

First Fit & Best Fit

Given a list of integers (each one can range from 1 to 100) like [10, 20, 90, 50, 10, 20]. This brings up an interesting question: How do you partition this list into a smaller one that has the summation of numbers inside to not be more than 100 and has the least number of sub-lists. **Ex.** [10, 20, 90, 50, 10, 10] is [10, 20, 50, 10], [90, 10]

The easiest way to partition (but not have the least number of sub-lists) this is to check each number and put it into the existing sub-lists. There are two ways to do that, which are:

- **First Fit** – Find the sub-list from left to right, and immediately fill in the one that can fit in the data.
Ex: If you want to put 20 into [[90, 5], [50], [70, 10]]. 20 cannot be put into [90, 5], but can be put into [50]. So the result is: [[90, 5], [50, 20], [70, 10]]
- **Best Fit** – Find the available sub-list that can be filled, then pick the sub-list with the summation closest to 100 to fill in. **Ex:** If you want to put 20 into [[90, 5], [50], [70, 10]]. 20 cannot be put in [90, 5], but can be put into [50] or [70, 10]. The latter is picked because the summation is the closest to 100. The result is [[90, 5], [50], [70, 10, 20]]

In the case of not being able to put the data into any sub-lists, just make a new sub-list at the end of the existing sub-lists.

Write these four functions that work like the comment below.

```
def first_fit(L, e):  
    # put e into sub-list inside L by using First Fit.  
  
def best_fit(L, e):  
    # put e into sub-list inside L by using Best Fit.  
  
def partition_FF(D):  
    # return a list of partitioned data from D using First Fit.  
  
def partition_BF(D):  
    # return a list of partitioned data from D using Best Fit.  
  
exec(input().strip()) # must have this line to submit into grader
```

Input

Python code for the functions.

Output

The results of code execution.

Example

Input (from keyboard)	Output (on screen)
<code>L=[[50],[90]];first_fit(L,10);print(L)</code>	<code>[[50, 10], [90]]</code>
<code>L=[[50],[90]];best_fit(L,10);print(L)</code>	<code>[[50], [90, 10]]</code>

<code>print(partition_FF([50,90,10,80,50,20]))</code>	<code>[[50, 10, 20], [90], [80], [50]]</code>
<code>print(partition_BF([50,90,10,80,50,20]))</code>	<code>[[50, 50], [90, 10], [80, 20]]</code>