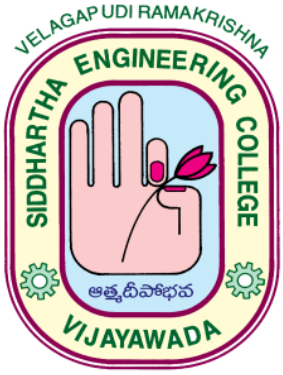
VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE
(AUTONOMOUS)
(AFFILIATED TO JNTUK, KAKINADA)
KANURU, VIJAYAWADA – 520 007
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

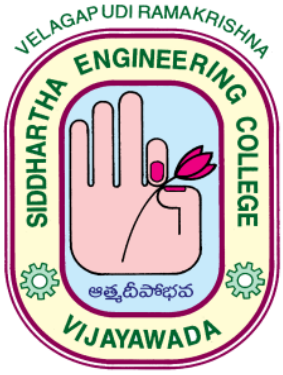


17CS3554 - COMPETITIVE CODING LAB

(LAB WORKBOOK)

Registration Number :
Student Name :
Year / Semester :
Section :
Course Instructor :

VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE
(AUTONOMOUS)
(AFFILIATED TO JNTUK, KAKINADA)
KANURU, VIJAYAWADA – 520 007
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



17CS3554 - COMPETITIVE CODING LAB

(LAB WORKBOOK)

B. Tech (Computer Science and Engineering)

III Year – V Semester

Academic Year: 2021-2022

TABLE OF CONTENTS

LAB CONTINUOUS EVALUATION REPORT

Session. No	Date	Experiment Name	Pre-Lab (5M)	In-Lab				Post Lab (5M)	Viva Voce (5M)	Total (50M)	Faculty Signature
				Logic (10M)	Execution (10M)	Result (10M)	Analysis (5M)				
1.		Stacks and Queues									
2.		Linked Lists and Hashing Techniques									
3.		Pointers									
4.		Dynamic Programming									
5.		Set, List, Map									
6.		Binary Trees and Binary Search Trees									
7.		Open Problems from Coding Contests									
8.		Open Problems from Coding Contests									
9.		Case Study-1									
10.		Case Study-2									
11.		Case Study-3									

No. of Experiments Recorded:

Faculty Signature

Registration No.: **Academic Year: 2021-2022**

Lab Session 01: Stacks and Queues

Date of the Session: **Time of the Session:**

PRE-LAB:

1. What is the difference between an Array and Stack?

2. What is the difference between linear and non-linear data structures?

3. Stacks and Queues are linear data structure or non-linear data structure?

4. Are linked lists considered linear or non-linear data structures?

5. What are the different types of linked list?

6. How to implement a stack using queue?

IN-LAB:

1. **Given a binary pattern that contains '?' wildcard character at few positions, find all possible combinations of binary strings that can be formed by replacing the wildcard character by either 0 or 1. The idea is to process each character of the pattern one by one and recur for the remaining pattern.**
   For example, for wildcard pattern 1?11?00?1? The possible combinations are
   1011000010
   1011000011
   1011000110
   1011000111
   1011100010
   1011100011
   1011100110
   1011100111
   1111000010
   1111000011
   1111000110
   1111000111
   1111100010
   1111100011
   1111100110
   1111100111

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. **You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost cylinder any number of times.**
   Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means you must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then print the height. The removals must be performed in such a way as to maximize the height.
   Note: An empty stack is still a stack.
   Input Format:
   - The first line contains three space-separated integers, n1, n2, and n3, describing the respective number of cylinders in stacks 1, 2, and 3. The subsequent lines describe the respective heights of each cylinder in a stack from top to bottom:
   - The second line contains n1 space-separated integers describing the cylinder heights in stack 1. The first element is the top of the stack.
   - The third line contains n2 space-separated integers describing the cylinder heights in stack 2. The first element is the top of the stack.
   - The fourth line contains n3 space-separated integers describing the cylinder heights in stack 3. The first element is the top of the stack.

   Output Format:
   Print a single integer denoting the maximum height at which all stacks will be of equal height.
   Sample Input:
   5 3 4
   3 2 1 1 1
   4 3 2
   1 1 4 1
   Sample Output:
   5

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

3. **Given a 2d grid map of '1's (land) and '0's (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.**

Example 1:

Input:

11110
11010
11000
00000

Output:

1

Example 2:

Input:

11000
11000
00100
00011

Output:

3

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. **Given a stack, sort it using recursion. Use of any loop constructs like while, for etc is not allowed. We can only use the following ADT functions on Stack S:**

   is_empty(S) : Tests whether stack is empty or not
   push(S) : Adds new element to the stack
   pop(S) : Removes top element from the stack
   top(S) : Returns value of the top element. Note that this function does not remove element from the stack

   Input: -3 <--- Top

   14
   18
   -5
   30

   Output: 30 <--- Top

   18
   14
   -3
   -5

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. **You have an empty sequence, and you will be given N queries. Each query is one of these three types:**
   1 x -Push the element x into the stack.
   2 -Delete the element present at the top of the stack.
   3 -Print the maximum element in the stack.

Input Format
The first line of input contains an integer N. The next N lines each contain an above-mentioned query. *(It is guaranteed that each query is valid.)*

Constraints
$1 \leq N \leq 10^5$
$1 \leq x \leq 10^9$
$1 \leq type \leq 3$

Output Format
For each type 3 query, print the maximum element in the stack on a new line.

Sample Input
10
1 97
2
1 20
2
1 26
1 20
2
3
1 91
3

Sample Output
26
91

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 02: Linked List and Hashing Techniques

Date of the Session: **Time of the Session:**

PRE-LAB:

Linked Lists:

1. List the differences between Linear Array and Linked List?

2. How can someone insert a node in a random location of the Linked List?

3. Where will be the free node available while inserting a new node in a linked list?

4. How can someone display singly linked list from first to last?

5. State the steps to insert data at the starting of a singly linked list?

6. Mention how to display Singly Linked List from First to Last?

Hashing:

7. What is a hash table and hash function?

8. If several elements are competing for the same bucket in the hash table, what is it called?

9. What is direct addressing and mention its search complexity?

IN-LAB:

1. **Find if an array is a subset of another array with O(n) complexity.**
   For example,
   arr1 = {1,2,3,4,5}
   arr2 = {3,4,5}
   arr2 is a subset of arr1.

   arr3 = {1,2,3,4,5}
   arr4 = {1,2,9}
   arr4 is not a subset of arr3.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. **Given a singly linked list of size N containing only English Alphabets. Your task is to complete the function arrangeC&V(), that arranges the consonants and vowel nodes of the list it in such a way that all the vowels nodes come before the consonants while maintaining the order of their arrival.**

   Input:

   The function takes a single argument as input, the reference pointer to the head of the linked list. There will be T test cases and for each test case the function will be called separately.

   Output:

   For each test case output a single line containing space separated elements of the list.

   User Task:

   The task is to complete the function arrange() which should arrange the vowels and consonants as required.

   Constraints:

   $1 \leq T \leq 100$

   $1 \leq N \leq 100$

   Example:

   Input:

   2

   6

   a e g h i m

   3

   q r t

   Output:

   a e i g h m

   q r t

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

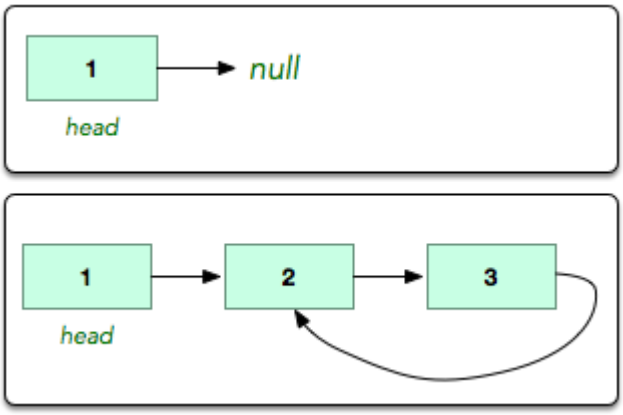
3. **A linked list is said to contain a cycle if any node is visited more than once while traversing the list. Complete the function provided for you in your editor. It has one parameter: a pointer to a Node object named head that points to the head of a linked list. Your function must return a boolean denoting whether or not there is a cycle in the list. If there is a cycle, return true; otherwise, return false.**
Note: If the list is empty, head will be null.

Input Format:
Our hidden code checker passes the appropriate argument to your function. You are not responsible for reading any input from stdin.

Constraints:
$1 \leq listsize \leq 100$

Output Format:
If the list contains a cycle, your function must return true. If the list does not contain a cycle, it must return false. The binary integer corresponding to the boolean value returned by your function is printed to stdout by our hidden code checker.

Sample Input:
The following linked lists are passed as arguments to your function:



Sample Output:
0
1

Explanation:
The first list has no cycle, so we return false and the hidden code checker prints 0 to stdout.
The second list has a cycle, so we return true and the hidden code checker prints 1 to stdout.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. **Given a string S and a string T, find the minimum window in S which will contain all the characters in T in linear time complexity.**
   Note that when the count of a character C in T is N, then the count of C in minimum window in S should be at least N.
   Example 1:
   Input:
   string = “this is a test string”, pattern = “tist”
   Output:
   Minimum window is “t stri”
   Example 2:
   S = "ADOBECODEBANC"
   T = "ABC"
   Minimum window is "BANC"

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. **A linked list is said to contain a cycle if any node is visited more than once while traversing the list.**
   The function has one parameter: a pointer to a Node object named that points to the head of a linked list. Your function must return a boolean denoting whether there is a cycle in the list. If there is a cycle, return true; otherwise, return false.
   Note: If the list is empty, will be null.
   Input Format:
   Our hidden code checker passes the appropriate argument to your function. You are not responsible for reading any input from stdin.
   Output Format:
   If the list contains a cycle, your function must return true. If the list does not contain a cycle, it must return false. The binary integer corresponding to the boolean value returned by your function is printed to stdout by our hidden code checker.

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 03: Pointers

Date of the Session: **Time of the Session:**

PRE-LAB:

1. **What is indirection?**

2. **How many levels of pointers can you have?**

3. **Is it better to use malloc() or calloc()?**

4. **How do you declare an array that will hold more than 64KB of data?**

5. **What is the difference between far and near?**

6. **What happens if you free a pointer twice?**

IN-LAB:

1. A pointer in C is a way to share a memory address among different contexts (primarily functions). They are primarily used whenever a function needs to modify the content of a variable, of which it doesn't have ownership.

   In order to access the memory address of a variable,val , we need to prepend it with & sign. E.g., &val returns the memory address of val.

   This memory address is assigned to a pointer and can be shared among various functions. E.g. will assign the memory address of to pointer . To access the content of the memory to which the pointer points, prepend it with a *. For example, *p will return the value reflected by and any modification to it will be reflected at the source ().

```
void increment(int *v)
{
   (*v)++;
}

int main()
{
   int a;
   scanf("%d", &a);
   increment(&a);
   printf("%d", a);
   return 0;
}
```

   You have to complete the function void update(int *a,int *b), which reads two integers as argument, and sets with the sum of them, and with the absolute difference of them.

   Input Format
   Input will contain two integers, a and b, separated by a newline.
   Output Format
   You have to print the updated value of a and b, on two different lines.

   Sample Input
   4
   5

Sample Output

9
1

Explanation

a'=4+5=9
b'=|4-5| = 1

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Given a string, find the length of the longest substring without repeating characters.
   Input:
   First line should contain no. of test cases
   Next ‘n’ lines should contain strings
   Output:
   length of the longest substring

   Input:
   abcabcbb
   bbbbb
   pwwkew

   Output:
   3
   1
   3

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

3. Given a pointer to a variable, the task is to complete the function updateVar() which will increment the value of the variable by 10. The function does not returns anything.

   Input:

   The first line of input contains a single integer T, which denotes the number of testcases. Then T test cases follows. Each test case consists of a single line which contains a single integer A.

   Output:

   Corresponding to each test case, print in a new line the updated value of A.

   User Task:

   Since this is a functional problem you don't have to worry about input, you just have to complete the function updateVar().

   Constraints:

   1 <= T <= 100

   1 <= A <= 103

   Example:

   Input:

   4

   2

   5

   9

   7

   Output:

   12

   15

   19

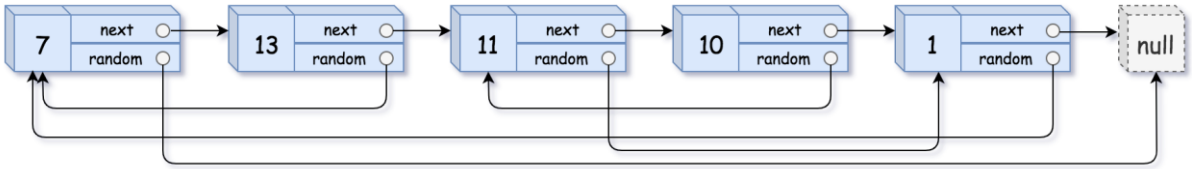
   17

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

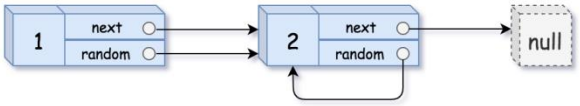
1. A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null.
   Return a deep copy of the list.
   The Linked List is represented in the input/output as a list of n nodes. Each node is represented as a pair of [val, random_index] where:
   - val: an integer representing Node.val
   - random_index: the index of the node (range from 0 to n-1) where random pointer points to, or null if it does not point to any node.

   Example 1:



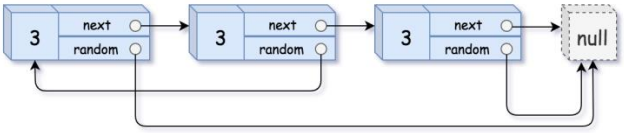
   Input: head = [[7,null],[13,0],[11,4],[10,2],[1,0]]
   Output: [[7,null],[13,0],[11,4],[10,2],[1,0]]

   Example 2:



   Input: head = [[1,1],[2,1]]
   Output: [[1,1],[2,1]]

   Example 3:



   Input: head = [[3,null],[3,0],[3,null]]
   Output: [[3,null],[3,0],[3,null]]

   Example 4:
   Input: head = []
   Output: []
   Explanation: Given linked list is empty (null pointer), so return null.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Given a linked list, rotate the list to the right by k places, where k is non-negative.
   Example:
   Input: 1->2->3->4->5->NULL, k = 2
   Output: 4->5->1->2->3->NULL

   Example 2:
   Input: 0->1->2->NULL, k = 4
   Output: 2->0->1->NULL

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 04: Dynamic Programming

Date of the Session: **Time of the Session:**

PRE-LAB:

1. Compare Dynamic programming Vs divide and conquer VS Greedy Method.

2. What is Dynamic Programming?

3. Mention the steps to solve DP.

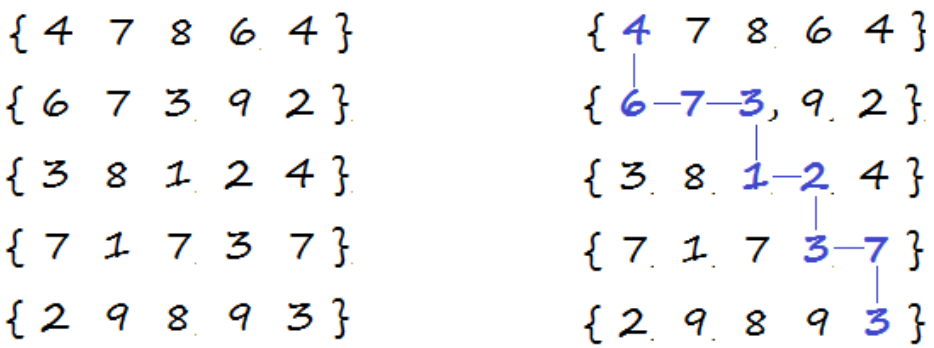
4. What is optimal substructure?

5. What do you mean by overlapping subproblems?

6. Differentiate Tabulation and Memoization.

IN-LAB:

1. Given a MxN matrix where each cell has a cost associated with it, find the minimum cost to reach last cell (M-1,N-1) of the matrix from its cell (0,0). We can only move one unit right or one unit down from any cell i. e from cell (i, j),we can move to (i,J+1) or (i+1,J)



The highlighted path shows the minimum cost path having cost of 36
The idea is to use recursion, the problem has an optimal substructure. That means the problem can be broken down into smaller, simple sub problems, which can further be divided into yet simple, small sub problems until the solution becomes trivial. The problem can be recursively defined as:
Cost to reach cell (m,n) = Cost[m][n] + min (cost to reach cell (m,n-1), cost to reach cell(m,n-1))

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Given a rod of length n inches and an array of prices that contains prices of all pieces of size smaller than n. Determine the maximum value obtainable by cutting up the rod and selling the pieces.

Example
Input:

Length	1	2	3	4	5	6	7	8
Price	1	5	8	9	10	17	17	20

Output
Maximum Obtainable Value is 22

Input

Length	1	2	3	4	5	6	7	8
Price	3	5	8	9	10	17	17	20

Output
Maximum Obtainable Value is 24

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

3. Given a m x n grid filled with non-negative numbers, find a path from top left to bottom right which minimizes the sum of all numbers along its path.
   Note: You can only move either down or right at any point in time.
   Example:
   Input:
   [
   [1,3,1],
   [1,5,1],
   [4,2,1]
   ]
   Output: 7
   Explanation: Because the path 1→3→1→1→1 minimizes the sum.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. George decided to make some money doing business on the Internet for exactly *n* days. He knows that on the *i*-th day ($1 \leq i \leq n$) he makes a_i money. He loves progress, that is why he wants to know the length of the maximum non-decreasing subsegment in sequence a_i. Let us remind you that the subsegment of the sequence is its continuous fragment. A subsegment of numbers is called non-decreasing if all numbers in it follow in the non-decreasing order. Help George cope with this task!

   Input
   The first line contains a single positive integer *n,* number of days
   The second line contains *n* integers, a_1, a_2, ……., a_n

   Output
   Print a single integer — the length of the maximum non-decreasing subsegment of sequence a.

   Sample Inputs
   6
   2 2 1 3 4 1

   3
   2 2 9

   Sample Outputs
   3

   3

   Note
   In the first test the maximum non-decreasing subsegment is the numbers from the third to the fifth one.
   In the second test the maximum non-decreasing subsegment is the numbers from the first to the third one.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Given a m * n matrix mat and an integer K, return a matrix answer where each answer[i][j] is the sum of all elements mat[r][c] for i - K ≤ r ≤ i + K, j - K ≤ c ≤ j + K, and (r, c) is a valid position in the matrix.

   Example 1:
   Input: mat = [[1,2,3],[4,5,6],[7,8,9]], K = 1
   Output: [[12,21,16],[27,45,33],[24,39,28]]

   Example 2:
   Input: mat = [[1,2,3],[4,5,6],[7,8,9]], K = 2
   Output: [[45,45,45],[45,45,45],[45,45,45]]

   Constraints:
   M == mat.length
   n == mat[i].length
   1 ≤m, n, K≤ 100
   1 ≤mat[i][j] ≤100

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 05: List, Set, Map

Date of the Session: **Time of the Session:**

PRE-LAB:

1. List down the primary interfaces provided by Java Collections Framework.

2. Differentiate between List and Set.

3. What is LinkedHashSet in Java Collections Framework?

4. Can you add a null element into a TreeSet or HashSet?

5. What is the ConcurrentHashMap in Java and do you implement it?

6. Differentiate between PriorityQueue and TreeSet

IN-LAB:

1. Steve has a string of lowercase characters in range ascii[‘a’..’z’]. He wants to reduce the string to its shortest length by doing a series of operations. In each operation he selects a pair of adjacent lowercase letters that match, and he deletes them. For instance, the string aab could be shortened to b in one operation.
   Steve’s task is to delete as many characters as possible using this method and print the resulting string. If the final string is empty, print Empty String

   Input Format:
   A single string, *s*

   Constraints:
   $1 \leq |s| \leq 100$

   Output Format:
   If the final string is empty, print Empty String; otherwise, print the final non-reducible string.

   Sample Input 0 **:** aaabccddd
   Sample Output 0 **:** abd
   Explanation 0:
   Steve performs the following sequence of operations to get the final string:
   aaabccddd → abccddd → abddd → abd

   Sample Input 1 **:** aa
   Sample Output 1 **:** Empty String
   Explanation 1:
   aa → Empty String

   Sample Input 2 **:** baab
   Sample Output 2 **:** Empty String
   Explanation 2:
   baab → bb → Empty String

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Adam has a string of lowercase letters that he wants to copy to a new string. He can perform the following operations with the given costs. He can perform them any number of times to construct a new string ***p:***
    - Append a character to the end of string ***p*** at a cost of 1 Dollar
    - Choose any substring of ***p*** and append it to the end of ***p*** at no charge

   Given ***n*** strings ***s[i]*** find and print the minimum cost of copying each ***s[i]*** to ***p[i]*** on a new line
   For example, given a string ***s=abcabc,*** it can be copied for ***3*** dollars. Start by copying ***a,b,*** and ***c*** individually at a cost of ***1*** dollar per character. String ***p=abc*** at this time. Copy ***p[0:2]*** to the end of ***p*** at no cost to complete the copy.

   Input Format
   The first line contains a single integer ***n***, the number of strings
   Each of the next ***n*** lines contains a single string, ***s[i]***

   Constraints
   $1 \leq n \leq 5$
   $1 \leq |s[i]| \leq 10^5$

   Subtasks
   $1 \leq |s[i]| \leq 10^3$ for 45% of the maximum score

   Output Format
   For each string ***s[i]*** print the minimum cost of constructing a new string ***p[i]*** on a new line

   Sample Input
   2
   abcd
   abab
   Sample output
   4
   2

   Explanation
   Query 0: We start with ***s="abcd"*** and ***p=""***
   1. Append character **'a'** to ***p*** at a cost of ***1*** dollar, ***p="a"***
   2. Append character **'b'** to ***p*** at a cost of ***1*** dollar, ***p="ab"***
   3. Append character **'c'** to ***p*** at a cost of ***1*** dollar, ***p="abc"***
   4. Append character **'d'** to ***p*** at a cost of ***1*** dollar, ***p="abcd"***

   Because the total cost of all operations is 1+1+1+1=4 dollars, we print ***4*** on a new line
   Query 1: We start with ***s="abab"*** and ***p=""***
   1. Append character **'a'** to ***p*** at a cost of ***1*** dollar, ***p="a"***
   2. Append character **'b'** to ***p*** at a cost of ***1*** dollar, ***p="a"***
   3. Append substring **"ab"** to ***p*** at no cost, ***p="abab"***

   Because the total cost of all operations is **1+1=2** dollars, we print **2** on a new line

   Note
   A substring of a string **S** is another string **S'** that occurs "in" **S.** For example, the substrings of the string **"abc"** are **"a", "b", "c", "ab", "bc",** and **"abc"**

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

3. We define the distance between two array values as the number of indices between the two values. Given ***a,*** find the minimum distance any pair of equal elements in the array. If no such value exists, print **-1.**
For example, if ***a=[3,2,1,2,3],*** there are two matching pairs of values: **3** and **2.** The indices of the **3**'s are ***i=0*** and ***j=4,*** so their distance is ***d[i,j]=|j-i|=4.*** The indices of the **2's** are ***i=1*** and ***j=3,*** so their distance is ***d[i,j]=|j-i|=2.***

Input Format
The fist line contains an integer ***n,*** the size of array ***a.***
The second line contains ***n*** space-separated integers ***a[i].***

Constraints
$1 \leq n \leq 10^3$
$1 \leq a[i] \leq 10^5$

Output Format
Print a single integer denoting the minimum ***d[i,j]*** in ***a.*** If no such value exists, print **-1.**

Sample Input
6
7 1 3 4 1 7

Sample Output
3

Explanation
Here, we have two options:
- ***a[1]*** and ***a[4]*** are both 1, so ***d[1,4] = |1-4| = 3***
- ***a[0]*** *and* ***a[5]*** are both 7, so ***d[0,5] = |0-5| = 5***

The answer is ***min(3,5) = 3***

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person.

   Input Format:
   The first line will have an integer ***n*** denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number.
   After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file.

   Constraints:
   A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros.
   $1 \leq n \leq 100000$
   $1 \leq Query \leq 100000$

   Output Format:
   For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number. See sample output for the exact format.
   To make the problem easier, we provided a portion of the code in the editor. You can either complete that code or write completely on your own.

   Sample Input:
   3
   uncle sam
   99912222
   tom
   11122222
   harry
   12299933
   uncle sam
   uncle tom
   harry

   Sample Output:
   uncle sam=99912222
   Not found
   harry=12299933

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. In computer science, a set is an abstract data type that can store certain values, without any particular order, and no repeated values {1,2,3}. is an example of a set, but {1,2,2} is not a set. Today you will learn how to use sets in java by solving this problem.
   You are given ***n*** pairs of strings. Two pairs ***(a,b)*** and ***(c,d)*** are identical if ***a=c*** and ***b=d*** . That also implies ***(a,b)*** is not same as ***(b,a)***. After taking each pair as input, you need to print number of unique pairs you currently have.
   Complete the code in the editor to solve this problem.

   Input Format:
   In the first line, there will be an integer ***T*** denoting number of pairs. Each of the next ***T*** lines will contain two strings seperated by a single space.

   Constraints:
   $1 \leq T \leq 100000$
   Length of each string is atmost 5 and will consist lower case letters only.

   Output Format:
   Print ***T*** lines. In the i_{th} line, print number of unique pairs you have after taking i_{th} pair as input.

   Sample Input:
   5
   john tom
   john mary
   john tom
   mary anna
   mary anna

   Sample Output:
   1
   2
   2
   3
   3

   Explanation:
   - After taking the first input, you have only one pair: (john,tom)
   - After taking the second input, you have two pairs: (john, tom) and (john, mary)
   - After taking the third input, you still have two unique pairs.
   - After taking the fourth input, you have three unique pairs: (john,tom), (john, mary) and (mary, anna)
   - After taking the fifth input, you still have three unique pairs.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 06: Binary Trees and Binary Search Trees

Date of the Session: **Time of the Session:**

PRE-LAB:

1. Differentiate Full binary trees and Perfect binary trees.

2. Create a Binary Search Tree for the following data and do in-order, Preorder and Post-order traversal of the tree. 50, 60, 25, 40, 30, 70, 35, 10, 55, 65, 5

3. Draw a balanced binary search tree containing the same numbers given in Q2.

4. How many non-null links are there in a binary tree with N nodes?

5. The integers 7, 1, 12, 8, 3, 0, –1, 9 are inserted in that order into an initially empty binary search tree. Draw the tree after the last insertion?

6. What is the property of Randomized BST?

IN-LAB:

1. Given an array A of N integers, classify it as being Good Bad or Average. It is called Good, if it contains exactly X distinct integers, Bad if it contains less than X distinct integers and Average if it contains more than X distinct integers.

   Input Format:
   First line consists of a single integer T denoting the number of test cases.
   First line of each test case consists of two space separated integers denoting N and X.
   Second line of each test case consists of N space separated integers denoting the array elements.

   Output Format:
   Print the required answer for each test case on a new line.

   Constraints:
   $1 \leq T \leq 50$
   $1 \leq X, N \leq 13000$

   Sample Input:
   4
   4 1
   1 4 2 5
   4 2
   4 2 1 5
   4 3
   5 2 4 1
   4 4
   1 2 4 5

Sample Output:

Average
Average
Average
Good

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. A sequence of numbers is called a wiggle sequence if the differences between successive numbers strictly alternate between positive and negative. The first difference (if one exists) may be either positive or negative. A sequence with fewer than two elements is trivially a wiggle sequence.

   For example, [1,7,4,9,2,5] is a wiggle sequence because the differences (6,-3,5,-7,3) are alternately positive and negative. In contrast, [1,4,7,2,5] and [1,7,4,5,5] are not wiggle sequences, the first because its first two differences are positive and the second because its last difference is zero.

   Given a sequence of integers, return the length of the longest subsequence that is a wiggle sequence. A subsequence is obtained by deleting some number of elements (eventually, also zero) from the original sequence, leaving the remaining elements in their original order.

   Example 1:
   Input: [1,7,4,9,2,5]
   Output: 6

   Example 2:
   Input: [1,17,5,10,13,15,10,5,16,8]
   Output: 7

   Example 3:
   Input: [1,2,3,4,5,6,7,8,9]
   Output: 2

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

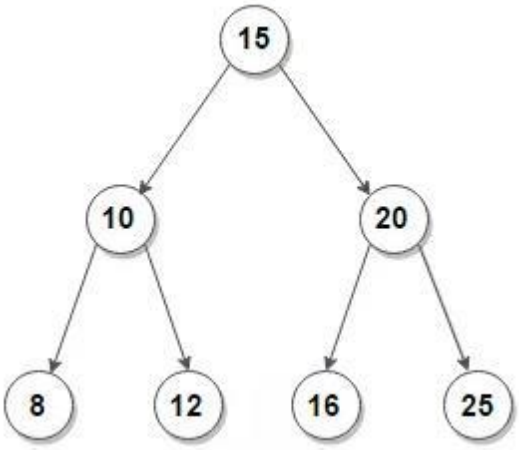
3. Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. (i.e., [0,1,2,4,5,6,7] might become [4,5,6,7,0,1,2]).
   Find the minimum element. You may assume no duplicate exists in the array.
   Example 1:
   Input: [3,4,5,1,2]
   Output: 1

   Example 2:
   Input: [4,5,6,7,0,1,2]
   Output: 0

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

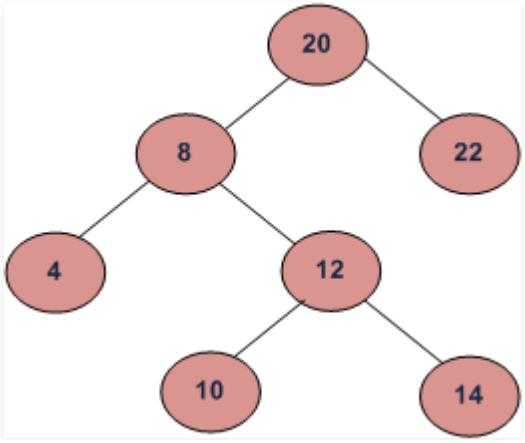
1. Find K^{th} smallest and largest element in BST
   Given a BST and a positive number K, Find Kth smallest and largest element in BST.
   For example, consider below BST if k=2, then find Kth smallest is 10 and Kth largest is 20 in the given tree.



Output: 10 20

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Sum and Product of minimum and maximum element of Binary Search Tree.
   Given a Binary Search Tree. The task is to find the sum and product of maximum and minimum value of tree.



OUTPUT:

Sum and product of maximum and minimum value of tree is 26 and 88.

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 07: Open Problems from Coding Contests

Date of the Session: **Time of the Session:**

PRE-LAB:

1. What is a hash table and hash function?

2. If several elements are competing for the same bucket in the hash table, what is it called?

3. What is direct addressing and mention its search complexity?

4. In function call, how the concepts of stack is used?

5. What is the search complexity in direct addressing?

6. Which data structures are applied when dealing with a recursive function?

IN-LAB:

1. Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sums to target.

   Each number in candidates may only be used **once** in the combination.

   Note:

   - All numbers (including target) will be positive integers.
   - The solution set must not contain duplicate combinations.

   Example 1:

   Input: candidates = [10,1,2,7,6,1,5], target = 8,

   A solution set is:

   [
   [1, 7],
   [1, 2, 5],
   [2, 6],
   [1, 1, 6]
   ]

   Example 2:

   Input: candidates = [2,5,2,1,2], target = 5,

   A solution set is:

   [
   [1,2,2],
   [5]
   ]

2. Jim enters a candy shop which has N different types of candies, each candy is of the same price. Jim has enough money to buy K candies. In how many different ways can he purchase K candies if there are infinite candies of each kind?

Input Format
The first line contains an integer T, the number of tests.
This is followed by 2T lines which contain T tests:
The first line (of each testcase) is an integer N and the second line (of each testcase) is an integer K.

Output Format
For each testcase, print the number of ways Jim can buy candies from the shop in a newline. If the answer has more than 9 digits, print the last 9 digits.

Note
This problem may expect you to have solved nCr Table

Constraints
$1 \leq T \leq 200$
$1 \leq N < 1000$
$1 \leq K < 1000$

Sample Input
2
4
1
2
3
Sample Output
4
4

3. Shyam Lal, a wealthy landlord from the state of Rajasthan, being an old fellow and tired of doing hard work, decided to sell all his farmland and to live rest of his life with that money. No other farmer is rich enough to buy all his land so he decided to partition the land into rectangular plots of different sizes with different cost per unit area. So, he sold these plots to the farmers but made a mistake. Being illiterate, he made partitions that could be overlapping. When the farmers came to know about it, they ran to him for compensation of extra money they paid to him. So, he decided to return all the money to the farmers of that land which was overlapping with other farmer's land to settle down the conflict. All the portion of conflicted land will be taken back by the landlord.

   To decide the total compensation, he has to calculate the total amount of money to return back to farmers with the same cost they had purchased from him. Suppose, Shyam Lal has a total land area of 1000 x 1000 equal square blocks where each block is equivalent to a unit square area which can be represented on the co-ordinate axis. Now find the total amount of money, he has to return to the farmers. Help Shyam Lal to accomplish this task.

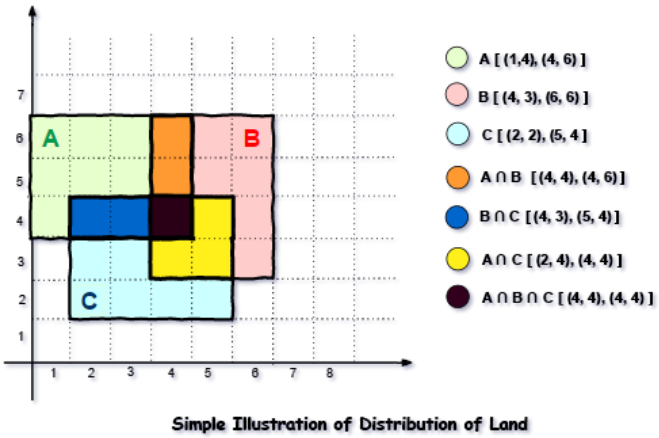
   Input Format:
   The first line of the input contains an integer N, denoting the total number of land pieces he had distributed. Next N line contains the 5 space separated integers (X1, Y1), (X2, Y2) to represent a rectangular piece of land, and cost per unit area C.
   (X1, Y1) and (X2, Y2) are the locations of first and last square block on the diagonal of the rectangular region.

   Output Format:
   Print the total amount he has to return to farmers to solve the conflict.

   Constraints:
   $1 \leq N \leq 100$
   $1 \leq X1 \leq X2 \leq 1000$
   $1 \leq Y1 \leq Y2 \leq 1000$
   $1 \leq C \leq 1000$

   Input
   3
   1 4 4 6 1
   4 3 6 6 2
   2 2 5 4 3
   Output
   35

   Explanation



Simple Illustration of Distribution of Land

For given sample input (see given graph for reference), compensation money for different farmers is as follows:
Farmer with land area A: C1 = 5 * 1 = 5
Farmer with land area B: C2 = 6 * 2 = 12
Farmer with land area C: C3 = 6 * 3 = 18
Total Compensation Money = C1 + C2 + C3 = 5 + 12 + 18 = 35

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. Dan is playing a video game in which his character competes in a hurdle race. Hurdles are of varying heights, and Dan has a maximum height he can jump. There is a magic potion he can take that will increase his maximum height by 1 unit for each dose. How many doses of the potion must he take to be able to jump all of the hurdles?

   Given an array of hurdle heights ***height***, and an initial maximum height Dan can jump, ***k***, determine the minimum number of doses Dan must take to be able to clear all the hurdles in the race.

   For example, if ***height=[1,2,3,3,2]*** and Dan can jump ***1*** unit high naturally, he must take ***3-1=2*** doses of potion to be able to jump all of the hurdles.

   Input Format:
   The first line contains two space-separated integers ***n*** and ***k***, the number of hurdles and the maximum height Dan can jump naturally.
   The second line contains ***n*** space-separated integers ***height[i]*** where 0≤i<n.

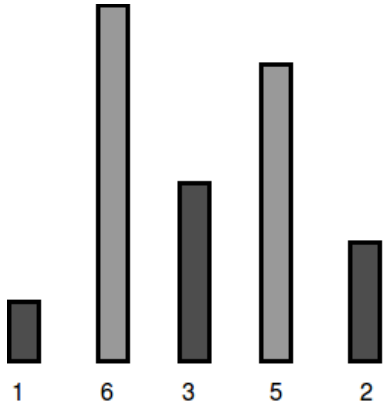
   Constraints:
   $1 \leq n, k \leq 100$
   $1 \leq height[i] \leq 100$

   Output Format:
   Print an integer denoting the minimum doses of magic potion Dan must drink to complete the hurdle race.

   Sample Input 0:
   5 4
   1 6 3 5 2

   Sample Output 0:
   2

   Explanation 0:
   Dan's character can jump a maximum of ***k=4*** units, but the tallest hurdle has a height of $h_1=6$:

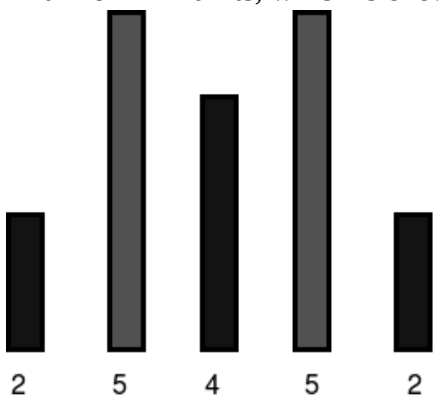


   To be able to jump all the hurdles, Dan must drink ***6-4=2*** doses.

Sample Input 1:
5 7
2 5 4 5 2

Sample Output 1:
0

Explanation 1:
Dan's character can jump a maximum of ***k=7*** units, which is enough to cross all the hurdles:



Because he can already jump all the hurdles, Dan needs to drink ***0*** doses.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Eliza likes to play games with integers. She has created a new game where she determines the difference between a number and its reverse. For instance, given the number ***12***, its reverse is ***21***. Their difference is ***9***. The number ***120*** reversed is ***21***, and their difference is ***99***.

   She decides to apply her game to decision making. She will look at a numbered range of days and will only go to a movie on a beautiful day.

   Given a range of numbered days, ***[i,…j]*** and a number ***k***, determine the number of days in the range that are beautiful. Beautiful numbers are defined as numbers where **|*i-reverse(i)*|** is evenly divisible by ***k***. If a day's value is a beautiful number, it is a beautiful day. Print the number of beautiful days in the range.

   Input Format:
   A single line of three space-separated integers describing the respective values of ***i***, ***j***, and ***k***.
   Constraints:
   $1 \leq i, j \leq 100$
   $1 \leq k \leq 100$
   Output Format:
   Print the number of beautiful days in the inclusive range between ***i*** and ***j***.
   Sample Input:
   20 23 6
   Sample Output:
   2
   Explanation
   Eliza may go to the movies on days 20, 21, 22, and 23. We perform the following calculations to determine which days are beautiful:
   Day 20 is beautiful because the following evaluates to a whole number: $\frac{|20-02|}{6} = \frac{18}{6} = 3$
   Day 21 is not beautiful because the following doesn't evaluate to a whole number: $\frac{|21-12|}{6} = \frac{9}{6} = 1.5$
   Day 22 is beautiful because the following evaluates to a whole number: $\frac{|22-22|}{6} = \frac{0}{6} = 0$
   Day 23 is not beautiful because the following doesn't evaluate to a whole number: $\frac{|23-32|}{6} = \frac{9}{6} = 1.5$
   Only two days, 20 and 22, in this interval are beautiful. Thus, we print 2 as our answer.

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 08: Open Problems from Coding Contests

Date of the Session: **Time of the Session:**

PRE-LAB:

1. Can you declare an array without assigning the size of an array?

2. How to find all permutations of String?

3. How to use static variable and static function?

4. When to use Inline functions?

5. Discuss how to implement queue using stack

6. What are the types of Binary tree?

IN-LAB:

1. Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. (i.e., [0,1,2,4,5,6,7] might become [4,5,6,7,0,1,2]).
   Find the minimum element. You may assume no duplicate exists in the array.
   Example 1:
   Input: [3,4,5,1,2]
   Output: 1

   Example 2:
   Input: [4,5,6,7,0,1,2]
   Output: 0

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Let's call an array A a mountain if the following properties hold:
   - A.length >= 3
   - There exists some 0 < i < A.length - 1 such that
     A[0] < A[1] < ... A[i-1] < A[i] > A[i+1] > ... > A[A.length - 1]

   Given an array that is definitely a mountain, return any i such that
   A[0] < A[1] < ... A[i-1] < A[i] > A[i+1] > ... > A[A.length - 1].

   Example 1:
   Input: [0,1,0]
   Output: 1

   Example 2:
   Input: [0,2,1,0]
   Output: 1

   Note:
   - 3 <= A.length <= 10000
   - 0 <= A[i] <= 10^6
   - A is a mountain, as defined above.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

3. Given a value N, find the number of ways to make change for N cents, if we have infinite supply of each of S = { S1, S2, .. , Sm} valued coins. The order of coins doesn't matter. For example, for N = 4 and S = {1,2,3}, there are four solutions: {1,1,1,1},{1,1,2},{2,2},{1,3}. So output should be 4. For N = 10 and S = {2, 5, 3, 6}, there are five solutions: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}. So the output should be 5.

   Input:
   The first line contains an integer 'T' denoting the total number of test cases. In each test cases, the first line contains an integer 'M' denoting the size of array. The second line contains M space-separated integers A1, A2, ..., AN denoting the elements of the array. The third line contains an integer 'N' denoting the cents.

   Output:
   Print number of possible ways to make change for N cents.

   Constraints:
   $1 \leq T \leq 50$
   $1 \leq N \leq 300$
   $1 \leq A[i] \leq 300$

   Example:
   Input:
   2
   3
   1 2 3
   4
   4
   2 5 3 6
   10

   Output:
   4
   5

   Explanation:
   Testcase 1: The possibilities are as such: {1, 1, 1, 1}, {1, 1, 2}, {1, 3}, {2, 2}.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

POST-LAB:

1. A word from the English dictionary is taken and arranged as a matrix. e.g. "MATHEMATICS"
   MATHE
   ATHEM
   THEMA
   HEMAT
   EMATI
   MATIC
   ATICS
   There are many ways to trace this matrix in a way that helps you construct this word. You start tracing the matrix from the top-left position and at each iteration, you either move RIGHT or DOWN, and ultimately reach the bottom-right of the matrix. It is assured that any such tracing generates the same word. How many such tracings can be possible for a given word of length m+n-1 written as a matrix of size m * n?

   Input Format
   The first line of input contains an integer T. T test cases follow.
   Each test case contains 2 space separated integers m & n (in a new line) indicating that the matrix has m rows and each row has n characters.

   Constraints
   $1 <= T <= 10^3$
   $1 \leq m,n \leq 10^6$

   Output Format
   Print the number of ways (S) the word can be traced as explained in the problem statement. If the number is larger than 10^9+7, print S mod (10^9 + 7) for each testcase (in a new line).

   Sample Input
   1
   2 3

   Sample Output
   3

   Explanation
   Let's consider a word AWAY written as the matrix
   AWA
   WAY
   Here, the word AWAY can be traced in 3 different ways, traversing either RIGHT or DOWN.

   AWA
    Y

   AW
    AY

   A
   WAY

   Hence the answer is 3.

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

2. Given a string S of '(' and ')' parentheses, we add the minimum number of parentheses ('(' or ')', and in any positions) so that the resulting parentheses string is valid.
   Formally, a parentheses string is valid if and only if:
   - It is the empty string, or
   - It can be written as AB (A concatenated with B), where A and B are valid strings, or
   - It can be written as (A), where A is a valid string.

   Given a parentheses string, return the minimum number of parentheses we must add to make the resulting string valid.

   Input:
   ()))((

   Output:
   4

   Explanation:
   The given string can be balanced to "((()))(())", requiring 4 more paranthesis minimum.

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 09: Case Study - 1

Date of the Session: **Time of the Session:**

Case Study: Hello World

Bob is a newbie to programming and while learning the programming language he came across the following rules:

1. Each program must start with '{' and will end with '}'.
2. Each program must contain only one main program. Main program will start with '<' and will end with '>'.
3. A program may or may not contain user defined functions but there is no limitation about the number of user defined functions present in the program. User defined program will start with '(' and will end with ')'.
4. loops are allowed only inside the functions (this function can be either main function or user defined function(s)). Every loop will start with '{' and will end with '}'.
5. User defined function(s) are not allowed to define inside main function or user defined function(s).
6. Nested loops (loop inside a loop) are allowed.
7. Instructions can be defined anywhere inside the program.

If any of the above conditions does not met, then the program will generate compilation errors. Today Bob has written his first program "Hello World", but he is not sure about the correctness of the program now your task is to help him to find out whether his program will compile without any errors or not.

Input Format:

Each Input will contain a single line L, where L is a program written by Bob.

Line 1	L,where L is a single line program written by Bob.

Constraints:

L is a text and can be composed of any of the characters {, }, (,) , <, >and P, where P will represent the instruction.

L will contain single spaced characters where each character will represent each line of the program.

Number of characters in the text < = 10000

Input Format:

Line 1	For Valid Input,print “No Compilation Errors“, if there is no compilation error For Invalid Input,print “Compilation Error“

Sample Input and Output:

SL. No.	Input	Output
1.	{ < > (P) }	No Compilation Errors
2.	{ < { } > ({ }))	Compilation Errors
3.	{ ({ }) }	Compilation Errors

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Registration No.: **Academic Year: 2021-2022**

Lab Session 10: Case Study - 2

Date of the Session: **Time of the Session:**

Case Study: Isotope

Scientist recently found a new element X, which can have different isotopes upto an atomic value of 199. Speciality of element X is that when two atoms fuse they produce energy multiple of their atomic value and forms an atom with atomic value of their multiple modulus 199.
For Eg:
If atom1 with value 56 and atom2 with value 61 fuse.
They produce energy of 3416 KJ (56 * 61)
Resulting atom will have atomic value (56*61) mod 199 = 33

Scientists created a nuclear reactor to create energy by this method. Everyday they get several atoms of X from supplier in a particular sequence. Since this element highly radioactive they cant risk by changing its sequence. So each atom can fuse only with another atom nearby. Nevertheless scientists can choose order of fusion thereby maximizing total energy produced.
Now, for given sequence of atomic values, output maximum energy that can be produced.
For Eg:
If sequence of atoms are
56,61, 2

Then they can produce 3416KJ by fusing 56&61 which results in an atom of value 33. Then they can fuse 33 and 2 to get energy of 66KJ. So total energy generated is 3482.

But if they cleverly choose to fuse atoms 61 & 2 first then they generate 122 KJ with a resulting atom of value 122. Then if they fuse 122 and 56, they can generate 6832 KJ. So total energy is 6954.

Hence Maximum energy that can be generated from this sequence is 6954.

Input Format:
Input starts with a number specifying number of inputs(n) followed by n space separated integers specifying atomic values of n atoms.

Line 1	N,where N is the number of atoms in the sequence to be fused.
Line 2	a_1 a_2 a_3 .... a_n where ai -> Atomic value of i^{th} atom. and two atoms are space delimited

Constraints:
$0 < a_i < 199$
$1 < n < 1000$

Output Format:

Print Determinant of the input matrix rounded up to 3 digits after decimal point, on console in the format shown in table below

Line 1	**For Valid Input,print** E,where E is Integer stating maximum energy that can be produced **For Invalid Input,print** INVALID INPUT

Sample Input and Output:

SL. No.	Input	Output
1.	3 15 75 60	8925
2.	3	INVALID INPUT
3.	3 15 0 6	INVALID INPUT
4.	3 5 5 199	INVALID INPUT
5.	4 15 75 60 45	18515

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**

Lab Session 11: Case Study - 3

Date of the Session: **Time of the Session:**

Case Study: Trace the Rats

Given a square maze (A) of dimension N, every entry (A_{ij}) in the maze is either an open cell 'O' or a wall 'X'. A rat can travel to its adjacent locations (left, right, top and bottom), but to reach a cell, it must be open. Your task is to place some rats in the maze such that no other rats can reach other rats. Output the maximum number of rats that can be placed in the maze A.

Input Format:

Input consists of two parts, viz.

1. Size of the maze (N)
2. The maze itself (A = N * N)

Constraints:

1<=N<=350

A_{ij} = {'O','X'}

Output Format:

Print the maximum number of rats that can be placed in the maze, such that they can't reach other.

Sample Input and Output:

SL. No.	Input	Output
1.	3 O O X O X O O O X	2

VELAGAPUDI RAMAKRISHNA
SIDDHARTHA ENGINEERING COLLEGE
VIJAYAWADA

Student's Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	**Evaluator's Observation** **Marks Secured: _____ out of _____** **Full Name of the Evaluator:** **Signature of the Evaluator:** **Date of Evaluation:**