ALGO

Tutorial 01

1. What is a stack.
2. Define Push, pop, peek, is empty, Size in Stack.
3. Give 7 examples of stack found in real life.
4. How to find out that stack is empty in a program?
5. What is Queue?
6. Give example of Queue in real life.
7. Draw the logical representation of Queue?
8. There are three instances which Queue could be empty.

Mention those?

Answer

01)

Stack is a Lear data structure that follows’ the LIFO (Last in First Out) principal. This deference as a container in Which insertion and deletion can be done form that are that one and known as the tip of the stack

Push POP



|  |
| --- |
|  |
|  |
|  |
|  |



EX- Pack of cards

02) PUSH = removes an on element from the stack

POP = adds an element in the stack.

Peek = get the top elements without removing it.

03) 1. Pack of cards

2. Women’s Bangles

3. Books and Clothes

4. Floors in a Building

5.Browsers

6.Mobile phone call logs

7. Companies

04) It is use to check if a stack is empty or not This method requires no parameters. It returns true if the stack is empty and false if the stack is not empty.

EX- 3 4 + = \* 7 /



|  |
| --- |
| 7 |
| 4 |
| 3 |

Put in middle.



3 + 4 = 7



1st step

|  |
| --- |
| \* |
| 2 |
| 7 |

7 \* 2 = 14

2nd step

|  |
| --- |
| / |
| 7 |
| 14 |

14 / 7 = 2

3rd step

|  |
| --- |
|  |
| 2 |

Answer

Answer is = 2

Ex – 2 3 1 \* + 9 -

|  |
| --- |
| \* |
| 1 |
| 3 |
| 2 |

2 \* 3 = 6

|  |
| --- |
| + |
| 1 |
| 6 |

6 + 1 = 7

|  |
| --- |
| - |
| 9 |
| 7 |

7 - 9

|  |
| --- |
|  |
|  |

* 2

Answer = - 2

05)

DFS problem



D

B

E

C

A

|  |
| --- |
|  |
| A |

1.



DFS

|  |  |
| --- | --- |
| A |  |

2.



|  |
| --- |
|  |
| C |
| B |



|  |  |  |
| --- | --- | --- |
| A | B |  |



3.



|  |
| --- |
|  |
| D |
| C |

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | D |  |

4.

|  |
| --- |
|  |
| E |
| C |



|  |
| --- |
|  |
| E |
| C |

5.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | E |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | D | E | C |

6.

Ex - 5 8 7 + - \*



|  |
| --- |
|  |
| 7 |
| 8 |
| 6 |

|  |
| --- |
|  |
| 15 |
| 5 |

8+7

2nd step 5 – 15 - 10

1st step

|  |
| --- |
| \* |
| -10 |

3rd step