

PROGRAMMING PRACTICES

ASSIGNMENT -2

1. For an $n \times n$ matrix diagonal elements are given. All non-diagonal elements are equal, say x . Find out the minimum value of x such that the sum of diagonal elements is less than the sum of non diagonal elements.

Input:

4
5 10 4 7

Output:

5 3 3 3
3 10 3 3
3 3 4 3
3 3 3 7

(Hint: Sum of diagonal element: 26, sum of other elements: 36)

2. In a matrix, calculate the sum of all neighbouring elements of each diagonal element (up, down, left, right, and 4 diagonal elements – a total of 8 elements). Print these values corresponding to each diagonal element. Also print the index of that diagonal element whose corresponding sum is highest.

Input:

5
3 2 0 4 5
1 10 4 -2 6
0 3 7 0 8
6 5 1 4 4
9 7 0 -1 3

Output:

3: 13
10: 20
7: 25
4: 22
3: 7
index: 3

3. Take a matrix of size $m \times n$. Find out its transpose using a function Transpose(arr, &m, &n).

Input:

2 3
1 1 1
2 2 2

Output:

1 2
1 2
1 2

4. Read a square matrix of size n. Do the following tasks:
- a. Print all unique values along each row.
 - b. Print all unique values along each column.
 - c. Print all unique values in the matrix.

Input:

4
5 1 2 1
4 10 3 5
2 1 4 4
1 2 0 7

Output:

Along rows:

1 2 5
3 4 5 10
1 2 4
0 1 2 7

Along columns:

1 2 4 5
1 2 10
0 2 3 4
14 5 7

Matrix:

0 1 2 3 4 5 7 10

5. Create an array of employee names.
- a. Arrange them in lexicographically sorted order
 - b. Print all unique names

Input:

8
Ram Mohan Shyam Amit Kritika Ram Mohit Amit

Output:

Sorted:

Amit Amit Kritika Mohan Mohit Ram Ram Shyam

Unique:

Amit Kritika Mohan Mohit Ram Shyam

6. Enter a square matrix and print the i th row and j th column whose sums are equal.

Input:

4

1 1 1 5
1 1 1 0
2 2 2 1
3 3 3 2

Output:

Row: 1

Column: 4

7. Check whether given strings are palindrome.

Input:

4

sos

abc

hello

abba

Output:

yes

no

no

yes

8. For an $n \times n$ matrix diagonal elements are given. Find if the matrix is such that the diagonal element is equal to the sum of its neighboring (up/down/right/left only) elements.

Input:

5

3 2 0 4 5

1 10 4 -2 6

0 3 7 -1 8

6 5 1 4 5

9 7 2 -1 4

Output:

Yes

9. You are given a list of $n-1$ integers and these integers are in the range of 1 to n . There are no duplicates in the list. One of the integers is missing in the list. Write an efficient code to find the missing integer.

Input :

7

1 2 4 6 3 7 8

Output :

5

10. There are 2 sorted arrays A and B of size n each. Write an algorithm to find the median of the array obtained after merging the above 2 arrays (i.e. array of length 2n).

Input :

5
1 12 15 26 38
5
2 13 17 30 45

Output :

16

11. Implement a queue without using any library.

Operations:

- 1 - enqueue()
- 2 - dequeue()-returns popped number.(return -1 if queue is empty)
- 3 - front() - returns number at front. (return -1 if queue is empty)

Input:

1 (Number of test cases)
6 (Number of operations to be performed)
1 10 (enqueue(10))
1 20 (enqueue(20))
2 (dequeue())
3 (front())
2 (dequeue())
3 (front())

Output:

10
20
20
-1

12. Implement queue using stack.(Without using any library.)

Operations:

- 1 - enqueue()
- 2 - dequeue()-returns popped number.(return -1 if queue is empty)
- 3 - front() - returns number at front. (return -1 if queue is empty)

Input:

1
6
1 10
1 20
2
3
2
3

Output:

10
20
20
-1

13. Implement Queue using linked list.

Operations:

- 1 - enqueue()
- 2 - dequeue()-returns popped number.(return -1 if queue is empty)
- 3 - front() - returns number at front. (return -1 if queue is empty)

Input:

1
6
1 10
1 20
2
3
2
3

Output:

10
20
20
-1

14. Implement a circular queue. (In case of overflow/underflow condition print accordingly.)

Operations:

- 1 - enqueue()
- 2 - dequeue()-returns popped number.
- 3 - front() - returns number at front.

Input:

1
5 (array size to implement circular queue)
6 (number of operations)
1 10
1 20
2
3
2
3

Output:

10
20
20
Underflow

15. Implement deque.(Without using any library).

Operations:

- 1 : push_front(X)
- 2 : push_back(X)
- 3 : pop_front() (returns popped number)
- 4 : pop_back()
- 5 : front() (return front number)
- 6 : back() (return number at back)

Input:

1
6
1 10
1 20
5
6
3
4

Output:

20
10
20
10

16. Given an integer K and a queue of integers, we need to reverse the order of the first K elements of the queue, leaving the other elements in the same relative order.

Only following standard operations are allowed on the queue.

- enqueue(x) : Add an item x to rear of queue
- dequeue() : Remove an item from front of queue
- size() : Returns number of elements in the queue.
- front() : Finds front item.

Input:

5 3
1 2 3 4 5

Output:

3 2 1 4 5

17. Write a programme for reversing a queue Q. Only following standard operations are allowed on the queue.

- a) enqueue()
- b) dequeue()
- c) empty()

Input:

[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Output:

[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

18. Given a number N. The task is to generate and print all binary numbers with decimal values from 1 to N.

Input:

2

2

5

Output:

1 10

1 10 11 100 101

19. Given an array and a positive integer k, find the first negative integer for each and every window(contiguous subarray) of size k.

Input:

2

5

-8 2 3 -6 10

2

8

12 -1 -7 8 -15 30 16 28

3

Output:

-8 0 -6 -6

-1 -1 -7 -15 -15 0

20. Given an array A and an integer K. Find the maximum for each and every contiguous subarray of size K.

Input :

2

9 3

1 2 3 1 4 5 2 3 6

10 4

8 5 10 7 9 4 15 12 90 13

Output :

3 3 4 5 5 5 6

10 10 10 15 15 90 90

21. Implement Stack.

Input :

push(2)

push(3)

pop()

push(4)

pop()

Output :

3, 4

22. Implement a stack using queue.

Input :

push(2)

push(3)

pop()

push(4)

pop()

Output :

3, 4

23. Implement a stack using linked list.

Input :

push(2)

push(3)

pop()

push(4)

pop()

Output :

3, 4

24. Given a stack of integers of size N , your task is to complete the function pairWiseConsecutive(), that checks whether numbers in the stack are pairwise consecutive or not. The pairs can be increasing or decreasing, and if the stack has an odd number of elements, the element at the top is left out of a pair. The function should retain the original stack content.

Only following standard operations are allowed on stack.

- push(X): Enter an element X on top of stack.
- pop(): Removes top element of the stack.
- empty(): To check if the stack is empty.

Input :

2

6

1 2 3 4 5 6

5

1 5 3 9 7

Output :

Yes

No

25. You are given an array A of size N . You need to first push the elements of the array into a stack and then print minimum in the stack at each pop.

Input :

5

1 2 3 4 5

Output :

1 1 1 1 1

26. Given a string *s* containing just the characters '(', ')', '{', '}', '[', and ']', determine if the input string is valid.(Solve using stack).

Input :

3

()

([{}])

(D]

Output :

Valid

Valid

Invalid

27. Given a postfix expression, the task is to evaluate the expression and print the final value. Operators will only include the basic arithmetic operators like *,/,+ and -

Input :

2

231*+9-

123+*8-

Output :

-4

-3

28. An string of words is given, the task is to reverse the string using stack.

Input :

2

ABCD

Hello World

Output :

DCBA

dlroW olleH

29. Compress the string when lower and upper case are same. In compressed string characters should be in lowercase.

Input :

3

aabbB

abc
Aaacca

Output :

2a3b
1a1b1c
3a2c1a

30. Given a stack, the task is to sort it such that the top of the stack has the greatest element.(Using stack operations only.)

Input:

Stack: 3 2 1

Output: 3 2 1

Note: Do not use inbuilt libraries.