You can view this report online at : https://www.hackerrank.com/x/tests/1253385/candidates/33619845/report

Full Name:	Rohan Raj
Email:	rr07656@st.habib.edu.pk
Test Name:	**CS101 Lab# 11 - Fall 2021**
Taken On:	10 Dec 2021 14:49:12 PKT
Time Taken:	13033 min 14 sec/ 13080 min
Work Experience:	< 1 years
Invited by:	Aisha
Skills Score:	
Tags Score:	CS101 100/100 Lists 100/100 NestedLists 100/100

100%
610/610

scored in **CS101 Lab# 11 - Fall 2021** in 13033 min 14 sec on 10 Dec 2021 14:49:12 PKT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Wanna (reverse a) piece of me? > Coding	29 min 41 sec	150/ 150	✓
Q2	Find the Median > Coding	20 min 14 sec	30/ 30	✓
Q3	Remove all occurrences from a list (no slicing or dicing) > Coding	19 min 32 sec	100/ 100	✓
Q4	Print Matrix > Coding	24 min 39 sec	40/ 40	✓
Q5	Pick a column, any column (no comprehension needed) > Coding	13 min 9 sec	100/ 100	✓
Q6	Matrix Addition > Coding	2 hour 53 min 32 sec	90/ 90	✓
Q7	Special Sort > Coding	7 hour 23 min 44 sec	40/ 40	✓
Q8	Prime Factors & Special Numbers > Coding	58 min 44 sec	60/ 60	✓

QUESTION 1
Correct Answer
Score 150

Wanna (reverse a) piece of me? > Coding

QUESTION DESCRIPTION

Challenge

In Python, lists have a built-in method called `reverse` that reverses **all** elements in a list **in place**.

Write a function called `reverse_slice` that takes a list `t` and two indexes, `start` and `stop`, as its

arguments, and reverses all elements in the list slice `t[start:stop]` **in place**. The function **does not return anything useful** to stress the fact that it modifies the list. If the list slice is empty, list `t` is left unmodified.

Note

List indexes may be zero, positive, or negative, or may even indicate a slice that is empty. Your solution must handle all such situations. The order of the remaining elements in the list must be preserved.

Sample interaction

```
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, 3, 7)
>>> t
[0, 1, 2, 6, 5, 4, 3, 7, 8, 9]
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, 2, -4)
>>> t
[0, 1, 5, 4, 3, 2, 6, 7, 8, 9]
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, 0, 10)
>>> t
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, 0, -1)
>>> t
[8, 7, 6, 5, 4, 3, 2, 1, 0, 9]
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, 20, 30)
>>> t
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> t = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> reverse_slice(t, -20, -10)
>>> t
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Input/Output

Your function will be provided the list and the index arguments, `t`, `start`, and `stop`, respectively. The returned value from your function, as well as its side-effect on the list `t`, will be checked by HackerRank.

Constraints

None

INTERVIEWER GUIDELINES

Solution

```
# Using the list.reverse method:
def reverse_slice(t, start, stop):
    slice = t[start:stop]
    slice.reverse()
    t[start:stop] = slice

# Using the reversed builtin "function":
def reverse_slice(t, start, stop):
    t[start:stop] = reversed(t[start:stop])
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1 def reverse_slice(t, start, stop):
```

```
    l1 = []
    l2 = []
    l3 = []
    if stop != 0:
        l1 = t[0:start]
        l2 = t[start: stop]
        l2.reverse()
        l3 = t[stop: len(t)]
        t[:] = l1 + l2 + l3
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.0261 sec	8.05 KB
Testcase 1	Easy	Sample case	Success	10	0.0442 sec	8.06 KB
Testcase 2	Easy	Sample case	Success	10	0.0233 sec	7.94 KB
Testcase 3	Easy	Sample case	Success	10	0.028 sec	8.15 KB
Testcase 4	Easy	Sample case	Success	10	0.0227 sec	8.16 KB
Testcase 5	Easy	Sample case	Success	10	0.0228 sec	8.04 KB
Testcase 6	Easy	Hidden case	Success	10	0.0281 sec	8.07 KB
Testcase 7	Easy	Hidden case	Success	10	0.0244 sec	8.21 KB
Testcase 8	Easy	Hidden case	Success	10	0.0421 sec	8.06 KB
Testcase 9	Easy	Hidden case	Success	10	0.0284 sec	8.16 KB
Testcase 10	Easy	Hidden case	Success	10	0.0238 sec	8.03 KB
Testcase 11	Easy	Hidden case	Success	10	0.024 sec	8.05 KB
Testcase 12	Easy	Sample case	Success	10	0.0283 sec	8.06 KB
Testcase 13	Easy	Sample case	Success	10	0.0214 sec	8.05 KB
Testcase 14	Easy	Sample case	Success	10	0.0264 sec	8.02 KB

No Comments

QUESTION 2

Correct Answer

Score 30

Find the Median › Coding

QUESTION DESCRIPTION

Problem

The median of a list of numbers is essentially it's middle element after sorting. The same number of elements occur after it as before. Given a list of numbers with an odd number of elements, can you find the median?

For example, the median of `arr = [1, 2, 3, 4, 5]` is `3`, the middle element in the sorted array.

Function Description

Complete the *findMedian* function in the editor below. It must return an integer that represents the median of the array.

findMedian has the following parameter(s):

- *arr*: an unsorted array of integers

Input Format

The first line contains the integer `n`, the size of `arr`.

The second line contains `n` space-separated integers.

Constraints

- 1 <= n <= 1000001
- n is odd
- -10000 <= arr[i] <= 10000

Output Format

Output one integer, the median.

Sample Input 0

```
7
0 1 2 4 6 5 3
```

Sample Output 0

```
3
```

Explanation 0

The sorted arr = [0, 1, 2, 3, 4, 5, 6]. It's middle element is at arr[3] = 3.

INTERVIEWER GUIDELINES

Solution

```
def findMedian(arr):
  return sorted(arr)[len(arr)//2]
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1 # Complete the findMedian function below.
2 def findMedian(arr):
3     arr.sort()
4     return (arr[int(n/2)])
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	5	0.0316 sec	7.96 KB
Testcase 1	Easy	Hidden case	Success	5	0.0328 sec	8.02 KB
Testcase 2	Easy	Hidden case	Success	5	0.0226 sec	8.04 KB
Testcase 3	Easy	Hidden case	Success	5	0.0237 sec	8.16 KB
Testcase 4	Easy	Hidden case	Success	5	0.0264 sec	7.93 KB
Testcase 5	Easy	Hidden case	Success	5	0.0318 sec	8.15 KB

No Comments

QUESTION 3

Correct Answer

Score 100

Remove all occurrences from a list (no slicing or dicing) › Coding

QUESTION DESCRIPTION

Challenge

Score 100

In Python, lists have a built-in method called `remove` that removes the first element whose value is equal to the given argument. If no such value is found, the method raises an error.

Write a function called `remove_all` that takes a list `t` and a value `v` as its arguments, and removes all occurrences of the value `v` from the list `t`. The function **returns** the number of items removed from the list. If the value `v` does not occur in list `t`, the function returns 0 and leaves list `t` unchanged.

Note

You may not use list indexing (e.g., `t[5]`, `t[-1]`, `t[i]`, etc.) or list slicing (e.g., `t[2:5]`, `t[i:i + 3]`, `t[:]`, etc.) in your solution. Your list may contain elements of any type, so do not use any type-specific code for checking membership of elements. The order of elements in the list must be preserved.

Sample interaction

```
>>> t = [8, 6, 6, 7, 5, 3, 7, 0, 7, 7, 9, 7]
>>> t
[8, 6, 6, 7, 5, 3, 7, 0, 7, 7, 9, 7]
>>> remove_all(t, 2)
0
>>> t
[8, 6, 6, 7, 5, 3, 7, 0, 7, 7, 9, 7]
>>> remove_all(t, 5)
1
>>> t
[8, 6, 6, 7, 3, 7, 0, 7, 7, 9, 7]
>>> remove_all(t, 7)
5
>>> t
[8, 6, 6, 3, 0, 9]
```

Input/Output

Your function will be provided the list and value arguments, `t` and `v`, respectively. The returned value from your function, as well as its side-effect on the list `t`, will be checked by HackerRank.

Constraints

None

INTERVIEWER GUIDELINES

Solution

```
# Using the list.count method and a for loop:
def remove_all(t, v):
    n = t.count(v)
    for _ in range(n):
        t.remove(v)
    return n
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1  def remove_all(t, v):
2      c = 0
3      for i in t:
4          if i == v:
5              c += 1
6      while v in t:
7          t.remove(v)
8      return c
```

```
9
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.0568 sec	10.1 KB
Testcase 1	Easy	Sample case	Success	10	0.0588 sec	9.89 KB
Testcase 2	Easy	Sample case	Success	10	0.064 sec	10.1 KB
Testcase 3	Easy	Hidden case	Success	10	0.0616 sec	10 KB
Testcase 4	Easy	Hidden case	Success	10	0.0694 sec	10 KB
Testcase 5	Easy	Hidden case	Success	10	0.0618 sec	9.95 KB
Testcase 6	Easy	Hidden case	Success	10	0.0658 sec	10.1 KB
Testcase 7	Easy	Sample case	Success	10	0.0634 sec	10.2 KB
Testcase 8	Easy	Sample case	Success	10	0.0563 sec	10.1 KB
Testcase 9	Easy	Sample case	Success	10	0.0612 sec	10.1 KB

No Comments

QUESTION 4

Correct Answer

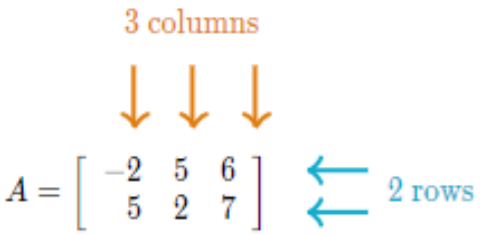
Score 40

Print Matrix › Coding

QUESTION DESCRIPTION

Problem

A **matrix** is a rectangular arrangement of numbers, symbols or expressions into rows and columns.
For example, matrix A has two **rows** and three **columns**.



In Python, we can implement a matrix as a nested list (list inside a list). We can treat each element as a row of the matrix.
In our first lab, we will be performing different arithmetic operations on two matrices.

As a refresher exercise before the first lab, lets do a small refresher exercise!

Challenge

Write a function ***print_matrix*** that takes a integer matrix `A` as input and prints the contents of the matrix passed to it.

Sample

```
>>> print_matrix([[4,8],[3,7]])
4 8
3 7

>>> print_matrix([[5,2],[0,1],[1,9]])
```

```
5 2
0 1
1 9

>>> print_matrix([[1],[2]])
1
2

>>> print_matrix([[5,2,4],[0,1,7],[1,9,8]])
5 2 4
0 1 7
1 9 8
```

INTERVIEWER GUIDELINES

Solution

```
def print_matrix(A):
    for i in range(len(A)):
        for j in range(len(A[0])):
            print(A[i][j], end=" ")
        print("")
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def print_matrix(A):
    for i in range (len(A)):
        for j in range (len(A[i])):
            print (A[i][j], end = " ")
        print()
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.0494 sec	9.18 KB
Testcase 1	Easy	Sample case	Success	10	0.0301 sec	9.06 KB
Testcase 2	Easy	Sample case	Success	10	0.0347 sec	9.02 KB
Testcase 3	Easy	Sample case	Success	10	0.0439 sec	9.06 KB

No Comments

QUESTION 5

Correct Answer

Score 100

Pick a column, any column (no comprehension needed) › Coding Lists

NestedLists CS101

QUESTION DESCRIPTION

Challenge

To represent a matrix, such as a spreadsheet or a collection of records, a common practice in Python is to use a list of lists. Each inner list represents a row within the matrix.

For example, the following matrix:

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 6 \\ 1 & 3 & 0 \end{pmatrix}$$

has the representation:

```
>>> mat_a = [[2, 3, 5], [4, 1, 6], [1, 3, 0]]
```

It is often useful to pick a column from the matrix.

Write a function called `pick` that takes a list `t` and an index `col` as its arguments, and **returns** a list containing values from each row's index `col`. The function leaves list `t` unmodified.

Sample interaction

```
>>> t = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> pick(t, 1)
[2, 5, 8]
>>> pick(t, 0)
[1, 4, 7]
>>> pick(t, -1)
[3, 6, 9]
>>> t
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Input/Output

Your function will be provided the list and the column index arguments, `t` and `col`, respectively. The returned value from your function, as well as its side-effect on the list `t`, will be checked by HackerRank.

Constraints

`-len(r) <= col < len(r)` where `isinstance(r, list) and r in t` is `True`.

INTERVIEWER GUIDELINES

```
# Using a list comprehension:
def pick(t, col):
    return [v[col] for v in t]

# Using the list.append method and a for loop:
def pick(t, col):
    u = []
    for v in t:
        u.append(v[col])
    return u
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1 def pick(t, col):
2     x = []
3     for i in t:
4         x.append(i[col])
5     return x
```

```
6
7  t = eval(input())
8  col = int(input())
9  print(pick(t, col))
10 print(t)
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.027 sec	8.02 KB
Testcase 1	Easy	Sample case	Success	10	0.0287 sec	8.14 KB
Testcase 2	Easy	Sample case	Success	10	0.0264 sec	8.18 KB
Testcase 3	Easy	Hidden case	Success	10	0.0259 sec	7.91 KB
Testcase 4	Easy	Hidden case	Success	10	0.0234 sec	7.91 KB
Testcase 5	Easy	Hidden case	Success	10	0.0298 sec	7.97 KB
Testcase 6	Easy	Hidden case	Success	10	0.0226 sec	8.03 KB
Testcase 7	Easy	Sample case	Success	10	0.0256 sec	8.03 KB
Testcase 8	Easy	Sample case	Success	10	0.0246 sec	8.08 KB
Testcase 9	Easy	Sample case	Success	10	0.0304 sec	8.14 KB

No Comments

QUESTION 6

Correct Answer

Score 90

Matrix Addition › Coding

QUESTION DESCRIPTION

Challenge

Write a function, `matrix_addition`, that takes two integer matrices `A` and `B` and returns a matrix whose entries are the sum of the corresponding entries in matrices `A` and `B`.

Sample

```
>>> matrix_addition( [[4,8],[3,7]] , [[1,0],[5,2]] )
[[5,8],[8,9]]
>>> matrix_addition( [[5,2],[0,1],[1,9]] , [[2,3],[4,1],[0,2]] )
[[7,5],[4,2],[1,11]]
>>> matrix_addition([[1],[2]],[[3,5],[4,6]])
Matrices A and B don't have the same dimension required for matrix
addition.
```

INTERVIEWER GUIDELINES

```
def matrix_addition(X, Y):
    # add matrices X and Y
    # return the resulting matrix

    size1 = (len(X), len(X[0]))
    size2 = (len(Y), len(Y[0]))
```

```
    if size1 != size2:
        return("Matrices A and B don't have the same dimension required
for matrix addition.")

    Z = []

    for i in range(len(X)):
        list = []
        for j in range(len(X[i])):
            list.append(X[i][j] + Y[i][j])
        Z.append(list)

    return Z
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def matrix_addition(A,B):
    x = []
    y = []
    for i in range (len (A)):
        if len(A[i]) != len(B[i]):
                return ("Matrices A and B don't have the same dimension
required for matrix addition.")
        else:
            for j in range (len(A[i])):
                x.append(A[i][j] + B[i][j])
            y.append(x)
            x = []

    return y
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.044 sec	9.04 KB
Testcase 1	Easy	Sample case	Success	10	0.0539 sec	9.27 KB
Testcase 3	Easy	Sample case	Success	10	0.0499 sec	9.2 KB
Testcase 4	Easy	Sample case	Success	10	0.049 sec	9.04 KB
Testcase 5	Easy	Sample case	Success	10	0.0463 sec	9.05 KB
Testcase 6	Easy	Sample case	Success	10	0.0469 sec	8.95 KB
Testcase 7	Easy	Sample case	Success	10	0.0359 sec	9.02 KB
Testcase 7	Easy	Sample case	Success	10	0.0387 sec	9.04 KB
Testcase 8	Easy	Sample case	Success	10	0.0387 sec	9.04 KB

No Comments

QUESTION 7

Correct Answer

Special Sort › Coding

QUESTION DESCRIPTION

Score 40

The university needs your help in sorting out its student data. The IT department has generated a list containing details of every student. However, they are having trouble sorting the data in a specific order. See if you can help your university's IT department by using your exceptional programming skills.

Write a function 'special_sort' that takes as parameter a list 'lst' and returns a sorted version of that list.

The list should be sorted according to the following order of precedence:

1. Class/Batch
2. Major
3. Name (alphabetical order)

```
>>> special_sort([['Sarwan', 'EE', '2021'], ['Lulowalokand Wala', 'CND',
'2021'], ['Hamza Junaid', 'EE', '2021'], ['Ahsan Qadeer', 'CS', '2020'],
['Muhammad Ali Bhutto', 'EE', '2020'], ['Marium Habiby', 'SDP', '2021'],
['Adil Ali Khan', 'EE', '2021']])
[['Ahsan Qadeer', 'CS', '2020'], ['Muhammad Ali Bhutto', 'EE', '2020'],
['Lulowalokand Wala', 'CND', '2021'], ['Adil Ali Khan', 'EE', '2021'],
['Hamza Junaid', 'EE', '2021'], ['Sarwan', 'EE', '2021'], ['Marium
Habiby', 'SDP', '2021']]
```

INTERVIEWER GUIDELINES

Solution

```
def special_sort(lst):
    # The list items are [name, major, batch]. The required sort order
    # is the reverse, i.e. batch, major, name. To sort according to
    # this reverse order, reverse each item in the list, sort the
    # list, and reverse each item again.
    students = lst[:]
    for student in students:
        student.reverse()
    students.sort()
    for student in students:
        student.reverse()
    return students
```

CANDIDATE ANSWER

Language used: **Python 3**

```
1  def special_sort(lst):
2      if lst == []:
3          return []
4      if lst != []:
5          x = lst[0][2]
6          z = lst[0][2]
7          y = []
8          v = []
9          q = []
10         for i in range(len(lst)):
11             if lst[i][2] < x:
12                 x = lst[i][2]
13         for l in range(len(lst)):
14             if lst[l][2] > z:
15                 z = lst[l][2]
16         if type (x) == str and type (z) == str:
17             x = int(x)
18             z = int(z)
```

```
            z = int(z)
            for k in range(x, z + 1):
                for j in range(len(lst)):
                    if lst[j][2] == str(k):
                        y.append(lst[j])
                if y != []:
                    b = major(y)
                    for s in b:
                        for t in s:
                            q.append(t)
                y = []
        elif type (x) == int and type (z) == int:
            for k in range(x, z + 1):
                for j in range(len(lst)):
                    if lst[j][2] == k:
                        y.append(lst[j])
                if y != []:
                    b = major(y)
                    for s in b:
                        for t in s:
                            q.append(t)
                y = []
        return (q)

def major(m):
    a = []
    b = []
    c = []
    for p in range(len (m)):
        if m[p][1] not in a:
            a.append(m[p][1])
    a.sort()
    for x in range (len(a)):
        for p in range(len (m)):
                if m[p][1] == a[x]:
                    b.append(m[p])
        r = name(b)
        c.append(r)
        b = []
    return (c)

def name (n):
    d = []
    h = []
    for e in range(len(n)):
        if n[e][0] not in d:
            d.append(n[e][0])
    d.sort()
    for f in range (len(d)):
        for g in range(len (n)):
                if n[g][0] == d[f]:
                    h.append(n[g])
    return (h)

print(special_sort(eval(input())))
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.0221 sec	8.41 KB
Testcase 1	Easy	Hidden case	Success	10	0.0263 sec	8.3 KB
Testcase 2	Easy	Hidden case	Success	10	0.0271 sec	8.25 KB
Testcase 3	Easy	Hidden case	Success	10	0.0262 sec	8.05 KB

No Comments

QUESTION 8

Correct Answer

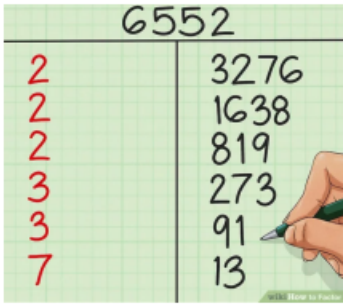
Score 60

Prime Factors & Special Numbers › Coding

QUESTION DESCRIPTION

A special number is a number that is equal to the sum of its prime factors.

"A prime number can only be divided by 1 or itself, so it cannot be factored any further! Every other whole number can be broken down into prime number factors. It is like the Prime Numbers are the basic building blocks of all numbers." - Math is fun



Write a function 'prime_factors' that takes as parameter 'num', a positive integer, and returns a list containing its prime factors.

```
>>> prime_factors(6552)
[2, 2, 2, 3, 3, 7, 13]
>>> prime_factors(1567)
[1567]
```

Write another function named 'special_numbers' that takes as parameter a positive integer 'num' and returns a list of all the specials numbers from 0 to that number inclusive. Use your 'prime_factors' function in this function.

```
>>> special_numbers(25)
[0, 2, 3, 4, 5, 7, 11, 13, 17, 19, 23]
```

Input Format

The input begins with `select` followed by n on the next line.

- If `select` is 1, then prime_factors must be called on n.
- If `select` is 2, then special_numbers must be called on n.

Your code must make the correct call based on `select`.

INTERVIEWER GUIDELINES

Solution

```
import math
def prime_factors(n):
    prime_factors = []
    # Extract all factors of n starting from 2. No new factors are to
    # be found beyond sqrt(n).
    for factor in range(2, int(math.sqrt(n))+1):
        # Store factor if it divides n and update n
```

```
        q, r = divmod(n, factor)
        while r == 0:
            prime_factors.append(factor)
            n = q
            q, r = divmod(n, factor)
    # A composite n be updated to 1 when the loop exits. A prime n
    # would remain unaffected. If so, store n as its prime factor.
    if n > 1:
        prime_factors.append(n)
    return prime_factors

def special_numbers(n):
    special_numbers = []
    for i in range(n+1):
        # A special number is the sum of its prime factors.
        if sum(prime_factors(i)) == i:
            special_numbers.append(i)
    return special_numbers
```

CANDIDATE ANSWER

Language used: **Python 3**

```
# Enter your code here.
select = int(input())
n = int(input())

def prime_factors(num):
    s = []
    d = 2
    while num >= d:
        while num % d == 0:
            s.append(d)
            num = num // d
        d += 1
    return s

def special_numbers(num):
    s = 0
    x = []
    for i in range (num + 1):
        s = 0
        l = prime_factors(i)
        for j in l:
            s += j
        if i == s:
            x.append(s)
    return x
if select == 1:
    print(prime_factors(n))
elif select == 2:
   print (special_numbers(n))
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.0319 sec	8.09 KB
Testcase 1	Easy	Sample case	Success	10	0.0325 sec	8.21 KB
Testcase 2	Easy	Hidden case	Success	10	0.0265 sec	8.18 KB
Testcase 3	Easy	Sample case	Success	10	0.0311 sec	8.05 KB

Testcase 4	Easy	Hidden case	Success	10	0.034 sec	8.16 KB
Testcase 5	Easy	Hidden case	Success	10	0.0266 sec	8.07 KB

No Comments

PDF generated at: 23 Dec 2021 09:56:31 UTC