'''

Charan is working on photoshop design in which he has to type a set of digits

from left to right. Each digit(0-9) is defined with different units of space

required and represented by an array sp[].

For example, sp[0] is the space required for the digit '0', sp[1] is the space

required for the digit[1], and so on..

Charan is trying to type the given digits in the same order which is given,

across several lines and each line can take atmost 50 units of space.

Given a set of digits as string, return an array ar[] which consists the

following two values.

1.ar[0] is the number lines required.

2.ar[1] is the units of space in the last line.

Input Format:

-------------

Line-1: 10 space seperated positive integers.

Line-2: A string str[] represents the set of digits.

Output Format:

--------------

Print two space seperated values.

Constraints:

sp.length == 10

2 <= sp[i] <= 20

1 <= str.length <= 1000

str contains only digits from 0 to 9.

Sample Input-1:

---------------

3 2 3 4 5 6 7 8 9 10

9345678123

Sample Output-1:

----------------

2 9

Explanation:

------------

9345678 can fit in line-1.

123 in line-2:

So

Number of lines=2

Last line consits 123 with units 2+3+4=9

Sample Input-2:

---------------

3 2 3 4 5 6 7 8 9 10

99999267

Sample Output-2:

----------------

2 18

Explanation:

------------

99999 can fit in line-1.

267 in line-2:

So

Number of lines=2

Last line consits 267 with units 3+7+8=18.

Write your python code below

'''

l=list(map(int,input().split()))

n=input()

sum=0

count=0

l1=[]

for i in n:

if(sum+l[int(i)]>50):

# print(l1)

l1.clear()

count+=1

sum=0

sum+=l[int(i)]

l1.append(int(i))

sum1=0

for i in l1:

sum1+=l[int(i)]

if(sum1<=50):

count+=1

print(count," ",sum1)

A device is secured with a digital lock,

The digital lock is made up of a 3\*3 grid like below:

| 1 | 2 | 3 |

| 4 | 5 | 6 |

| 7 | 8 | 9 |

The lock is secured by creating pattern using the above grid.

A valid pattern as formed as follows:

Rules for a valid pattern:

- Each pattern must connect at least m keys and at most n keys.

- All the keys must be distinct.

- If the line connecting two consecutive keys in the pattern passes through

any other keys, the other keys must have already the part of the pattern.

No jumps through a key which is not part of the pattern is allowed.

- The order of keys used matters.

You will be given two integers m and n, where 1 ≤ m ≤ n ≤ 9,

Your task is to count the total number of patterns of the device lock,

which consist of minimum of m keys and maximum n keys.

Examples of valid and invalid patterns:

Invalid move: 4 - 1 - 3 - 6

Line 1 - 3 passes through key 2 which had not been part of the pattern.

Invalid move: 4 - 1 - 9 - 2

Line 1 - 9 passes through key 5 which had not been part of the pattern.

Valid move: 2 - 4 - 1 - 3 - 6

Line 1 - 3 is valid because it passes through key 2, which had been part

of the pattern

Valid move: 6 - 5 - 4 - 1 - 9 - 2

Line 1 - 9 is valid because it passes through key 5, which had been part

of the pattern

Input Format:

-------------

Two space separated integers M and N.

Output Format:

--------------

Print an integer, number of valid patterns formed using the grid.

Sample Input-1:

---------------

1 1

Sample Output-1:

----------------

9

Sample Input-2:

---------------

1 2

Sample Output-2:

----------------

65







