Rajalakshmi Engineering College

Name: RANJITHAM R
Email: 240801268@rajalakshmi.edu.in
Roll no: 240801268
Phone: 9514729237
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 0

Section 1 : MCQ

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

Answer

Status : Skipped ***Marks : 0/1***

2. Elements are Added on _____ of the Stack.

Answer

-

Status : - ***Marks : 0/1***

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

Answer

-

Status : *-* ***Marks : 0/1***

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

Answer

-

Status : *-* ***Marks : 0/1***

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

Answer

-

Status : *-* ***Marks : 0/1***

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

Answer

-

Status : *-* ***Marks : 0/1***

7. Consider the linked list implementation of a stack.

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

Answer

-

Status : - *Marks : 0/1*

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

Answer

-

Status : - *Marks : 0/1*

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

Answer

-

Status : - *Marks : 0/1*

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

Answer

-

Status : - *Marks : 0/1*

11. 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

-

Status : - *Marks : 0/1*

12. 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

-

Status : - ***Marks : 0/1***

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

Answer

-

Status : - ***Marks : 0/1***

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

Answer

-

Status : - ***Marks : 0/1***

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

-

Status : *-* ***Marks : 0/1***

16. 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

-

Status : *-* ***Marks : 0/1***

17. 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

-

Status : *-* ***Marks : 0/1***

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

Answer

-

Status : *-* ***Marks : 0/1***

19. 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

-

Status : -

Marks : 0/1

20. 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

-

Status : *-* ***Marks : 0/1***