshmi.edu.in  
Roll no: 241501053  
Phone: 9566773627  
Branch: REC  
Department: I AI & ML FA  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 3\_COD\_Question 5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 0

#### Section 1 : Coding

##### 1. Problem Statement

Milton is a diligent clerk at a school who has been assigned the task of managing class schedules. The school has various sections, and Milton needs to keep track of the class schedules for each section using a stack-based system.

He uses a program that allows him to push, pop, and display class schedules for each section. Milton's program uses a stack data structure, and each class schedule is represented as a character. Help him write a program using a linked list.

##### ***Input Format***

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the character onto the stack. If the choice is 1, the following input is a space-separated character, representing the class schedule to be pushed onto the stack.

Choice 2: Pop class schedule from the stack

Choice 3: Display the class schedules in the stack.

Choice 4: Exit the program.

### ***Output Format***

The output displays messages according to the choice and the status of the stack:

- If the choice is 1, push the given class schedule to the stack and display the following: "Adding Section: [class schedule]"
- If the choice is 2, pop the class schedule from the stack and display the following: "Removing Section: [class schedule]"
- If the choice is 2, and if the stack is empty without any class schedules, print "Stack is empty. Cannot pop."
- If the choice is 3, print the class schedules in the stack in the following: "Enrolled Sections: " followed by the class schedules separated by space.
- If the choice is 3, and there are no class schedules in the stack, print "Stack is empty"
- If the choice is 4, exit the program and display the following: "Exiting the program"
- If any other choice is entered, print "Invalid choice"

Refer to the sample output for the exact format.

### ***Sample Test Case***

Input: 1 d

1 h

3

2

3

4

Output: Adding Section: d

Adding Section: h

Enrolled Sections: h d

Removing Section: h

Enrolled Sections: d

Exiting program

**Answer**

-

**Status :** Skipped

**Marks : 0/10**