

CS RAIDERS

DATA STRUCTURE

2024/2025

1 Process of inserting an element in stack is called.....

- Create
- Push
- Evaluation
- Pop

2 Process of removing an element in stack is called.....

- Create
- Push
- Evaluation
- Pop

3 in a stack, if a user tries to remove an element from an empty stack is called.....

- Underflow
- Empty collection
- Overflow
- Garbage Collection

4 Pushing an element into stack already having five elements and stack of 5, then stack becomes.....

- Underflow
- User flow
- Overflow
- Crash

5 Which of the following is not the application of stack?

- A parentheses balancing program
- Tracking of local variables at run time
- Compiler Syntax Analyzer
- Data transfer between two asynchronous process

6 Consider the usual algorithm for determining whether a sequence of parentheses is balanced. the maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: ((())((())?)

- 1
- 2
- 3
- 4 or more

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

- 1
- 40
- 74
- 18

8 Here is an infix expression: $4 + 3 * (6 * 3 - 12)$. Suppose that we are using the usual stack algorithm to 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

- 1
- 2
- 3
- 4

9 the data structure required to check whether an expression contains a balanced parentheses is?

- Stack
- Queue
- Array
- Tree

10) The postfix form of the expression $(A+B)*(C*D-E)*F/G$ is?

- AB+CD*E-FG/**
- AB+CD*E-F**G/
- AB+CD*E-*F*G/
- AB+CDE*-F*G/

11) The process of accessing data stored in a serial access memory is similar to manipulating data on a

- Heap
- Binary Tree
- Array
- Stack

12) The postfix form of $A*B+C/D$ is?

- *AB/CD+
- AB*CD/+
- A*BC+/D
- ABCD+/*

13) Which data structure is needed to convert infix notation to postfix notation

- Stack
- Queue
- Branch
- Tree

14) The prefix form of $A-B/(C*D^E)$ is ?

- /*^ACBDE
- ABCD*^DE
- A/B*C^DE
- A/BC*^DE

15) The prefix form of an infix expression $(p + q) - (r * t)$ is ?

- + pq - *rt
- +pqr*t
- +pq*rt
- +*pqrt

16) Which data structure is used for implementing recursion?

- List
- Queue
- Array
- Stack

17) The result of evaluating the postfix expression
 $5, 4, 6, +, *, 4, 9, 3, /, +, *$ is?

- 600
- 350
- 650
- 588

18) Convert the following infix expressions into its equivalent
postfix expressions. $(A+B^D)/(E-F)+G$

- $(ABD^+EF-/G+)$
- $(ABD+^EF-/G+)$
- $(ABD^+EF/-G+)$
- $(ABDEF+^/-G+)$

19) Convert the following infix expression to postfix form using a
stack, $x+y*z+(p*q+r)*s$, Follow usual precedence rule and
assume that the expression is legal.

- $xyz^*+pq*r+s^*$
- $xyz^*+pq*r+s+*$
- $xyz+*pq*r+s^*$
- $xyzp+**qr+s^*$

20 which of the following is not an inherent application of stack

- Reversing a string
- Evaluation of postfix expression
- implementation of recursion
- job scheduling

21 Assume that the operator +, -, x are left associative and ^ is right associative. the order of precedence (from highest to lowest) is ^, x, +, - . The postfix expression for the infix expression $a+b\times c-d^f$ is?

- infix Expression
- prefix Expression
- postfix Expression
- Stack

22 The result of evaluating the postfix expression
5, 4, 6, +, *, 4, 9, 3, /, +, * is?

- abcx+def^^-
- abcx+de^f^-
- ab+cxd-e^f^
- +axbc^^def

23 if the element “A”, “B”, “C” and “D” are placed in a stack and are deleted one at a time, what is the order of removal?

- ABCD
- DCBA
- DCAB
- ABDC

24 Suppose a stack is to be implemented with a linked list instead of an array. what would be effect on the time complexity of the push and pop operations of the stack implemented using linked list (Assuming stack is implemented efficiently)?

- O(1) for insertion and O(n) for deletion
- O(1) for insertion and O(1) for deletion
- O(n) for insertion and O(1) for deletion
- O(n) for insertion and O(n) for deletion

25 The following postfix expression with single digit operands is evaluated using a stack. $823^/23*+51*-$

Note that $^$ is the exponentiation operator, the top two elements of the stack after the first $*$ is evaluated are:

- 6, 1
- 5, 7
- 3, 2
- 1, 5

26 To evaluate an expression without any embedded function calls:

- As many stacks as the height of the expression tree are needed
- one stack is enough
- two stacks are needed
- A turing machine is needed in the general case

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

- 284
- 213
- 142
- 71

- 28 Stack A has the entries a, b, c (with a on top). Stack B is empty. An entry popped out of stack A can be printed immediately or pushed to stack B, An entry popped out of the stack B can be only be printed. in this arrangement, which of the following permutations of a, b, c are not possible?
- b a c
 - b c a
 - c a b
 - a b c
- 29 Consider the following operations performed on a stack of size 5: Push(a); Pop(c); Pop(); Push(d); Pop(); Pop(); Push(e) Which of the following statements is correct?
- Underflow occurs
 - Stack operations are performed smoothly
 - Overflow occurs
 - None of the above
- 30 if the sequence of operations - push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2), pop are performed on a stack, the sequence of popped out values
- 2, 2, 1, 1, 2
 - 2, 2, 1, 2, 2
 - 2, 1, 2, 2, 1
 - 2, 1, 2, 2, 2

31 Consider the following operation performed on a stack of size 5.
Push(1); Pop(); Push(2); Push(3);
Pop(); Push(4); Pop(); Pop(); Push(5);
After the completion of all operation, the number of elements present in stack is?

- 1
- 2
- 3
- 4

32 declare a stack of characters

while (there are more characters in the word to read)

{

 read a character

 push the character on the stack

}

while (the stack is not empty)

{

 pop a character off the stack

 write the character to the screen

}

what is output for input “geeksquiz”?

- geeksquizgeeksquiz
- ziuqskeeg
- geeksquiz
- ziuqskeegziuqskeeg

33) Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

```
void fun(int n)
{
    Stack S;
    while (n > 0)
    {
        Push(&s, n%2);
        n = n / 2;
    }
    while (! isEmpty(&s))
        printf ("%d", pop(&S));
}
```

What does the above function do in general?

- prints binary representation of n in reverse order
- prints binary representation of n
- prints the value of logn
- prints the value of login in reverse order

- 34) Following is an incorrect pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while (more input is available)
{
    read a character
    if (the character is a \'C\')
        push it on the stack
    else if (the character is a \')\' and (the stack is not empty )
        pop a character off the stack
    else
        print "unbalanced" and exit
}
print "balanced"
```

which of these unbalanced sequences does the above code think is balanced?

- ((()
- ()())()
- ((())()
- (())()

35 # include

```
# define EOF -1
void push (int); /* push the argument on the stack */
int pop (void); /* pop the top of the stack */
void flagError ();
int main()
{
    int c, m, n, r;
    while ((c = getchar () ) != EOF)
    {
        if (isdigit(c))
            push (c);
        else if ((c == '+') || (c == '*'))
        {
            m = pop();
            n = pop();
            r = (c == '+') ? n + m : n*m;
            push(r);
        }
        else if (c != ' ')
            flagError ();
    }
    printf("%c", pop());
}
```

- 15
- 25
- 30
- 150

Stack

- 1 Push 2 Pop 3 Under flow 4 Over flow 5 Data Transfer
- 6 3 7 -18 8 4 9 Stack 10 AB+CD*E-*F*G/
- 11 Stack 12 AB*CD/+ 13 Stack 14 -A/B*C^DE 15 -+pq*rt
- 16 Stack 17 350 18 (ABD^+EF-/G+) 19 xyz*+pq*r+s*+
- 20 job scheduling 21 postfix 22 abcx+def^^- 23 DCBA
- 24 O(1) for insertion and O(1) for deletion 25 6,1 26 one stack
- 27 142 28 c a b 29 Stack operation are performed smoothly
- 30 2,2,1,1,2 31 1 32 ziufskeeg
- 33 prints binary representation of n 34 ((())) 35 25