「交代交代」 /難易度:3

問題タイプ:コーディング問題 目標タイム:30分 アルゴリズム/ 実装問題

問題文

あなたはやり遂げるのに $task_i$ 時間かかる仕事を N 個抱えています。

同じ仕事を長時間行うと集中力が切れてしまう為、 1 番目の仕事をやりとげるか time 時間行うと 2番目の仕事にとりかかり、 2 番目の仕事をやりとげるか time 時間行うと 3番目の仕事にとりかかり…と N 番目の仕事まで続けます。

その後、まだ仕事が残っているようなら再び 1 番目から N 番目までのやり遂げていない仕事に対して同様にとりかかります。

全ての仕事を終えたとき、取り掛かった履歴を以下の様な形式で出力してください。

例

完了するのにそれぞれ6時間、4時間、9時間かかる3件の仕事を抱えており、 一つの仕事を連続して行う最大時間数が5時間の場合

1番目の仕事に5時間取り掛かる

2番目の仕事に4時間取り掛かる(完了)

3番目の仕事に5時間取り掛かる

全ての仕事を一通りこなしましたがまだ仕事が残っています。再び1番目の仕事から順に取り掛かります。

1番目の仕事に1時間取り掛かる(完了)

2番目の仕事は既に終えているので省略

3番目の仕事に4時間取り掛かる(完了)

全ての仕事を終えました

上記の場合

5
 4

3 5

1 1

3 4

と出力してください

入力される値

N time

task_1 task_2 task_3 ... task_N

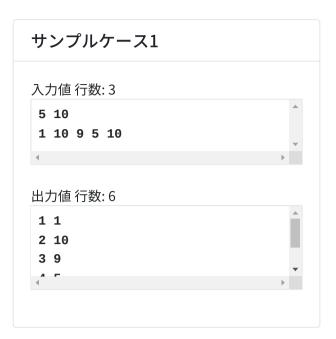
- *N*: あなたが抱えている仕事の件数
- time: 一つの仕事を連続して行う最大時間数
- ullet $task_i:i$ 番目の仕事をやり遂げるのに必要な総時間数

期待される出力値

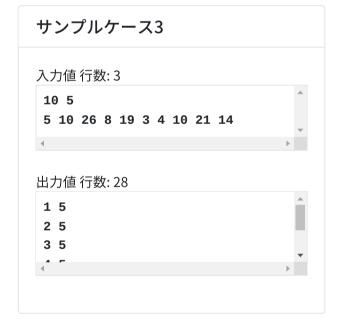
• 仕事に取り掛かった順番とその時間 出力方法は問題文の例やサンプルケースの解説を参考にしてください

制約

- $N, time, task_i$:整数
- $1 \le N \le 100$
- $1 \leq time, task_i \leq 100$







テストする

2022/03/25 10:42 TechFUL

サンプルケース1

入力值

5 10

1 10 9 5 10

期待される出力値

1 1

2 10

3 9

4 5

5 10

説明

1番目の仕事を1時間行い、完了します。

2番目の仕事を10時間行い、完了します。

3番目の仕事を9時間行い、完了します。

4番目の仕事を5時間行い、完了します。

5番目の仕事を10時間行い、完了します。

全ての仕事をやり遂げたので、その履歴を以下のように出力します。

1 1

2 10

3945

5 10

サンプルケース2

入力値

4 5 3 10 11 9

期待される出力値

1 3

2 5

3 5

4 5

2 5

3 54 4

3 1

説明

1番目の仕事を 3 時間行い、完了します。

2番目の仕事を5時間行います。

3番目の仕事を5時間行います。

4番目の仕事を5時間行います。

2番目の仕事を5時間行い、完了します。

3番目の仕事を5時間行います。

4番目の仕事を4時間行い、完了します。

3番目の仕事を1時間行い、完了します。

サンプルケース3

入力値

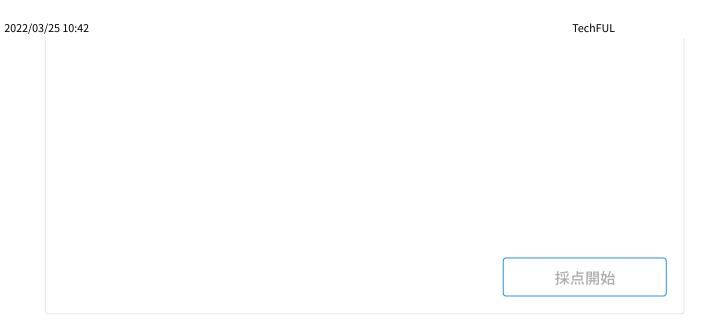
10 5

5 10 26 8 19 3 4 10 21 14

期待される出力値

```
TechFUL
当ころ しっち り 耳くこ 正
1 5
2 5
3 5
4 5
5 5
6 3
7 4
8 5
9 5
10 5
2 5
3 5
4 3
5 5
8 5
9 5
10 5
3 5
5 5
9 5
10 4
3 5
5 4
9 5
3 5
9 1
3 1
```

```
v i
                                                           配置変更
                                                                         C++
     #include<bits/stdc++.h>
1
2
     using namespace std;
3
4
5
     int main(){
6
         int n,time;
7
         cin >> n >> time;
8
         vector<int> num(n);
         for(int i=0;i<n;i++){
9
10
             cin >> num[i];
11
         }
12
         int i=0;
13
         int stop = n;
14
         while(true){
             if(stop <= 0){</pre>
15
16
                 break;
17
18
             stop--;
19
             int work = min(time,num[i]);
             if(work <= 0){
20
21
                 continue;
22
             cout << i+1 <<" "<< work << endl;</pre>
23
             num[i] -= work;
24
25
             if(stop <= 0){
26
                break;
27
             i++;
28
             if(i == n){
29
30
                i = 1;
31
                 stop = n;
```



2017 444 Inc. all rights reserved