

Rajalakshmi Engineering College

Name: Libin Mathew
Email: 241801137@rajalakshmi.edu.in
Roll no: 241801137
Phone: 9947119753
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 18

Section 1 : MCQ

1. 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

2. 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

3. Elements are Added on _____ of the Stack.

Answer

Top

Status : Correct

Marks : 1/1

4. 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

5. 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

6. 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

7. 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

8. 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

9. 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

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. 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

12. 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

13. 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

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

Answer

All of the mentioned options

Status : Correct

Marks : 1/1

15. 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

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

Answer

-18

Status : Correct

Marks : 1/1

17. 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

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

Answer

None of the mentioned options

Status : Wrong

Marks : 0/1

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

Answer

142

Status : Correct

Marks : 1/1

20. 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

Rajalakshmi Engineering College

Name: Libin Mathew

Email: 241801137@rajalakshmi.edu.in

Roll no: 241801137

Phone: 9947119753

Branch: REC

Department: I AI & DS AF

Batch: 2028

Degree: B.E - AI & DS

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 4

Attempt : 1

Total Mark : 10

Marks Obtained : 1

Section 1 : Coding

1. Problem Statement

You are a software developer tasked with building a module for a scientific calculator application. The primary function of this module is to convert infix mathematical expressions, which are easier for users to read and write, into postfix notation (also known as Reverse Polish Notation). Postfix notation is more straightforward for the application to evaluate because it removes the need for parentheses and operator precedence rules.

The scientific calculator needs to handle various mathematical expressions with different operators and ensure the conversion is correct. Your task is to implement this infix-to-postfix conversion algorithm using a stack-based approach.

Example

Input:

a+b

Output:

ab+

Explanation:

The postfix representation of (a+b) is ab+.

Input Format

The input is a string, representing the infix expression.

Output Format

The output displays the postfix representation of the given infix expression.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: a+(b*e)

Output: abe*+

Answer

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

```
struct Stack {
    int top;
    unsigned capacity;
    char* array;
};
```

```
struct Stack* createStack(unsigned capacity) {
    struct Stack* stack = (struct Stack*)malloc(sizeof(struct Stack));
    if (!stack)
```

```

    return NULL;

    stack->top = -1;
    stack->capacity = capacity;
    stack->array = (char*)malloc(stack->capacity * sizeof(char));

    return stack;
}

int isEmpty(struct Stack* stack) {
    return stack->top == -1;
}

char peek(struct Stack* stack) {
    return stack->array[stack->top];
}

char pop(struct Stack* stack) {
    if (!isEmpty(stack))
        return stack->array[stack->top--];
    return '$';
}

void push(struct Stack* stack, char op) {
    stack->array[++stack->top] = op;
}

// You are using GCC
int isOperand(char ch) {
    return (ch>='a' && ch<='z') || (ch>='A' && ch<='Z') || (ch>='0' && ch<='9');
}

int Prec(char ch) {
    switch(ch){
        case '+':
        case '-': return 1;
        case '*':
        case '/': return 2;
        case '^': return 3;
    }
    return -1;
}

```

```
int isRightAs(char ch){
    return ch == '^';
}
```

```
void infixToPostfix(char* exp) {
    int i,k=0;
    int len = strlen(exp);
    struct Stack* stack = createStack(len);
    char* result = (char*)malloc((len + 1)* sizeof(char));
```

```
    for(i=0;i<len;i++){
        char ch = exp[i];
```

```
        if(isOperand(ch)){
            result[k++] = ch;
        }
```

```
        else if(ch == '('){
            push(stack, ch);
        }
```

```
        else if(ch == ')'){
            while(!isEmpty(stack) && peek(stack) != '(')
                result[k++] = pop(stack);
            if(!isEmpty(stack) && peek(stack) != '(')
                pop(stack);
        }
```

```
    }
    else{
        while(!isEmpty(stack) && peek(stack)!='(' && (Prec(ch) <
Prec(peek(stack)) || (Prec(ch) == Prec(peek(stack)) && !isRightAs(ch)))){
            result[k++] = pop(stack);
        }
        push(stack,ch);
    }
}
```

```
while(!isEmpty(stack))
    result[k++] = pop(stack);
result[k] = '\0';
printf("%s\n",result);
free(result);
free(stack->array);
free(stack);
}
```

```
int main() {  
    char exp[100];  
    scanf("%s", exp);  
  
    infixToPostfix(exp);  
    return 0;  
}
```

Status : Partially correct

Marks : 1/10

Rajalakshmi Engineering College

Name: Libin Mathew
Email: 241801137@rajalakshmi.edu.in
Roll no: 241801137
Phone: 9947119753
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In an educational setting, Professor Smith tasks Computer Science students with designing an algorithm to evaluate postfix expressions efficiently, fostering problem-solving skills and understanding of stack-based computations.

The program prompts users to input a postfix expression, evaluates it, and displays the result, aiding students in honing their coding abilities.

Input Format

The input consists of the postfix mathematical expression.

The expression will contain real numbers and mathematical operators (+, -, *, /), without any space.

Output Format

The output prints the result of evaluating the given postfix expression.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 82/

Output: 4

Answer

```
// You are using GCC
```

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

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

```
#include<ctype.h>
```

```
#define MAX 100
```

```
float stack[MAX];
```

```
int top = -1;
```

```
void push(float val){
```

```
    if(top<MAX - 1){
```

```
        stack[++top] = val;
```

```
    }
```

```
}
```

```
float pop(){
```

```
    if(top >= 0){
```

```
        return stack[top--];
```

```
    }
```

```
    return 0;
```

```
}
```

```
int main(){
```

```
    char exp[MAX];
```

```
    scanf("%s",exp);
```

```
    for(int i=0;exp[i] != '\0'; i++){
```

```
        char ch = exp[i];
```

```

    if(isdigit(ch)){
        push((float)(ch - '0'));
    }else{
        float b = pop();
        float a = pop();
        float res = 0;
        switch (ch){
            case '+':
                res = a + b;
                break;
            case '-':
                res = a - b;
                break;
            case '*':
                res = a * b;
                break;
            case '/':
                res = a / b;
                break;
        }
        push(res);
    }
}

float finalres = pop();
if ((int)finalres == finalres)
    printf("%d\n", (int)finalres);
else
    printf("%f\n", finalres);
return 0;
}

```

Status : Correct

Marks : 10/10

2. Problem Statement

Siri is a computer science student who loves solving mathematical problems. She recently learned about infix and postfix expressions and was fascinated by how they can be used to evaluate mathematical expressions.

She decided to write a program to convert an infix expression with operators to its postfix form. Help Siri in writing the program.

Input Format

The input consists of a single line containing an infix expression.

Output Format

The output prints a single line containing the postfix expression equivalent to the given infix expression.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: $(2 + 3) * 4$

Output: 23+4*

Answer

-

Status : Skipped

Marks : 0/10

3. Problem Statement

Latha is taking a computer science course and has recently learned about infix and postfix expressions. She is fascinated by the idea of converting infix expressions into postfix notation. To practice this concept, she wants to implement a program that can perform the conversion for her.

Help Latha by designing a program that takes an infix expression as input and outputs its equivalent postfix notation.

Example

Input:

$(3+4)5$

Output:

34+5

Input Format

The input consists of a string, the infix expression to be converted to postfix notation.

Output Format

The output displays a string, the postfix expression equivalent of the input infix expression.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: A+B*C-D/E

Output: ABC*+DE/-

Answer

-

Status : Skipped

Marks : 0/10