

Programming Using C

week 15 practice session coding

Name:K.Kamaleshwaran

Department:AIML-'B'

Roll No.:242501079

An automatic cutting machine is used to cut rods into segments. The cutting machine can only hold a rod of one length in memory, and it can only make one cut at a time. Given the array lengths[] representing the desired lengths of each segment, determine if it is possible to make the necessary cuts using the machine. The rod's marked cut lengths already in the code given.

Example

```

n = 5
angles = (0, 3.14)
radius = 7

```

The red is initially unringed) $\times 2 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9$. Put out all the segments of length $4 = 1 + 2$ having a red 1. Then look for the length 7 and use the set of red segments of lengths 2 and 3. Since 7 is greater than or equal to red length $\times 2$, the last set can be made. Put out "Residue"

Example

```

n = 8
lengths = [3, 7, 8]
minLength = 7

```

The cost is actually $\text{arr.length} + d + 2 + 1 + 1$ (arriving here in this case, the initial cost is not of length $d + 1 + 1 + d$). Regardless of the length of the first city, the remaining cities will be shorter than arr.length . Because $n - 1 + 2$ can't be made, the answer is "Deposited".

Function Description

Complete the function `isPrime` in the editor below.

with the following parameters:
 1) length of the segments is l
 2) the minimum length the surface can accept

using "Possible" if all in F could be made \mathcal{O} -formation, and the string "Impossible".

- $2 \leq n \leq 10^5$
- $1 \leq i \leq 10^5$
- $1 \leq \text{lengths}[i] \leq 10^5$
- The sum of the elements of lengths equals the actual total length

Input Format for Custom Training

The first two columns are integers, n , the number of elements, n length.

Each level of the subsequence `subseq[i + 1 : n]` contains an integer, `length[i]`

The end line contains an integer, *midlength*, the maximum length accepted by the machine

Sample Case 1

Sample Input For Custom Testing

```

1  -- length() when n = 4
2  -- length() = {1, 1, 1, 1}
3
4
5
6  -- sort(length)

```

Example: Chuliput

Procedure

Supplementary

The second cut is $1 + 1 + 4 + 1 = 7$ units long. Put the red into lengths of $1 + 1 + 4 = 6$ and 1. Then cut the 6 and join into lengths 3 and $1 + 4 = 5$. The remaining segment is $1 + 2 = 3$ units and that is long enough to make the final cut.

Sample Case 1

Sample Input For Confidence Testing

TCOM	Reaction
1	$\rightarrow \text{imag[im]} \text{ with } n = 1$
2	$\rightarrow \text{imag[im]} = \{1, 1, 1\}$
3	
4	
5	$\rightarrow \text{antiimag[im]}$

Sample Output

Impressario

The second rail is $V = 0 - 2 \times 12$ mils long. After cutting it into 12, the rail will be inserted in under the second rail.

Direct payment

```

1 //lengths: int, startEnd[]: function int[]
2
3 //The function is expected to return a string.
4 //The function accepts following parameters:
5 // 1. int lengths: array lengths
6 // 2. int startEnd[]: array[]
7 //
8
9 //To return the string that the function, you should return the string of characters or array of characters.
10
11 For example:
12
13 char *char_string_using_start_end(char start[]) {
14     char *char[] = {"Start of function of string"};
15
16     return 0;
17 }
18
19 char *char_string_using_lengths(char start[]) {
20     char *s = "Hello" and *s = "Hello"
21     s = "Hello" (function of string)
22
23     return 0;
24 }
25
26 charSubstr(char *
27 char startEnd[]: int lengths, const, long *lengths, long minlength) {
28     long lengths[]: long[]
29
30     for(int i=lengths; i<lengths; i++)
31     {
32         minlength=lengths[i];
33     }
34     long current_minlength;
35     for(int i=lengths; i<lengths; i++)
36     {
37         current_minlength=lengths[i];
38         long minlength=lengths[i];
39         if(minlength>lengths[i])
40         {
41             return "Invalid";
42         }
43     }
44     return "Valid";
45 }
46
47 Return "Valid";

```

Forward all emails to: info@hugoboss.com