1. Washing Machine

ID:12155
Solved By 253 Users

TCS NQT

The program must accept an integer representing the weight of the clothes to be washed in a washing machine. There are **3** water levels in the washing mechanism which are given below.
- **Low Water Level:** The estimated time is **25** minutes, where the approximate weight is between **1** gram to **2000** grams (both inclusive).
- **Medium Water Level:** The estimated time is **35** minutes, where the approximate weight is between **2001** grams and **4000** grams (both inclusive).
- **High Water Level:** The estimated time is **45** minutes, where the approximate weight is above **4000** grams.

The maximum capacity of the machine is **7000** grams. The program must print the estimated time for the given weight of clothes **W** as the output. If the weight is more than 7000 grams, the program must print the string value "**OVERLOADED**" as the output. If the weight is **0**, the program must print **0** as the output. For all other values of W, the program must print "**INVALID INPUT**" as the output.

Input Format:
The first line contains an integer representing the weight of the clothes.

Output Format:
The first line contains the estimated time for the given weight of clothes W or the string value as per the given conditions.

Example Input/Output 1:
Input:
2000

Output:
25

Example Input/Output 2:
Input:
7500

Output:
OVERLOADED

Example Input/Output 3:
Input:
-5000

Output:
INVALID INPUT

Max Execution Time Limit: 50 millisecs

2. Invalid Mobile Numbers

ID:12220
Solved By 107 Users

TCS NQT

The program must accept a list of mobile numbers (in separate lines) as the input. The character **q** or **Q** represents the end of the input. The program must print the number of invalid mobile numbers among the given list of mobile numbers as the output. A valid mobile number contains exactly **10** digits.

Boundary Condition(s):
1 <= Length of each string <= 20

Input Format:
The lines, each contains a string value.

Output Format:
The first line contains the number of invalid mobile numbers.

Example Input/Output 1:
Input:
9854653221
997878A1576
9992224578
999225789900
986578989B
817524990
Q

Output:
4

Explanation:
The **4** invalid mobile numbers are given below.
997878A1576
999225789900
986578989B
817524990
Hence the output is 4.

Example Input/Output 2:
Input:
7006868327
8745152751
90379383946799
7180S16U65
9404722712T
929191K54588G6
84385659
q

Output:
5

Max Execution Time Limit: 50 millisecs

3. Grouping Colorful LEDs

ID:12057
Solved By 70 Users

Accenture

The program must accept the names of the colors in an LED serial set as the input. The program must find the number of groups of LEDs having the same color at the beginning and the end in the given LED serial set. Also consider each LED in the given LED serial set as a group. Finally, the program must print the number of groups of LEDs as the output.

Boundary Condition(s):
1 <= Length of each color's name <= 20

Input Format:
The first line contains a space separated string values representing the names of the colors in an LED serial set.

Output Format:
The first line contains the number of groups LEDs having the same color at the beginning and the end in the given LED serial set.

Example Input/Output 1:
Input:
Red Blue Green Blue

Output:
5

Explanation:
The **5** groups of LEDs are given below.
Red
Blue
Green
Blue
Blue Green Blue

Example Input/Output 2:
Input:
Yellow Red Yellow Green Blue Yellow Green

Output:
11

Explanation:
The **11** groups of LEDs are given below.
Yellow
Red
Yellow
Green
Blue
Yellow
Green
Yellow Red Yellow
Yellow Red Yellow Green Blue Yellow
Yellow Green Blue Yellow
Green Blue Yellow Green

Max Execution Time Limit: 50 millisecs

4. Car Parking - Park & Search

ID:12188
Solved By 48 Users

TCS NQT

In a car parking area, **N** cars are parked. The maximum number of cars that can be parked in the area is **M**. Each car must be parked at the next available position of the last car parked.
The program must accept the registration numbers of the N cars and the value of M as the input. The program also accepts Q queries as the input. Each query contains an integer representing the option **X** and a string representing the registration number **R** of a car.
- **Option X = 1:** The program must add the registration number R to the already parked cars and print the string value "**Car parked at position:**" followed by the position of the newly parked car. If the parking area is full, the program must print the string value "**No space for parking**" as the output.
- **Option X = 2:** The program must search the given registration number R among the parked cars. If it is found, the program must print the string value "**Car position:**" followed by the position of the car. If the registration number R is not found, the program must print the string value "**Car does not exists**" as the output.
- For any other options, the program must print the string value "**Invalid**" as the output.

For each query, the program must print the output in separated lines based on the given conditions.

Boundary Condition(s):
1 <= N <= M <= 100
6 <= Length of each car's registration number <= 12
1 <= Q <= 20

Input Format:
The first line contains N.
The second line contains the registration number of the N cars separated by a space.
The third line contains the maximum number of cars M that can be parked in the area.
The fourth line contains Q.
The next Q lines, each contains an integer X and the registration number R of a car separated by a space.

Output Format:
The first Q lines, each contains a string value based on the given conditions.

Example Input/Output 1:
Input:
```
5
TN6548 MH1247 KA3057 KL2154 TN6999
7
6
1 KA1200
```

2 KL2154
4 JK9842
1 MH0055
1 KA5555
2 TK1872

Output:
Car parked at position: 6
Car position: 4
Invalid
Car parked at position: 7
No space for parking
Car does not exists

Explanation:
Here **Q=6**.
Query 1: The given option is **1** and the maximum cars that can be parked is **7**. Here only **5** cars are parked, so the car **KA1200** is parked at the position **6**.
Query 2: The given option is **2** and the registration number is **KL2154**. The car with the registration number KL2154 is already parked at the position **4**.
Query 3: The given option is **4**, the options other than 1 and 2 are **Invalid**.
Query 4: The given option is **1** and the maximum cars that can be parked is **7**. Here only **6** cars are parked, so the car **MH0055** is parked at the position **7**.
Query 5: The given option is **1** and the number of cars parked is **7**. Now the parking area is full, so no space for parking.
Query 6: The given option is **2** and the registration number is **TK1872**. There is no car with the registration number TK1872, so the car does not exists.

Example Input/Output 2:
Input:
6
TN4578 KL1458 KA5785 TN1245 TN6945 KA2456
9
8
1 KL1125
3 TN5785
2 TN1245
1 TN2582
1 PB1241
2 KA4545
1 MH4575
2 TN4578

Output:
Car parked at position: 7
Invalid

```
Car position: 4
Car parked at position: 8
Car parked at position: 9
Car does not exists
No space for parking
Car position: 1
```

Max Execution Time Limit: 50 millisecs

5. function productOfSums

ID:12074
Solved By 38 Users

Accenture

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all test cases. Do not write the **main()** function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **productOfSums** accepts two arguments - **SIZE** and **arr**, an integer representing the size of the integer array arr and the integer array arr. The array can be divided into two parts, where the first part of the array (starting from **0** to **i**) is in ascending order and the second part of the array (starting from **i** to **SIZE-1**) is in descending order.

The function/method **productOfSums** must find the sum of the first part **F** and the sum of the second part **S**. Finally, the function must return an integer representing the product of F and S. The function must return **-1** if the size of the array is less than **3**.

Your task is to complete the code in the function **productOfSums** so that it passes all the test cases.

Boundary Condition(s):
1 <= N <= 100
-10^5 <= Each integer value <= 10^5

Example Input/Output 1:
Input:

7
4 7 15 11 10 5 2

Output:
1118

Explanation:
The sum of integers in the first part of the array is **26** (4 + 7 + 15).
The sum of integers in the second part of the array is **43** (15 + 11 + 10 + 5 + 2).
The product of **26** and **43** is **1118**.

Example Input/Output 2:
Input:
4
4 8 -9 -10

Output:
-132

Max Execution Time Limit: 50 millisecs

6. function customCaesarCipher

ID:12186
Solved By 29 Users

TCS NQT

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all test cases. Do not write the main() function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **customCaesarCipher** accepts two arguments - **key** and **message**, an integer representing the key to be used to encrypt the given message and a string representing the message to be encrypted.

The function/method **customCaesarCipher** must return a string containing the encrypted message. If the value of key is less than **0**, the function/method must return the string value **"INVALID INPUT"**.

The encryption algorithm is given below.
- Each alphabet in the message must be shifted by key number of places down the alphabets.
- Each digit in the message must be shifted by key number of places down the digits.
- For all other characters in the message, they remain the same.

Note: The English alphabet set and the digits (0 to 9) must be considered in a cyclic fashion for the encryption.

Your task is to complete the code in **customCaesarCipher** so that it passes all the test cases.

Example Input/Output 1:
Input:
1
All is well. #45.9K Tweets

Output:
Bmm jt xfmm. #56.0L Uxffut

Explanation:
Here the given string is **All is well. #45.9K Tweets** and the key value is **1**.
After shifting all the alphabets and digits in the string by **1**, the string becomes **Bmm jt xfmm. #56.0L Uxffut**
Hence the output is Bmm jt xfmm. #56.0L Uxffut

Example Input/Output 2:
Input:
4
Good bye... Mr. XYZ

Output:
Kssh fci... Qv. BCD

Example Input/Output 3:
Input:
-2
SkillRack

Output:
INVALID INPUT

Max Execution Time Limit: 50 millisecs

7. Count Strong Points

ID:12034
Solved By 24 Users

Accenture

The program must accept an integer matrix of size **RxC** as the input. The program must print the number of strong points in the given matrix as the output. The strong points are located at the element(s) whose all the surrounding elements are smaller than the given element.

Boundary Condition(s):
2 <= R, C <= 50
1 <= Matrix element value <= 1000

Input Format:
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.

Output Format:
The first line contains the number of strong points in the given matrix.

Example Input/Output 1:
Input:

```
3 3
56 92 45
19 41 51
55 31 80
```

Output:

```
3
```

Explanation:
The **3** strong points are given below.
92 > [56, 19, 41, 51, 45]
55 > [19, 41, 31]
80 > [41, 51, 31]
Hence the output is 3.

Example Input/Output 2:
Input:

```
6 5
69 45 47 35 62
43 68 22 55 72
```

53 96 21 24 49
89 34 86 10 37
94 31 93 12 70
74 81 13 60 95

Output:
6

Max Execution Time Limit: 50 millisecs

8. Trains - Departure Time

ID:12182
Solved By 20 Users

TCS NQT

A train **X** is travelling from station **A** to station **B** via **N** stations. The departure time (in 24-hr format) of the train X at each station is passed as the input to the program. To avoid congestion, the railway department has planned to launch another train Y from the station A to the station B with the same time intervals as the train X. The departure time of the train Y at the station A is also passed as the input. The program must print the departure time (in 24-hr format) of the train Y at each station as the output.

Boundary Condition(s):
2 <= N <= 100
0 <= HH <= 23
0 <= MM <= 59

Input Format:
The first line contains N.
The second line contains the departure time (in 24-hr format) of the train X at each station separated by a space.
The third line contains the departure time of the train Y at the station A.

Output Format:
The first line contains the departure time (in 24-hr format) of the train Y at each station separated by a space.

Example Input/Output 1:
Input:
5

```
10:00 10:15 11:05 11:20 11:30
13:00
```

Output:

```
13:00 13:15 14:05 14:20 14:30
```

Explanation:
The train X departs from the 1st station at 10:00. So the train Y departs from the 1st station at 13:00.
The train X departs from the 2nd station at 10:15. So the train Y departs from the 2nd station at 13:15.
The train X departs from the 3rd station at 11:05. So the train Y departs from the 3rd station at 14:05.
The train X departs from the 4th station at 11:20. So the train Y departs from the 4th station at 14:20.
The train X departs from the 5th station at 11:30. So the train Y departs from the 5th station at 14:30.
Hence the output is
13:00 13:15 14:05 14:20 14:30

Example Input/Output 2:
Input:

```
6
04:20 04:40 04:55 05:30 06:05 06:36
05:10
```

Output:

```
05:10 05:30 05:45 06:20 06:55 07:26
```

Max Execution Time Limit: 50 millisecs

9. function findSequence

ID:12058
Solved By 19 Users

Accenture

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all test cases. Do not write the **main()** function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **findSequence** accepts two arguments - **SIZE** and **arr**, an integer representing the size of the integer array arr and the integer array arr.

The function/method **findSequence** must return **1** if it is possible to split the array arr into two sets in such a way that the first set is strictly decreasing while the second set is strictly increasing. Else the function must return **0**.

Your task is to complete the code in the function **findSequence** so that it passes all the test cases.

Boundary Condition(s):
4 <= N <= 100
-10^5 <= Each integer value <= 10^5

Example Input/Output 1:
Input:
7
4 2 -1 0 1 2 5

Output:
1

Explanation:
There are two possible ways to split the array which are given below.
(**4 2 -1**) and (**0 1 2 5**)
(**4 2**) and (**-1 0 1 2 5**)

Example Input/Output 2:
Input:
6
4 1 9 10 12 11

Output:
0

Max Execution Time Limit: 50 millisecs

10. function countHardPrograms

ID:12075
Solved By 18 Users

Accenture

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all test cases. Do not write the **main()** function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **countHardPrograms** accepts three arguments - **N**, **arr** and **K**, an integer representing the number of chapters in a competitive programming book, an array of integers representing the number of programs in the N chapters and an integer K representing the maximum number of programs a page can hold.

The format of the competitive programming book is as follows.
- The N chapters are numbered from **1** to **N** (The chapter 1 always starts from the page 1).
- The programs in each chapter are numbered from **1** to **X**, where X is the number of programs in a chapter.
- Each page can hold up to K programs. Only the last page of a chapter may contain less than K programs.
- Each chapter starts on a new page, so a page will never contains programs from more than one chapter.

The function/method **countHardPrograms** must return an integer representing the number of hard programs in the book. The hard programs are identified based on the following condition.
- If there is a program on a page whose page number is equal to its number(the program's number), then the program is called a hard program.

Your task is to complete the code in the function **countHardPrograms** so that it passes all the test cases.

Boundary Condition(s):
1 <= N, K, Each integer value <= 100

Example Input/Output 1:
Input:
5 3
4 2 1 10 3

Output:
3

Explanation:
The **1**st chapter contains **4** programs.
The **2**nd chapter contains **2** programs.
The **3**rd chapter contains **1** program.
The **4**th chapter contains **10** programs.
The **5**th chapter contains **3** programs.
Each page can hold K = 3 programs.
The **1**st page contains the programs **1**, **2**, **3** (chapter 1).
The **2**nd page contains the program **4** (chapter 1).
The **3**rd page contains the programs **1**, **2** (chapter 2).
The **4**th page contains the program **1** (chapter 3).
The **5**th page contains the programs **1**, **2**, **3** (chapter 4).
The **6**th page contains the programs **4**, **5**, **6** (chapter 4).
The **7**th page contains the programs **7**, **8**, **9** (chapter 4).
The **8**th page contains the program **10** (chapter 4).
The **9**th page contains the programs **1**, **2**, **3** (chapter 5).
The **3** hard programs are given below.
Chapter 1: 1st program (Page number is **1**).
Chapter 4: 6th program (Page number is **6**).
Chapter 4: 7th program (Page number is **7**).

Example Input/Output 2:
Input:
6 5
6 3 4 13 5 14

Output:
7

Max Execution Time Limit: 50 millisecs

11. Count of Non-repeated Characters

ID:12333
Solved By 72 Users

WIPRO

The program must accept a string **S** as the input. The program must print the count of characters that are not repeated in the string S as the output.

Boundary Condition(s):
1 <= Length of S <= 1000

Input Format:
The first line contains S.

Output Format:
The first line contains the count of characters that are not repeated in the string S.

Example Input/Output 1:
Input:
skillrack

Output:
5

Explanation:
The non-repeated characters in the given string are **s**, **i**, **r**, **a** and **c**.
So their count 5 is printed as the output.

Example Input/Output 2:
Input:
Aabac#1123

Output:
6

Explanation:
The non-repeated characters in the given string are **A**, **b**, **c**, **#**, **2** and **3**.
So their count 6 is printed as the output.

Max Execution Time Limit: 50 millisecs

12. Generate Key

ID:12330
Solved By 64 Users

WIPRO

The program must accept three integers **X**, **Y** and **Z** as the input. Each of these are four-digit integers. The program must generate a four-digit key **K** based on the following conditions.

- The **1st** digit in K must be equal to the **SMALLEST** digit in the thousands place of all three integers.
- The **2nd** digit in K must be equal to the **LARGEST** digit in the hundreds place of all three integers.
- The **3rd** digit in K must be equal to the **SMALLEST** digit in the tens place of all three integers.
- The **4th** digit in K must be equal to the **LARGEST** digit in the units place of all three integers.

Boundary Condition(s):
1000 <= X, Y, Z = 9999

Input Format:
The first line contains X, Y and Z separated by a space.

Output Format:
The first line contains K.

Example Input/Output 1:
Input:
5107 1234 2948

Output:
1908

Explanation:
The digits in the thousands place of all three integers are **5**, **1** and **2**. Here **1** is the smallest digit, so 1 is printed.
The digits in the hundreds place of all three integers are **1**, **2** and **9**. Here **9** is the largest digit, so 9 is printed.
The digits in the tens place of all three integers are **0**, **3** and **4**. Here **0** is the smallest digit, so 0 is printed.
The digits in the units place of all three integers are **7**, **4** and **8**. Here **8** is the largest digit, so 8 is printed.

Example Input/Output 2:
Input:
5496 7413 2100

Output:
2406

Max Execution Time Limit: 50 millisecs

13. Sum of Non-prime Digits

ID:12334
Solved By 57 Users

WIPRO

The program must accept an integer **N** as the input. The program must print the sum of non-prime digits in N as the output. If there is no such digit, the program must print **-1** as the output.

Boundary Condition(s):
1 <= N <= 10^8

Input Format:
The first line contains N.

Output Format:
The first line the sum of non-prime digits in N or -1.

Example Input/Output 1:
Input:
219517

Output:
11

Explanation:
The non-prime digits in **219517** are **1**, **9** and **1**.
The sum of non-prime digits is **11** (1 + 9 + 1). So 11 is printed.

Example Input/Output 2:
Input:
3005

Output:
0

Explanation:
The non-prime digits in **3005** are **0** and **0**.
The sum of non-prime digits is **0**. So 0 is printed.

Example Input/Output 3:
Input:
725

Output:
-1

Max Execution Time Limit: 50 millisecs

14. Decode Digital Lock

ID:12331
Solved By 50 Users

WIPRO

The program must accept an integer **X** representing the encoded key of a digital lock. There are two ways to decode the key of the digital lock.
- If X is an armstrong number, then the key is the sum of even digits in the encoded key.
- If X is not an armstrong number, then the key is the sum of odd digits in the encoded key.
An armstrong number is a number that is the sum of its own digits each raised to the power of the number of digits in the number.
The program must decode the key X and print it as the output.

Boundary Condition(s):
1 <= X <= 10^8

Input Format:
The first line contains X.

Output Format:
The first line contains the decoded key.

Example Input/Output 1:
Input:
1634

Output:
10

Explanation:
The given integer **1634** ($1^4 + 6^4 + 3^4 + 4^4$) is an armstrong number.
The sum of even digits in 1634 is **10** (6 + 4). So 10 is printed.

Example Input/Output 2:
Input:
4105

Output:
6

Max Execution Time Limit: 50 millisecs

15. Swap Every Two Digits

ID:12332
Solved By 43 Users

WIPRO

The program must accept two integers **X** and **Y** as the input. The program must swap every two digits in both X and Y. Then the program must print the sum of the revised values of X and Y as the output. If the number of digits in an integer is odd, the last digit remains the same as there is no digit to swap.

Boundary Condition(s):
1 <= X, Y <= 10^8

Input Format:
The first line contains X and Y separated by a space.

Output Format:
The first line contains the sum of the revised values of X and Y.

Example Input/Output 1:
Input:
27521 7809

Output:
81041

Explanation:
After swapping every two digits in X, it becomes **72251**.
After swapping every two digits in Y, it becomes **8790**.
The sum of 72251 and 8790 is **81041**.
So 81041 is printed as the output.

Example Input/Output 2:
Input:
1005 12345

Output:
21585

Explanation:
After swapping every two digits in X, it becomes **150**.
After swapping every two digits in Y, it becomes **21435**.
The sum of 150 and 21435 is **21585**.
So 21585 is printed as the output.

Max Execution Time Limit: 50 millisecs

16. Physical Fitness Test

ID:12234
Solved By 35 Users

TCS NQT

A physical fitness test (running test) is conducted to select candidates for military training in a country. A batch of **3** candidates appearing for the running test on track for **3** rounds. The program must accept **9** integers indicating the oxygen level after each round for 3 candidates as the input. For each candidate, the program must calculate the average oxygen level over the 3 rounds. The candidate who has the highest average oxygen level is the most fit candidate. The program must print the string value "**Candidate Number:** " followed by the most fit candidate's number as the output. If more than one trainee attains the same highest average level, they all need to be printed.

Boundary Condition(s):
1 <= Oxygen level <= 100

Input Format:
The first 3 lines, each contains an integer representing the oxygen level of the 3 candidates after the round 1.
The middle 3 lines, each contains an integer representing the oxygen level of the 3 candidates after the round 2.
The last 3 lines, each contains an integer representing the oxygen level of the 3 candidates after the round 3.

Output Format:
The line(s), each contains the string value "Candidate Number: " followed by the most fit candidate's number.

Example Input/Output 1:
Input:
```
96
93
96
93
91
93
91
93
91
```

Output:
```
Candidate Number: 1
Candidate Number: 3
```

Explanation:
The oxygen level of **candidate 1** after each round is **96 93 91**
The oxygen level of **candidate 2** after each round is **93 91 93**
The oxygen level of **candidate 3** after each round is **96 93 91**
The average oxygen level of each candidate is **93.33**, **92.33** and **93.33**.
Here the **candidate 1** and the **candidate 3** have the highest average oxygen level.
Hence the output is
Candidate Number: 1
Candidate Number: 3

Example Input/Output 2:
Input:
```
92
86
95
85
85
87
82
82
82
```

Output:
```
Candidate Number: 3
```

Max Execution Time Limit: 50 millisecs

17. Wall Painting Cost

ID:12236
Solved By 35 Users

TCS NQT

Interior wall painting cost is **X** rupees per square feet and exterior wall painting cost is **Y** rupees per square feet. A house has **M** interior walls and **N** exterior walls to paint. The program must accept the surface area (in units of square feet) of the M interior walls and N exterior walls as the input. The program must print the total estimated cost with the precision up to **2** decimal places as the output.

Boundary Condition(s):
1 <= X, Y, M, N <= 100

Input Format:
The first line contains X and Y separated by a space.
The second line contains M and N separated by a space.
The next (M+N) lines, each contains a floating point value representing the surface area of a wall (Interior walls followed by exterior walls).

Output Format:
The first line contains the total estimated cost with the precision up to 2 decimal places.

Example Input/Output 1:
Input:

```
18 12
6 3
14.3
15.2
10.5
12.5
15.2
14.3
10.25
10.25
10.0
```

Output:

```
1842.00
```

Explanation:
The interior wall painting cost is **18** rupees per square feet and the exterior wall painting cost is **12** rupees per square feet.
Here **M=6** and **N=3**.
The total surface area of **6** interior walls is **82** (14.3+15.2+10.5+12.5+15.2+14.3) square feet.
The total cost to paint **6** interior walls is **1476** (82*18) rupees.
The total surface area of **3** exterior walls is **30.5** (10.25+10.25+10.0) square feet.
The total cost to paint **3** exterior walls is **366** (30.5*12) rupees.
The total estimated cost with the precision upto two decimal places is **1842.00** (1476 + 366) rupees.

Example Input/Output 2:
Input:

```
7 10
4 3
24.5
12.2
13.7
5.6
32.5
10.2
12.3
```

Output:

```
942.00
```

Max Execution Time Limit: 50 millisecs

18. Water Tank & Buckets

ID:12221
Solved By 28 Users

TCS NQT

There is a water tank with the maximum capacity of **X** litres and a bucket with the maximum capacity of **Y** litres. A boy wants to fill greater than or equal to 80% of the water tank with the water using the bucket. The amount of water taken at a time in the bucket is not fixed as it can be any value less than or equal to Y. The program must accept the values of X, Y and a list of integers representing the amount of water taken in the buckets to fill the water tank. The program must print the number of buckets **B** poured into the water tank to reach greater than or equal to

80% of water. If it is not possible to fill 80% of the water tank, the program must print the string value "**TANK NOT FULL**" as the output.

Boundary Condition(s):
1 <= Y <= X <= 1000

Input Format:
The first line contains X.
The second line contains Y.
The line(s), each contains an integer representing the amount of water taken in the bucket.

Output Format:
The first line contains B or the string value "TANK NOT FULL".

Example Input/Output 1:
Input:
```
100
20
20
15
10
20
20
20
```

Output:
```
5
```

Explanation:
Here the maximum capacity of a tank is **100** litres and the maximum capacity of a bucket is **20** litres.
The amount of water taken in the buckets to fill the water tank are **20**, **15**, **10**, **20**, **20** and **20**.
After pouring the first **5** buckets of water into the tank. The tank contains **85** (20+15+10+20+20) litres of water which is greater than **80%** of its capacity.
Hence the output is 5.

Example Input/Output 2:
Input:
```
150
50
10
20
50
10
10
```

Output:
```
TANK NOT FULL
```

Example Input/Output 3:
Input:
```
100
10
10
10
10
10
10
10
10
10
```

Output:
```
8
```

Max Execution Time Limit: 50 millisecs

19. Metro Train Fare

ID:12237
Solved By 28 Users

TCS NQT

A metro train is a ring route train which runs in a circular fashion. There are **N** stations in the train's route. The names of the N stations and the distance between each station (from the previous station to the current station) are passed as the input to the program. For every **1000** meters, the ticket fare is **X** rupees. The name of the source station, the name of destination station and the value of X are also passed as the input to the program. The program must print the total ticket fare F as the output. The total ticket fare **F** must be ceiled.

Boundary Condition(s):
1 <= N <= 30
Length of each station's name = 2
1 <= Distance between two stations <= 5000
1 <= X <= 25

Input Format:
The first line contains N.
The second line contains the names of the N stations separated by a space.
The third line contains the distance between each station separated by a space.
The fourth line contains the name of the source station and the destination station separated by a space.
The fifth line contains X.

Output Format:
The first line contains the total ticket fare F.

Example Input/Output 1:
Input:
8
AB DK GH KJ OP RS XY TK
600 800 750 900 1600 900 1400 1500
XY KJ
5

Output:
23

Explanation:
The source station is **XY** and the destination station is **KJ**.
The stations present between the source and the destination are **XY**, **TK**, **AB**, **DK**, **GH** and **KJ**.
The distance between **XY** to **TK** is **1500** meters.
The distance between **TK** to **AB** is **600** meters.
The distance between **AB** to **DK** is **800** meters.
The distance between **DK** to **GH** is **750** meters.
The distance between **GH** to **KJ** is **900** meters.
The total distance covered is **4550** meters.
The total ticket cost is **22.75**.
After ceiling the ticket cost, it becomes **23**.
So 23 is printed as the output.

Example Input/Output 2:
Input:
8
AB DK GH KJ OP RS XY TK
600 800 750 900 1600 900 1400 1500
RS XY
10

Output:
14

Example Input/Output 3:
Input:
5
GT XY AK VU MN
1300 710 970 550 920
MN GT
7

Output:
10

Max Execution Time Limit: 50 millisecs

20. Chocolate Vending Machine

ID:12235
Solved By 26 Users

TCS NQT

There is a chocolate vending machine which contains at most **N** chocolates. The chocolate vending machine is never remaining empty as when last **K** chocolates are left, the machine is refilled with new chocolates in such a way that the machine gets full. There are **C** customers numbered from **1** to **C** who want to buy some chocolates from the machine. The program must accept the values of N, K and C integers representing the number of chocolates that the C customers want to buy as the input. After processing each customer's request, the program must print the output based on the following conditions.
- The program must print the string value "**Customer:** " followed by the customer's number.
- If it is possible to sell the chocolates to a customer's request, the program must print the string value "**Number of chocolates sold:** " followed by the number of chocolates sold to the customer. Then the program must print the string value "**Number of chocolates available:** " followed by the number of chocolates remaining in the machine after the purchase.
- Else the program must print the string value "**Invalid**" as the output.

Boundary Condition(s):
1 <= K <= N <= 1000
1 <= C <= 100
1 <= Each integer value <= N

Input Format:
The first line contains N and K separated by a space.

The second line contains C.
The third line contains C integers separated by a space.

Output Format:
For each customer, the first line contains the string value "Customer: " followed by the customer's number.
The second line contains the string value "Invalid" or the string value "Number of chocolates sold: " followed by the number of chocolates sold to the customer as per the given conditions.
The third line contains the string value "Number of chocolates available: " followed by the number of chocolates remaining in the machine after the purchase.

Example Input/Output 1:
Input:

```
10 5
4
3 8 5 6
```

Output:

```
Customer: 1
Number of chocolates sold: 3
Number of chocolates available: 7
Customer: 2
Invalid
Customer: 3
Number of chocolates sold: 5
Number of chocolates available: 10
Customer: 4
Number of chocolates sold: 6
Number of chocolates available: 10
```

Explanation:
Here **N=10** and **K=5**.
Initially the number of chocolates available in the vending machine is 10.
The **customer 1** wants to buy **3** chocolates. It is possible to sell 3 chocolates. Now the number of chocolates sold is **3** and the number of chocolates remaining is **7**.
The **customer 2** wants to buy **8** chocolates. It is not possible to sell 8 chocolates because the vending machine has only **7** chocolates.
The **customer 3** wants to buy **5** chocolates. It is possible to sell 5 chocolates. Now the number of chocolates sold is **5** and the number of chocolates remaining is **2** which is less than K. So the machine is refilled with new chocolates (now the machine has **10** chocolates).
The **customer 4** wants to buy **6** chocolates. It is possible to sell 6 chocolates. Now the number of chocolates sold is **6** and the number of chocolates remaining is **4** which is less than K. So the machine is refilled with new chocolates (now the machine has **10** chocolates).

Example Input/Output 2:
Input:

```
18 11
7
4 1 15 4 5 2 3
```

Output:

```
Customer: 1
Number of chocolates sold: 4
Number of chocolates available: 14
Customer: 2
Number of chocolates sold: 1
Number of chocolates available: 13
Customer: 3
Invalid
Customer: 4
Number of chocolates sold: 4
Number of chocolates available: 18
Customer: 5
Number of chocolates sold: 5
Number of chocolates available: 13
Customer: 6
Number of chocolates sold: 2
Number of chocolates available: 18
Customer: 7
Number of chocolates sold: 3
Number of chocolates available: 15
```

Max Execution Time Limit: 50 millisecs

21. Count Unique Words

ID:12339
Solved By 47 Users

WIPRO

The program must accept a string **S** containing multiple words as the input. The program must print the number of unique words (i.e., the words that are not repeated) present in the string S as the output.

Boundary Condition(s):
1 <= Length of S <= 1000

Input Format:
The first line contains S.

Output Format:
The first line contains the number of unique words present in the string S.

Example Input/Output 1:
Input:
I love to code and I like to participate in coding competitions

Output:
8

Explanation:
The **8** unique words are given below.
love code and like participate in coding competitions

Example Input/Output 2:
Input:
Go Back Welcome back

Output:
4

Max Execution Time Limit: 50 millisecs

22. Barcode Number - Old ID

ID:12340
Solved By 46 Users

WIPRO

The program must accept an integer representing a barcode number of a product containing cosmetics. Each digit in the barcode number represents the ID of the cosmetic item. The oldest cosmetic item is the one with the smallest ID. There can be multiples of the same cosmetic item in the product. The program must print the ID of the oldest cosmetic item in the given product as the output.

Boundary Condition(s):
1 <= N <= 10^8

Input Format:
The first line contains an integer representing a barcode number of a product containing cosmetics.

Output Format:
The first line contains an integer representing the ID of the oldest cosmetic item in the given product.

Example Input/Output 1:
Input:
32957

Output:
2

Explanation:
Here **N = 32957**.
The smallest ID in **32957** is **2**. So it is printed as the output.

Example Input/Output 2:
Input:
98575

Output:
5

Max Execution Time Limit: 50 millisecs

23. Shortest & Second Shortest Distance

ID:12343
Solved By 42 Users

TCS DIGITAL

There are **N** paths to travel from the city **A** to the city **B**. The distance of each path is passed as the input to the program. The program must print the shortest distance and the second shortest distance among the given N distances as the output. If all the distances are equal, then the program must print the string value "**Equal**" as the output.

Boundary Condition(s):
2 <= N <= 100
1 <= Distance of each path <= 10^8

Input Format:
The first line contains N.
The next N lines, each contains an integer value representing the distance of each path from the city A to the city B.

Output Format:
The first line contains the shortest distance and the second shortest distance among the given N distances separated by a space or the first line contains a string value "Equal".

Example Input/Output 1:
Input:
```
5
400
100
500
200
300
```

Output:
```
100 200
```

Explanation:
Here **N=5**, the distance of each path from the city A to the city B are given below.
400 100 500 200 300
The distance **100** is the shortest distance and **200** is the second shortest distance. So 100 200 is printed.

Example Input/Output 2:
Input:
```
4
45
20
65
20
```

Output:
```
20 45
```

Example Input/Output 3:
Input:
```
6
75
```

```
75
75
75
75
75
```

Output:

```
Equal
```

Max Execution Time Limit: 50 millisecs

24. Minimum Profit - N days

ID:12341
Solved By 39 Users

WIPRO

A shop has **M** sales each day for **N** days. Each day different types of items were sold and had different profits associated with them, but the number of items sold on each day was the same. The program must accept the values of M, N and the profits earned from the M sales for N days as the input. The program must print the minimum profit earned on each day as the output.

Boundary Condition(s):
2 <= M, N <= 50
1 <= Profit for each sale <= 1000

Input Format:
The first line contains M and N separated by a space.
The next M lines, each contains N integers separated by a space.

Output Format:
The first line contains the minimum profit earned on each day separated by a space.

Example Input/Output 1:
Input:

```
5 3
5 7 10
2 14 6
1 8 11
3 9 15
12 4 7
```

Output:
1 4 6

Explanation:
On the first day, the profits earned from the **5** sales are **5**, **2**, **1**, **3** and **12**. Here **1** is the minimum profit, so 1 is printed.
On the second day, the profits earned from the **5** sales are **7**, **14**, **8**, **9** and **4**. Here **4** is the minimum profit, so 4 is printed.
On the third day, the profits earned from the **5** sales are **10**, **6**, **11**, **15** and **7**. Here **6** is the minimum profit, so 6 is printed.
Hence the output is
1 4 6

Example Input/Output 2:
Input:
2 5
25 20 18 24 19
21 11 23 17 15

Output:
21 11 18 17 15

Max Execution Time Limit: 50 millisecs

25. Fully Automatic Vending Machine

ID:12238
Solved By 31 Users

TCS NQT

A fully automatic vending machine can serve the range of products as follows.

Coffee:
1. Espresso Coffee
2. Cappuccino Coffee
3. Latte Coffee

Tea:
1. Plain Tea
2. Assam Tea
3. Ginger Tea

4. Cardamom Tea
5. Masala Tea
6. Lemon Tea
7. Green Tea
8. Organic Darjeeling Tea

Soups:
1. Hot and Sour Soup
2. Veg Corn Soup
3. Tomato Soup
4. Spicy Tomato Soup

Beverages:
1. Hot Chocolate Drink
2. Badam Drink
3. Badam-Pista Drink

A customer has to enter two keys for the menu **M** and the sub-menu **S**. The key for selecting the menu is as follows.
C or **c** - **Coffee**
T or **t** - **Tea**
S or **s** - **Soup**
B or **b** - **Beverages**
The second key is the product number.
The program must accept the values of M and S as the input. The program must print the output based on the following conditions.
- If the customer has entered the right keys, the program must print the string value "**Enjoy Your** " followed by the chosen product name.
- Else the program must print the string value "**Invalid Option!**" as the output.

Input Format:
The first line contains M.
The second line contains S.

Output Format:
The first line contains the string value "Enjoy Your " followed by the chosen product name or the string value "Invalid Option!" as per the given conditions.

Example Input/Output 1:
Input:
```
t
4
```

Output:
```
Enjoy Your Cardamom Tea
```

Explanation:
The customer selects '**t**' in the menu and **4** in the sub menu. The chosen menu and sub-menu are valid.
Hence the output is
Enjoy Your Cardamom Tea

Example Input/Output 2:
Input:
B
5

Output:
Invalid Option!

Example Input/Output 3:
Input:
K
1

Output:
Invalid Option!

Max Execution Time Limit: 50 millisecs

26. Cricket Tournament

ID:12254
Solved By 30 Users

TCS NQT

The program must accept the list of team names participating in a cricket tournament. The character **q** or **Q** in the input represents the end of the team list. Each team must play exactly one game against the other teams. The program must print the string value "**Total Matches:** " followed by the total number of matches possible excluding the semi-final and the final. Then the program must print the pairing of teams in the order of their occurrence as the output.

Boundary Condition(s):
3 <= Number of teams <= 12

Input Format:
The lines, each contains a string value representing the team name participating in a cricket tournament.

Output Format:
The first line contains the string value "Total Matches: " followed by the total number of matches possible excluding the semi-final and the final.
The lines from the 2nd line containing the pairing of teams in the order of their occurrence

Example Input/Output 1:
Input:
Chennai
Mumbai
Delhi
Kolkata
Q

Output:
Total Matches: 6
Chennai vs Mumbai
Chennai vs Delhi
Chennai vs Kolkata
Mumbai vs Delhi
Mumbai vs Kolkata
Delhi vs Kolkata

Explanation:
The given **4** teams are **Chennai**, **Mumbai**, **Delhi** and **Kolkata**.
Total Matches: 6
The pairing of teams in the order of their occurrence are given below.
Chennai vs Mumbai
Chennai vs Delhi
Chennai vs Kolkata
Mumbai vs Delhi
Mumbai vs Kolkata
Delhi vs Kolkata

Example Input/Output 2:
Input:
Mumbai
Chennai
Hyderabad
Bangalore
Kolkata
q

Output:

```
Total Matches: 10
Mumbai vs Chennai
Mumbai vs Hyderabad
Mumbai vs Bangalore
Mumbai vs Kolkata
Chennai vs Hyderabad
Chennai vs Bangalore
Chennai vs Kolkata
Hyderabad vs Bangalore
Hyderabad vs Kolkata
Bangalore vs Kolkata
```

Max Execution Time Limit: 50 millisecs

27. Count & Remove Vowels

ID:12255
Solved By 28 Users

TCS NQT

The program must accept a space separated string **S** as the input. The program must print the output based on the following conditions.
- The program must print the string value "**a:** " followed by the number of occurrences of **a** and **A** in the string S.
- The program must print the string value "**e:** " followed by the number of occurrences of **e** and **E** in the string S.
- The program must print the string value "**i:** " followed by the number of occurrences of **i** and **I** in the string S.
- The program must print the string value "**o:** " followed by the number of occurrences of **o** and **O** in the string S.
- The program must print the string value "**u:** " followed by the number of occurrences of **u** and **U** in the string S.
- Finally, the program must print the string S after removing the vowels. If the string S becomes empty after removing the vowels, the program must print **-1** as the output.

Boundary Condition(s):
1 <= Length of S <= 1000

Input Format:
The first line contains S.

Output Format:
The first line contains the string value "a: " followed by the number of occurrences of 'a' and 'A' in the string S.
The second line contains the string value "e: " followed by the number of occurrences of 'e' and 'E' in the string S.
The third line contains the string value "i: " followed by the number of occurrences of 'i' and 'I' in the string S.
The fourth line contains the string value "o: " followed by the number of occurrences of 'o' and 'O' in the string S.
The fifth line contains the string value "u: " followed by the number of occurrences of 'u' and 'U' in the string S.
The sixth line contains the modified string S after removing the vowels or -1.

Example Input/Output 1:
Input:

```
Hello! Welcome to SkillRack
```

Output:

```
a: 1
e: 3
i: 1
o: 3
u: 0
Hll! Wlcm t SkllRck
```

Explanation:
Here the given string is **Hello! Welcome to SkillRack**
The count of occurrences of '**a**' and '**A**' in the given string is **1**. So **a: 1** is printed.
The count of occurrences of '**e**' and '**E**' in the given string is **3**. So **e: 3** is printed.
The count of occurrences of '**i**' and '**I**' in the given string is **1**. So **i: 1** is printed.
The count of occurrences of '**o**' and '**O**' in the given string is **3**. So **o: 3** is printed.
The count of occurrences of '**u**' and '**U**' in the given string is **0**. So **u: 0** is printed.
After removing the vowels, the string becomes **Hll! Wlcm t SkllRck**. So Hll! Wlcm t SkllRck is printed.

Example Input/Output 2:
Input:

```
Aei ouaE IOU
```

Output:

```
a: 2
e: 2
i: 2
o: 2
u: 2
-1
```

Example Input/Output 3:
Input:

```
Abcd eeee II OOO uuu aBCD
```

Output:

```
a: 2
e: 4
i: 2
o: 3
u: 3
bcd BCD
```

Max Execution Time Limit: 50 millisecs

28. Team Selection - Height

ID:12268
Solved By 23 Users

TCS NQT

The program must accept the name and the height (in feet) of the students present in a class. The class teacher wants to form a team of **T** students from the class. The team selection criteria for the game is height (i.e., the first T tallest students will be selected). If two or more students have the same height, then priority is given to the order of their occurrence. The program must print the names of the selected students in the order of their occurrence as the output. The character **q** or **Q** in the input represents the end of the student name and height list.

Boundary Condition(s):
1 <= T <= Number of students in the class <= 30
1 <= Length of each student's name <= 20
2 <= Height <= 7

Input Format:
The first line contains T.
The lines, each contains a string value and a floating point value representing the name and the height of a student.
The last line contains a character 'q' or 'Q' represents the end of the student name and height list.

Output Format:
The first T lines, each contains a string value representing the name of the selected students.

Example Input/Output 1:

Input:

```
4
Abcd 5.6
Kaakoo 5.5
Yoogu 4.9
Meegkka 4.5
Tikkku 5.5
Sinnsoo 4.9
q
```

Output:

```
Abcd
Kaakoo
Yoogu
Tikkku
```

Explanation:

Here **T = 4**, the name and the height of the students are given below.

Abcd 5.6
Kaakoo 5.5
Yoogu 4.9
Meegkka 4.5
Tikkku 5.5
Sinnsoo 4.9

The first **4** tallest students in the order of their occurrence are given below.

Abcd
Kaakoo
Yoogu
Tikkku

Example Input/Output 2:

Input:

```
5
Abc 5.6
xyyzzz 5.1
Ijkk 5.9
pppqr 6.1
Raaath 5.6
caaarl 5.7
maarq 5.7
Tood 5.5
Q
```

Output:

```
Abc
```

Ijkk
pppqr
caaarl
maarq

Max Execution Time Limit: 50 millisecs

29. Smallest Integer - Product of Digits

ID:12342
Solved By 17 Users

TCS DIGITAL

The program must accept an integer **N** as the input. The program must print the smallest integer **X** such that the product of all the digits in X is equal to N. If it is not possible to form the integer X, the program must print **-1** as the output.

Boundary Condition(s):
1 <= N <= 10^8

Input Format:
The first line contains N.

Output Format:
The first line contains X.

Example Input/Output 1:
Input:
10

Output:
25

Explanation:
Here N = 10.
The smallest possible integer X is **25**. The product of **2** and **5** is **10**.
So **25** is printed as the output.

Example Input/Output 2:
Input:
256

Output:
488

Example Input/Output 3:
Input:
85

Output:
-1

Max Execution Time Limit: 50 millisecs

30. Exam Seating Arrangement

ID:12239
Solved By 12 Users

TCS NQT

At an exam center, **M** students are allocated for one classroom as per the School Rules. The examination staff / invigilater makes seating arrangements where the classroom contains N benches arranged in columns separated by an equal distance (Each column contains **N** benches). The examination is conducted for two different standards (**X** and **XII**). The students from the standard X can sit one after the other in the order of their Registration Number in the odd columns (1, 3, 5, 7 and so on). The students from standard XII can sit one after the other in the order of their Registration Number in the even columns (2, 4, 6, 8 and so on). The registration number of students are in continuous range with no drop, and the standard name (X or XII) should be prefixed. All students enter in the classroom in a random order. Few students may remain absent.

The program must accept the registration numbers of the students who have entered the classroom as the input. The program must print the status of seating arrangement at the exam time as the output. "**ABSENT**" should be marked at the place of the Registration Numbers of the absentees.

Note: The value of M is always divisible by N.

Boundary Condition(s):
1 <= N <= M <= 100

Input Format:
The first line contains M and N separated by a space.

The line(s), each contains a string value representing the registration number of a student who entered the classroom.

Output Format:
The first N lines, each contains M/N string values separated by a space representing the status of seating arrangement at the exam time.

Example Input/Output 1:
Input:

```
10 5
X001
X002
XII002
XII005
X005
X003
XII004
XII001
X004
```

Output:

```
X001 XII001
X002 XII002
X003 ABSENT
X004 XII004
X005 XII005
```

Example Input/Output 2:
Input:

```
12 4
X001
X002
X003
X006
XII002
XII003
X007
X008
X004
X005
XII001
XII004
```

Output:

```
X001 XII001 X005
X002 XII002 X006
```

```
X003 XII003 X007
X004 XII004 X008
```

Max Execution Time Limit: 50 millisecs

31. Count Characters & Words

ID:12283
Solved By 43 Users

TCS NQT

The program must accept a string **S** as the input. The program must print the number of characters present in the string S. Then the program must print the number of words present in the string S as the output.

Boundary Condition(s):
1 <= Length of S <= 100

Input Format:
The first line contains S.

Output Format:
The first line contains the number of characters present in the string S.
The second line contains the number of words present in the string S.

Example Input/Output 1:
Input:

```
Apple orange banana
```

Output:

```
19
3
```

Explanation:
Here the given string is **Apple orange banana**.
The length of the given string is **19**. So 19 is printed.
There are **3** words in the string. So 3 is printed.

Example Input/Output 2:
Input:

```
Note pen Pencil BOOK ink
```

Output:
24
5

Max Execution Time Limit: 50 millisecs

32. Fibonacci Series - Count Odd & Even

ID:12276
Solved By 33 Users

TCS NQT

The program must accept an integer **N** as the input. The program must print the Fibonacci series until the given integer N. Then the program must print the number of odd integers present in the Fibonacci series as the output. Finally, the program must print the number of even integers (except 0) present in the Fibonacci series as the output.

Boundary Condition(s):
2 <= N <= 10^8

Input Format:
The first line contains N.

Output Format:
The first line contains the Fibonacci series until the given integer N.
The second line contains the number of odd integers present in the Fibonacci series.
The third line contains the number of even integers present in the Fibonacci series.

Example Input/Output 1:
Input:
25

Output:
0 1 1 2 3 5 8 13 21
6
2

Explanation:
Here **N=25**, the Fibonacci series until **25** is given below.
0 1 1 2 3 5 8 13 21
The number of odd integers in the series is **6** and the number of even integers (except 0) in the

series is **2.**
Hence the output is
0 1 1 2 3 5 8 13 21
6
2

Example Input/Output 2:
Input:
89

Output:
0 1 1 2 3 5 8 13 21 34 55 89
8
3

Max Execution Time Limit: 50 millisecs

33. Bunch of Room Keys

ID:12269
Solved By 23 Users

TCS NQT

In a lodge, there is a bunch of room keys having key number stamped on each key. There may be few duplicate keys in the bunch (i.e., more than one key with the same number). A room can be opened with a key which must have the room's number. There is a rare chance of blank keys (**-**) which are the keys without numbers, and no room can be opened with those keys. The program must accept the list of key numbers available in the bunch as the input. The character **q** or **Q** in the input represents the end of the key numbers list. The program must print the output based on the following conditions.
- The program must print the string value "**Blank Keys:** " followed by the number of blank keys in the bunch.
- Then the program must print the string value "**Total Keys:** " followed by the total number of keys available in the bunch.
- Finally, the program must print the string value "**Number of rooms:** " followed by the total number of rooms available in the lodge as the output.

Boundary Condition(s):
1 <= Length of each key number <= 5

Input Format:
The lines, each contains a string value representing the key numbers available in the bunch.
The last line contains the character 'q' or 'Q' represents the end of the key numbers list.

Output Format:
The first line contains the string value "Blank Keys: " followed by the number of blank keys in the bunch.
The second line contains the string value "Total Keys: " followed by the total number of keys available in the bunch.
The third line contains the string value "Number of rooms: " followed by the total number of rooms available in the lodge.

Example Input/Output 1:
Input:

```
A101
A103
A102
B202
-
A102
A101
B201
B203
-
B203
Q
```

Output:

```
Blank Keys: 2
Total Keys: 11
Number of rooms: 6
```

Explanation:
The number of blank keys in the list of key numbers is **2**. So **Blank Keys: 2** is printed.
There are **9** numbered keys and **2** blank keys, the total number of keys in the bunch is **11**. So **Total Keys: 11** is printed.
The number of unique key numbers in the list is **6**. So **Number of rooms: 6** is printed.

Example Input/Output 2:
Input:

```
A100
B205
B103
C115
C100
-
```

```
B205
A100
-
B205
-
C114
q
```

Output:

```
Blank Keys: 3
Total Keys: 12
Number of rooms: 6
```

Max Execution Time Limit: 50 millisecs

34. Bunch of Room Keys

ID:12269
Solved By 23 Users

TCS NQT

In a lodge, there is a bunch of room keys having key number stamped on each key. There may be few duplicate keys in the bunch (i.e., more than one key with the same number). A room can be opened with a key which must have the room's number. There is a rare chance of blank keys (**-**) which are the keys without numbers, and no room can be opened with those keys. The program must accept the list of key numbers available in the bunch as the input. The character **q** or **Q** in the input represents the end of the key numbers list. The program must print the output based on the following conditions.
- The program must print the string value "**Blank Keys:** " followed by the number of blank keys in the bunch.
- Then the program must print the string value "**Total Keys:** " followed by the total number of keys available in the bunch.
- Finally, the program must print the string value "**Number of rooms:** " followed by the total number of rooms available in the lodge as the output.

Boundary Condition(s):
1 <= Length of each key number <= 5

Input Format:
The lines, each contains a string value representing the key numbers available in the bunch.
The last line contains the character 'q' or 'Q' represents the end of the key numbers list.

Output Format:
The first line contains the string value "Blank Keys: " followed by the number of blank keys in the bunch.
The second line contains the string value "Total Keys: " followed by the total number of keys available in the bunch.
The third line contains the string value "Number of rooms: " followed by the total number of rooms available in the lodge.

Example Input/Output 1:
Input:
```
A101
A103
A102
B202
-
A102
A101
B201
B203
-
B203
Q
```

Output:
```
Blank Keys: 2
Total Keys: 11
Number of rooms: 6
```

Explanation:
The number of blank keys in the list of key numbers is **2**. So **Blank Keys: 2** is printed.
There are **9** numbered keys and **2** blank keys, the total number of keys in the bunch is **11**. So **Total Keys: 11** is printed.
The number of unique key numbers in the list is **6**. So **Number of rooms: 6** is printed.

Example Input/Output 2:
Input:
```
A100
B205
B103
C115
C100
-
B205
A100
-
B205
```

```
-
C114
q
```

Output:

```
Blank Keys: 3
Total Keys: 12
Number of rooms: 6
```

Max Execution Time Limit: 50 millisecs

35. Most Frequent Alphabet - Substrings

ID:12355
Solved By 15 Users

CTS GenCNext

The program must accept a string **S** containing only lower case alphabets and **Q** queries as the input. Each query contains two integers representing the starting and the ending indices of a substring in S. For each query, the program must print the most frequently occurring alphabet in the specified substring. If two or more alphabets have the same frequency, then the program must print the alphabet that is least in the alphabetical order.

Boundary Condition(s):
2 <= Length of S <= 1000
1 <= Q <= 10^5

Input Format:
The first line contains S.
The second line contains Q.
The next Q lines, each containing two integers representing the starting and the ending indices of a substring in S.

Output Format:
The first Q lines, each containing the most frequently occurring alphabet in the specified substring.

Example Input/Output 1:
Input:

```
badbadbed
4
```

0 8
1 4
0 5
2 7

Output:
b
a
a
b

Explanation:
Here the given string **S** is **badbadbed** and the number of queries **Q** is **4**.
For **1**st query, the substring is **badbadbed**. The alphabets **b** and **d** occur the same maximum number of times (**3 times**). So the alphabet that is least in the alphabetical order **b** is printed.
For **2**nd query, the substring is **adba**. The alphabet **a** occurs the maximum number of times (**2 times**). So **a** is printed.
For **3**rd query, the substring is **badbad**. The alphabets **b**, **a** and **d** occur the same maximum number of times (**2 times**). So the alphabet that is least in the alphabetical order **a** is printed.
For **4**th query, the substring is **dbadbe**. The alphabets **d** and **b** occur the same maximum number of times (**2 times**). So the alphabet that is least in the alphabetical order **b** is printed.
Hence the output is
b
a
a
b

Example Input/Output 2:
Input:
comeonmonsoon
7
1 6
4 8
8 12
5 10
1 12
2 5
7 9

Output:
m
n
n
n
o

e
n

Max Execution Time Limit: 50 millisecs

36. Customer & Queries

ID:12366
Solved By 13 Users

CTS GenCNext

In a shop, there are C customers stand in a queue to buy some items. Each customer has a unique ID (from **1** to **C**). The program must accept the value of **C** and **Q** queries as the input. Each query contains 3 integers, where the 1st integer represents the query type. The query type can be any one of the following.
- **Query type 1:** The 2nd and 3rd integers representing the customer ID and the number of items purchased by the customer.
- **Query type 2:** The 2nd and 3rd integers representing the starting and ending customer ID range.
The program must design a system that handles all the transaction queries based on the query type.
- Query type 1: If the customer has already purchased some items, it adds the quanity of the new sale to the previous quantity and updates the record. Else it will make a fresh entry.
- Query type 2: It prints total number of items purchased by the given range of the customer ID.
Note: At least one query of type 2 is always present in the given Q queries.

Boundary Condition(s):
2 <= C <= 50
1 <= Q <= 1000

Input Format:
The first line contains C and Q separated by a space.
The next Q lines, each containing three integers separated by a space.

Output Format:
The first line contains the integer value(s) separated by a space.

Example Input/Output 1:
Input:
4 5
1 4 10
2 1 3

1 2 4
1 4 2
2 2 4

Output:
0 16

Explanation:
Here **C = 4** (number of customers) and **Q = 5** (number of queries).
1st query - The type is **1**. It makes a fresh entry for the customer ID 4.
4 : 10
2nd query - The type is **2**. The total number of items purchased by the given range of the customer ID (**1** to **3**) is **0**.
So **0** is printed as the output.
3rd query - The type is **1**. It makes a fresh entry for the customer ID 2.
2 : 4
4th query - The type is **1**. It updates the record of the customer ID 4.
4 : 12
5th query - The type is **2**. The total number of items purchased by the given range of the customer ID (**2** to **4**) is **16**.
So **16** is printed as the output.

Example Input/Output 2:
Input:
8 6
1 6 15
1 1 28
2 2 6
1 2 70
2 1 5
2 1 6

Output:
15 98 113

Max Execution Time Limit: 50 millisecs

37. Walls Reconstruction - Strictly Increasing Order

ID:12284
Solved By 13 Users

TCS NQT

There are **N** walls constructed in increasing order of height. A mason wants to reconstruct the walls with minimal changes in height so that the height of the N walls are in strictly increasing order. The program must accept N integers representing the height of the N walls as the input. The program must print the total height of the N walls after the reconstruction as the output.

Boundary Condition(s):
2 <= N <= 100
1 <= Each integer value <= 1000

Input Format:
The first line contains N.
The second line contains N integers separated by a space.

Output Format:
The first line contains the total height of the N walls after the reconstruction.

Example Input/Output 1:
Input:
5
2 2 2 5 8

Output:
22

Explanation:
Here N = 5, the heights of the 5 walls are 2, 2, 2, 5 and 8.
After reconstruction of the 2nd wall and 3rd wall (with minimal changes in height), the heights of the 5 walls become 2, **3**, **4**, 5 and 8.
Now the heights of the 5 walls are in strictly increasing order.
The total height of the 5 walls after the reconstruction is **22** (2+3+4+5+8).
So 22 is printed as the output.

Example Input/Output 2:
Input:
6
1 3 3 3 3 4

Output:
26

Max Execution Time Limit: 50 millisecs

38. Count Submatrices - Product K

ID:12373
Solved By 7 Users

CTS GenCNext

The program must accept an integer matrix of size **RxC** and an integer **K** as the input. The product of a submatrix in the given matrix is the product of all the elements present the submatrix. The program must print the number of submatrices that contain the top-left element of the given matrix with a product less than or equal to K.
Note: The size of the each submatrix must be **MxN**, where 2 <= M <= R and 2 <= N <= C.

Boundary Condition(s):
2 <= R, C <= 10
1 <= Matrix element value <= 9
1 <= K <= 10^18

Input Format:
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.
The $(R+2)^{nd}$ line contains K.

Output Format:
The first line contains the number of submatrices that contain the top-left element of the given matrix with a product less than or equal to K.

Example Input/Output 1:
Input:
3 4
1 2 4 1
6 3 2 2
4 1 3 2
100

Output:
7

Explanation:
The top-left element of the given matrix is 1.
The **7** possible submatrices are given below.
1 2

6 3
The product of the above submatrix is **36** which is less than or equal to 100.
4 1
2 2
The product of the above submatrix is **16** which is less than or equal to 100.
2 4 1
3 2 2
The product of the above submatrix is **96** which is less than or equal to 100.
4 1
2 2
3 2
The product of the above submatrix is **96** which is less than or equal to 100.
6 3
4 1
The product of the above submatrix is **72** which is less than or equal to 100.
3 2
1 3
The product of the above submatrix is **18** which is less than or equal to 100.
3 2 2
1 3 2
The product of the above submatrix is **72** which is less than or equal to 100.

Example Input/Output 2:
Input:
4 5
3 1 6 4 2
6 4 1 1 4
6 2 9 6 6
7 6 2 6 4
950

Output:
3

Max Execution Time Limit: 50 millisecs

39. Count Submatrices - Product K

ID:12373
Solved By 7 Users

CTS GenCNext

The program must accept an integer matrix of size **RxC** and an integer **K** as the input. The product of a submatrix in the given matrix is the product of all the elements present the submatrix. The program must print the number of submatrices that contain the top-left element of the given matrix with a product less than or equal to K.
<u>**Note:**</u> The size of the each submatrix must be **MxN**, where 2 <= M <= R and 2 <= N <= C.

Boundary Condition(s):
2 <= R, C <= 10
1 <= Matrix element value <= 9
1 <= K <= 10^18

Input Format:
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.
The $(R+2)^{nd}$ line contains K.

Output Format:
The first line contains the number of submatrices that contain the top-left element of the given matrix with a product less than or equal to K.

Example Input/Output 1:
Input:
3 4
1 2 4 1
6 3 2 2
4 1 3 2
100

Output:
7

Explanation:
The top-left element of the given matrix is 1.
The **7** possible submatrices are given below.
1 2
6 3
The product of the above submatrix is **36** which is less than or equal to 100.
4 1
2 2
The product of the above submatrix is **16** which is less than or equal to 100.
2 4 1
3 2 2
The product of the above submatrix is **96** which is less than or equal to 100.
4 1
2 2
3 2

The product of the above submatrix is **96** which is less than or equal to 100.
6 3
4 1
The product of the above submatrix is **72** which is less than or equal to 100.
3 2
1 3
The product of the above submatrix is **18** which is less than or equal to 100.
3 2 2
1 3 2
The product of the above submatrix is **72** which is less than or equal to 100.

Example Input/Output 2:
Input:

```
4 5
3 1 6 4 2
6 4 1 1 4
6 2 9 6 6
7 6 2 6 4
950
```

Output:

```
3
```

Max Execution Time Limit: 50 millisecs

40. Area of Largest House

ID:12356
Solved By 6 Users

CTS GenCNext

The program must accept an integer matrix of size **RxC** representing a residential area. The matrix contains only **0s** and **1s**. If a cell contains some part of a house roof, then it is assigned the value **1**. If a cell represents a vacant plot, then it is assigned the value **0**. A group of adjacent cells (vertically and horizontally) with value 1 represents a single house. The area of a house is equal to the number of 1s present in it. The program must print the area of the largest house in the given residential area as the output.
Note: At least one cell contains 1 in the given matrix.

Boundary Condition(s):
2 <= R, C <= 50

Input Format:
The first line contains R and C separated by a space.
The next R lines, each contains C integers separated by a space.

Output Format:
The first line contains the area of the largest house in the given residential area.

Example Input/Output 1:
Input:
5 5
0 1 0 1 0
0 0 1 1 0
0 1 0 1 1
1 1 0 0 0
1 1 0 1 1

Output:
5

Explanation:
The **2** house with the largest area is highlighted below.
0 1 0 **1** 0
0 0 **1 1** 0
0 **1** 0 **1 1**
1 1 0 0 0
1 1 0 1 1
Here the area of the largest house is **5**, so 5 is printed as the output.

Example Input/Output 2:
Input:
3 5
1 1 0 1 1
1 1 1 0 1
1 0 1 0 1

Output:
7

Max Execution Time Limit: 50 millisecs

41. Nuclear Reaction - Total Energy

ID:12511

Solved By 35 Users

CTS GenCNext

In a chemistry lab, the combination of two nuclear chemicals produces initial energy as **E**. This energy E changes at a consistent rate **R** every second. The scientist wishes to calculate the total energy produced if the reaction is allowed to occur for **T** seconds. The program must accept the values of E, R and T as the input. The program must print the total energy produced as the output.

Boundary Condition(s):
1 <= E <= 10^5
1 <= R, T <= 1000

Input Format:
The first line contains three integer values separated by a space.

Output Format:
The first line the total energy produced.

Example Input/Output 1:
Input:
6 4 4

Output:
48

Explanation:
For **T = 1**, the initial energy **6** is generated.
For **T = 2**, a consistent rate of **4** is added. So it becomes **10**.
For **T = 3**, again **4** is added to the previous energy value. So it becomes **14**.
For **T = 4**, again **4** is added to the previous energy value. So it becomes **18**.
The total energy produced after **4** seconds is **48** (6 + 10 + 14 + 18).

Example Input/Output 2:
Input:
2 3 5

Output:
40

Max Execution Time Limit: 50 millisecs

42. Flower Sticks - Bouquet

ID:12501
Solved By 33 Users

CTS GenCNext

There are **N** flower sticks to make a bouquet. The first **K** flower sticks must be arranged in increasing order and the remaining flower sticks must be arranged in decreasing order. The program must accept N integers representing the heights of the N flower sticks and the value of K as the input. The program must print the heights of the N flower sticks after making the bouquet.

Boundary Condition(s):
1 <= K <= N <= 100
1 <= Each integer value <= 10^5

Input Format:
The first line contains N.
The second line contains N integers separated by a space.
The third line contains K.

Output Format:
The first line contains the heights of the N flower sticks after making the bouquet.

Example Input/Output 1:
Input:
10
90 22 92 42 6 24 30 58 50 29
6

Output:
6 22 24 42 90 92 58 50 30 29

Explanation:
Here **K=6** and the given **10** integers are **90 22 92 42 6 24 30 58 50 29**
After arranging the first **6** flower sticks in increasing order and the remaining **4** flower sticks in decreasing order, the height of the 10 flower sticks become
6 22 24 42 90 92 58 50 30 29

Example Input/Output 2:
Input:
8
10 9 8 7 1 2 3 4

Output:
7 8 9 10 4 3 2 1

Max Execution Time Limit: 50 millisecs

43. function findOddEvenDifference

ID:12353
Solved By 29 Users

Tech Mahindra

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all the test cases. Do not write the main() function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **findOddEvenDifference** accepts two arguments - **SIZE** and **arr**, an integer representing the size of the array and an array of integers arr.

The function/method **findOddEvenDifference** must return an integer representing the difference between the sum of odd integers and even integers in the given array.

Your task is to implement the function **findOddEvenDifference** so that it passes all the test cases.

Example Input/Output 1:
Input:
5
20 10 43 45 26

Output:
32

Explanation:
Here **N=5** and the given **5** integers are **20 10 43 45 26**.
The sum of odd integers is **88** (43 + 45).
The sum of even integers is **56** (20 + 10 + 26).

The difference between the sum of odd integers and even integers is **32** (88 - 56).
Hence the output is 32.

Example Input/Output 2:
Input:
3
11 9 15

Output:
35

Example Input/Output 3:
Input:
4
40 5 20 11

Output:
-44

Max Execution Time Limit: 50 millisecs

44. Chemical Mixture - Explosion Rate

ID:12513
Solved By 25 Users

CTS GenCNext

In a high energy materials research laboratory, there is a stand containing two rows. Each row has **N** chemicals. Each chemical has some explosion rate. The scientists have decided to combine the two chemicals to create a very powerful bomb. The explosion rate of the mixture is the product of their respective explosion rates. The way of mixing the chemicals is given below.
- The scientists reverse the positions of the chemicals in the second row.
- Then they mix each chemical from the first row with the corresponding chemical in the second row.
- If the explosion rate of a mixture is negative, they will not mix those two chemicals.
The program must accept the explosion rates of the 2*N chemicals as the input. The program must print the sum of the explosion rates of all the mixtures as the output.

Boundary Condition(s):
2 <= N <= 1000
-10^6 <= Explosion rate of each chemical <= 10^6

Input Format:
The first line contains N.
The next two lines, each containing N integers separated by a space.

Output Format:
The first line contains the sum of the explosion rates of all the mixtures.

Example Input/Output 1:
Input:
5
5 1 -5 4 -6
-2 9 5 -8 2

Output:
58

Explanation:
Here **N = 5**.
The explosion rates of the 5 chemicals in the first row are **5 1 -5 4 -6**.
The explosion rates of the 5 chemicals in the second row are **-2 9 5 -8 2**.
After reversing the positions of the chemicals in the second row, they become **2 -8 5 9 -2**.
The 5 possible mixtures and their explosion rate are given below.
5 * 2 = **10**
1 * -8 = **-8**
-5 * 5 = **-25**
4 * 9 = **36**
-6 * -2 = **12**
The 2nd and 3rd mixtures give the negative explosion rate. So they are not considered.
The total explosion rate = **58** (10 + 36 + 12).
So 58 is printed as the output.

Example Input/Output 2:
Input:
6
7 2 12 -6 7 -1
-3 2 -10 9 -4 -2

Output:
185

Max Execution Time Limit: 50 millisecs

45. function findMaxDifference

ID:12350
Solved By 23 Users

Tech Mahindra

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all test cases. Do not write the main() function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **findMaxDifference** accepts two arguments - **SIZE** and **arr**, an integer representing the size of the integer array and the array arr.

The function/method **findMaxDifference** must return an integer representing the maximum difference between the adjacent integers in the given array arr.

Your task is to implement the function **findMaxDifference** so that it passes all the test cases.

Example Input/Output 1:
Input:
5
11 12 8 13 15

Output:
4

Explanation:
The difference between **11** and **12** is **-1**.
The difference between **12** and **8** is **4**.
The difference between **8** and **13** is **-5**.
The difference between **13** and **15** is **-2**.
The maximum difference is **4**, which is printed as the output.

Example Input/Output 2:
Input:
6
-13 -4 0 -1 3 -2

Output:
5

Max Execution Time Limit: 50 millisecs

46. function calculateTotalTax

ID:12351
Solved By 22 Users

Tech Mahindra

You are required to complete the given code. You can click on Run anytime to check the compilation/execution status of the program. You can use printf to debug your code. The submitted code should be logically/syntactically correct and pass all the test cases. Do not write the main() function as it is not required.

Code Approach: For this question, you will need to complete the code as in the given implementation. We do not expect you to modify the approach.

The function/method **calculateTotalTax** accepts two arguments - **B** and **billingAmounts**, an integer representing the number of bills and an array of integers representing a list of B billing amounts.

The function/method **calculateTotalTax** must return an integer representing the total tax amount for the given B billing amounts.

Up to the amount 1000, there is no tax applicable and then, a flat tax of **10%** is applicable for the remaining amount as per the tax rate.

Note: Each billing amount is always a multiple of **10**.

Your task is to implement the function **calculateTotalTax** so that it passes all the test cases.

Example Input/Output 1:
Input:
4
4000 2000 1000 5000

Output:
800

Explanation:
For the **1st** bill **4000**, 10% tax is applicable for **3000**. So the tax is **300**.
For the **2nd** bill **2000**, 10% tax is applicable for **1000**. So the tax is **100**.
For the **3rd** bill **1000**, there will be no tax.

For the **4th** bill **5000**, 10% tax is applicable for **4000**. So the tax is **400**.
The total tax amount is **800** (300+100+400).

Example Input/Output 2:
Input:
7
1900 290 240 1100 2500 830 500

Output:
250

Max Execution Time Limit: 50 millisecs

47. Possible Ways - Consecutive 1s

ID:12500
Solved By 7 Users

CTS GenCNext

The program must accept a string **S** and an integer **X** as the input. The string S contains only **0s** and **1s**. The integer X represents the number of changes allowed to modify the string S. The program must print an integer representing the number of ways possible to get the longest consecutive 1s in the string S with exactly X changes.
Note: The value of X is always less than or equal to the number of 0s in the string S.

Boundary Condition(s):
2 <= Length of S <= 1000

Input Format:
The first line contains S.
The second line contains X.

Output Format:
The first line contains the number of ways possible to get the longest consecutive 1s in the string S with exactly X changes.

Example Input/Output 1:
Input:
01010101
1

Output:
3

Explanation:
Here **X = 1**. Only one change is allowed.
All possible combinations are given below.
11010101 (2 consecutive 1s)
0**111**0101 (3 consecutive 1s)
010**111**01 (3 consecutive 1s)
01010**111** (3 consecutive 1s)
There are **3** ways to get the longest consecutive 1s.
So 3 is printed as the output.

Example Input/Output 2:
Input:
010110001001
2

Output:
2

Max Execution Time Limit: 50 millisecs

48. Digits Sum Starts with D

ID:12541
Solved By 6 Users

WIPRO

The program must accept two integers **N** and **D** as the input. The program must form a series of integers based on the following conditions.
- All integers in the series must be sorted in ascending order.
- The sum of digits in each integer must start with the given digit D.
- All integers in the series must be from 1 to N.
Finally, the program must print the integers in the series as the output.

Boundary Condition(s):
1 <= N <= 10^7
1 <= D <= 9

Input Format:
The first line contains N and D separated by a space.

Output Format:
The first line contains the integers in the series separated by a space.

Example Input/Output 1:
Input:
500 2

Output:
2 11 20 101 110 200 299 389 398 399 479 488 489 497 498 499

Explanation:
Here N = **500** and D = **2**.
So the sum of the digits in each integer must start with **2**.
The integers in the series from **1 to 500** are given below.
2 11 20 101 110 200 299 389 398 399 479 488 489 497 498 499

Example Input/Output 2:
Input:
109 1

Output:
1 10 19 28 29 37 38 39 46 47 48 49 55 56 57 58 59 64 65 66 67 68 69 73 74 75 76 77 78 79 82 83 84 85 86 87 88 89 91 92 93 94 95 96 97 98 99 100 109

Max Execution Time Limit: 50 millisecs

49. Smart Undercover Agent

ID:12502
Solved By 5 Users

CTS GenCNext

A smart undercover agent sent a message to his country's military commander. The message consists of two lines. The first line contains a text **T** and the second line contains the name of the leader **N** of a terrorist camp. The total number of terrorists in the camp is the number of times the leader's name occurs in the first line. If the given message is invalid (i.e., if it is not possible to find the number of terrorists in the camp from the message given by the agent), the program must print **-1** as the output.

Boundary Condition(s):
1 <= Length of T, N <= 1000

Input Format:
The first line consists T.
The second line contains N.

Output Format:
The first line contains the number of terrorists in the camp from the message given by the agent or -1.

Example Input/Output 1:
Input:
moasaosamosamaoasmaa
osama

Output:
4

Explanation:
Here the given text is **moasaosamosamaoasmaa** and the name of the leader is **osama**.
The name **osama** occurs **4** times in the given text.
So 4 is printed as the output.

Example Input/Output 2:
Input:
jjukjukjukuukjukkkjuk
juk

Output:
-1

Max Execution Time Limit: 50 millisecs

50. MNC COMPANIES - SET 005

Solved Challenges
0/10
Robber & Gold Boxes

ID:12512
Solved By 3 Users

CTS GenCNext

In a magical island, there are four types of boxes **A**, **B**, **C**, **D** made up of gold.
The boxes of type A contains exactly A gold coins.
The boxes of type B contains exactly B gold coins.
The boxes of type C contains exactly 2*C gold coins.
The boxes of type D contains exactly 2*D gold coins.
A robber wants to collect the gold coins, but the magical island allows him to carry exactly **X** gold coins. The values A, B, C, D and X are passed as the input to the program. The program must print the maximum number of boxes that the robber can collect as the output.
Note: It is always possible to collect exactly X gold coins.

Boundary Condition(s):
1 <= A, B, C, D, X <= 10^4

Input Format:
The first line contains four integer values separated by a space.
The second line contains X.

Output Format:
The first line contains the maximum number of boxes that the robber can collect.

Example Input/Output 1:
Input:
5 8 5 3
23

Output:
4

Explanation:
Type A box contains **5** gold coins.
Type B box contains **8** gold coins.
Type C box contains **10** gold coins (2 * 5).
Type D box contains **6** gold coins (2 * 3).
The robber can collect exactly 23 gold coins.
One of the possible ways to collect 23 gold coins with the maximum number of boxes is given below.
3 boxes of type D + 1 box of type A = (3 * 6) + (1 * 5) = **23**
Hence the output is 4

Example Input/Output 2:
Input:
2 3 2 3
15

Output:
7

Explanation:
Type A box contains **2** gold coins.
Type B box contains **3** gold coins.
Type C box contains **4** gold coins (2 * 2).
Type D box contains **6** gold coins (2 * 3).
The robber can collect exactly 15 gold coins.
One of the possible ways to collect 15 gold coins with the maximum number of boxes is given below.
6 boxes of type A + 1 box of type B = (6 * 2) + (1 * 3) = **15**
Hence the output is 7

Max Execution Time Limit: 50 millisecs