[[](https://codechalleng.es/profiles/danielhao5)](https://codechalleng.es/profiles/danielhao5)Bite 265. Optimal fund raising

Here is another classic algorithm problem wrapped in a Bite. Enjoy!

A mayor in a remote mountain village faced an emergency and he needs your help. He just learned a poor boy is very sick, and will need to go to the city hospital immediately.

He only has one night to collect the contribution from the villagers, so that he can escort the boy and his family to the city next morning. Luckily, he has kept a 'secret list' of each family's "financial status + willingness to help" in his file over the years.

So the challenge is how to find the starting family in the geographicly sparse mountain area, making sure he can collect maximum funds.

The goal is to help the mayor to collect the maximum contribution and determine which families to start (and end). These are the key elements of this mission:

To find the families in the closest neighborhood in the village with best possible donations that he can collect. You can think of this closest neighborhood as - "contiguous families block" in the "list", without circulating back. Since time is essence in this matter, he wants to get the funds quickly and prepare for next day's trip.

Let's make some concrete examples to illustrate these points:

* Input: the list represents that each family's 'capacity' to contribute in $dollars\_amount: it could be one of these values: positive, negative, or zero. For example: [0, 1, 2, 4, -3, -3]
* Note - the mountain village could have hundreds of families scattered around in wide area, so he would just go in one-direction. Therefore he prefers one-way trips (not circular!) - one pass. (this is known in CS as O(n) time requirement)
* Output: 3 items he needs to know - the maximum contribution (amount), starting*family, ending*family (eg. by its index). See below examples.
* Bonus - it will be great if you can help him find out the possibility of positive donations - e.g. there is at least one family can contribute before he starts the night trip. If the "donations" are negative - meaning all families are poor - then you could give him a warning message: "Mission impossible. No one can contribute.", exiting the program with (0, 0, 0) - see example 4 below.

##### Examples:

###### Example 1

input:

village = [0, -3, 2, 1, -7, 5, 3, -1, 6]

# index 0 1 2 3 4 5 6 7 8

# \*\*\*\*\*\*\*\*\*\*\*\*

>>> max\_fund(village)

(13, 6, 9) # Since mayor is not an engineer, you will need to convert the index offset

▸␣␣␣ ▸from zero to one. (eg. start + 1, end + 1)

fund: 13;

starting at 6; index 5 + 1

ending at 9; indext 8 + 1 (from mayor's perspective - the first potion should be ONE)

###### Example 2

input:

One = [0, 1, -1, -5, 0, 4, -3, -2]

>>> max\_fund(one)

(4, 6, 6))

###### Example 3

input:

penniless = [0, 0, 0, 0, 1, -5, -2, -1, -3]

>>> max\_fund(penniless)

(1, 5, 5))

###### Example 4

input:

extremes = [-1, -2, -3, -4, -5]

>>> max\_fund(extremes)

'Mission Impossible. No one can contribute!'

(0, 0, 0)