Arbitrary Shopping

An avid shopper goes to a clothing store and picks any arbitrary outfit. Later he buys all consecutive outfits picked up, as long as he has the money to pay for them up to the n^{th} outfit. For example, if he starts from outfit *i*, he will continue to outfit *i + 1*, *i + 2* and so on until he cannot afford to purchase another outfit. Help him determine what is the maximum number of outfits he can buy.

Function Description

Complete the function *getMaximumOutfits* in the editor below. The function must return an integer that denotes the maximum number of outfits that can be bought.

getMaximumOutfits has the following parameters:

money: An integer, which denotes the amount of money.

outfits: An array of integers, which denotes the outfits.

Constraints

- $1 \le n \le 10^5$
- $1 \le outfits[i] \le 100$
- $1 \le money \le 10^6$

Input Format For Custom Testing

Sample Case 0

Sample Input For Custom Testing

```
3
10
10
10
5
```

Sample Output

```
0
```

Explanation

There are *3* outfits each costing *10*. With *money = 5*, he cannot buy any of the outfits. Hence, the answer is *0*.

Sample Case 1

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour.

Start tour

For help on how to read input and write output in Python 3, click here.

Original Code | Python 3

```
22  # Complete the 'getMaximumOutfits' function below.
23  #
24  # The function is expected to return an INTEGER.
25  # The function accepts following parameters:
26  #  1. INTEGER_ARRAY outfits
27  #  2. INTEGER money
28  #
29
30  def getMaximumOutfits(outfits, money):
```

```
35      for i, o in enumerate(outfits):
36          if money == 0:
37              return bought
38
39          if money >= o:
40              if DEBUG: print("%(money)s - %(o)s (BOUGHT!)" % vars())
41              bought += 1
42              money -= o
43          else:
44              if DEBUG: print("%(money)s - %(o)s" % vars())
45
46      return bought
47
48  if __name__ == '__main__':…
```

Line: 10 Col: 1

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

Compiled successfully. 3/8 test cases passed.

Tip: Debug your code against custom input

Test Case #1:	✔	Test Case #4:	✖	Test Case #7:	✖
Test Case #2:	✔	Test Case #5:	✖	Test Case #8:	✖
Test Case #3:	✔	Test Case #6:	✖		

Testcase 1: ***Success***

Input [Download]

```
3
10
10
10
5
```

Your Output

```
0
```

Expected Output [Download]

```
0
```

Testcase 2: ***Success***

Input [Download]

```
3
5
10
10
5
```

Your Output

```
1
```

Expected Output [Download]

1
2
3
4
5
6

Testcase 3: ***Success***

Your Output

```
Output hidden
```

Testcase 4: ***Wrong Answer***

Your Output

```
Output hidden
```

Testcase 5: ***Wrong Answer***

Your Output

```
Output hidden
```

Testcase 6: ***Wrong Answer***

Your Output

```
Output hidden
```

Testcase 7: ***Wrong Answer***

Your Output

```
Output hidden
```

Testcase 8: ***Wrong Answer***

Your Output

```
Output hidden
```