Question 1

**Problem Description:**

A non-empty array A consisting of N integers is given. Array A represents numbers on a tape. Any integer P, such that 0 < P < N, splits this tape into two non-empty parts: A[0], A[1], ..., A[P ? 1] and A[P], A[P + 1], ..., A[N -1].

The difference between the two parts is the value of: |(A[0] + A[1] + ... + A[P- 1]) - (A[P] + A[P + 1] + ... + A[N - 1])|

In other words, it is the absolute difference between the sum of the first part and the sum of the second part. For example, consider array A such that:

A[0] = 3

A[1] = 1

A[2] = 2

A[3] = 4

A[4] = 3

We can split this tape in four places:

P = 1, difference = |3 - 10| = 7

P = 2, difference = |4 - 9| = 5

P = 3, difference = |6 - 7| = 1

P = 4, difference = |10 - 3| = 7

Write a program that, given a non-empty array A of N integers, returns the minimal difference that can be achieved. For example, given:

A[0] = 3

A[1] = 1

A[2] = 2

A[3] = 4

A[4] = 3

the function should return 1, as explained above.

Write an efficient algorithm for the following assumptions:

N is an integer within the range [2..100,000];

Each element of array A is an integer within the range [-1,000..1,000].

Question 2

**Problem Description:**

A string S consisting of N characters is considered to be properly nested if any of the following conditions is true:

S is empty;

S has the form "(U)" or "[U]" or "{U}" where U is a properly nested string;

S has the form "VW" where V and W are properly nested strings.

For example, the string "{[()()]}" is properly nested but "([)()]" is not.

Write a program that given a string S consisting of N characters, returns 1 if S is properly nested and 0 otherwise.

For example, given S = "{[()()]}", the function should return 1 and given S = "([)()]", the function should return 0, as explained above.

a = “(())”

dict = {}

dict[“(“] = “)”

dict[“[“] = “]”

dict[“{“] = “}”

list1=[]

for i in a

if i == ‘)‘ and list1[-1] == “(”

list1.pop(-1)

else :

list1.append(i)

if (len(list1) > 0):

return 0

else:

return -1

Write an efficient algorithm for the following assumptions:

N is an integer within the range [0..200,000];

string S consists only of the following characters: "(", "{", "[", "]", "}" and/or ")".

Question 3

**Problem Description:**

You are given two non-empty arrays A and B consisting of N integers. Arrays A and B represent N voracious fish in a river, ordered downstream along the flow of the river.

The fish are numbered from 0 to N - 1. If P and Q are two fish and P < Q, then fish P is initially upstream of fish Q. Initially, each fish has a unique position.

Fish number P is represented by A[P] and B[P]. Array A contains the sizes of the fish. All its elements are unique. Array B contains the directions of the fish. It contains only 0s and/or 1s, where:

0 represents a fish flowing upstream,

1 represents a fish flowing downstream.

If two fish move in opposite directions and there are no other (living) fish between them, they will eventually meet each other. Then only one fish can stay alive ? the larger fish eats the smaller one. More precisely, we say that two fish P and Q meet each other when P < Q, B[P] = 1 and B[Q] = 0, and there are no living fish between them. After they meet:

If A[P] > A[Q] then P eats Q, and P will still be flowing downstream,

If A[Q] > A[P] then Q eats P, and Q will still be flowing upstream.

We assume that all the fish are flowing at the same speed. That is, fish moving in the same direction never meet. The goal is to calculate the number of fish that will stay alive.

For example, consider arrays A and B such that:

A[0] = 4 B[0] = 0

A[1] = 3 B[1] = 1

A[2] = 2 B[2] = 0

A[3] = 1 B[3] = 0

A[4] = 5 B[4] = 0

Initially all the fish are alive and all except fish number 1 are moving upstream. Fish number 1 meets fish number 2 and eats it, then it meets fish number 3 and eats it too. Finally, it meets fish number 4 and is eaten by it. The remaining two fish, number 0 and 4, never meet and therefore stay alive.

Write a program that, given two non-empty arrays A and B consisting of N integers, returns the number of fish that will stay alive. For example, given the arrays shown above, the function should return 2, as explained above. Write an efficient algorithm for the following assumptions:

N is an integer within the range [1..100,000];

each element of array A is an integer within the range [0..1,000,000,000];

each element of array B is an integer that can have one of the following values: 0, 1;

the elements of A are all distinct.

Question 4

**Problem Description:**

A string S consisting of N characters is called properly nested if:

S is empty;

S has the form "(U)" where U is a properly nested string;

S has the form "VW" where V and W are properly nested strings.

For example, string "(()(())())" is properly nested but string "())" isn't.

Write a program that, given a string S consisting of N characters, returns 1 if string S is properly nested and 0 otherwise.

For example, given S = "(()(())())", the function should return 1 and given S = "())", the function should return 0, as explained above.

Write an efficient algorithm for the following assumptions:

N is an integer within the range [0..1,000,000];

string S consists only of the characters "(" and/or ")".

Question 5

**Problem Description:**

You are going to build a stone wall. The wall should be straight and N meters long, and its thickness should be constant; however, it should have different heights in different places. The height of the wall is specified by an array H of N positive integers. H[I] is the height of the wall from I to I+1 meters to the right of its left end. In particular, H[0] is the height of the wall's left end and H[N-1] is the height of the wall's right end.

The wall should be built of cuboid stone blocks (that is, all sides of such blocks are rectangular). Your task is to compute the minimum number of blocks needed to build the wall.

Write a program that, given an array H of N positive integers specifying the height of the wall, returns the minimum number of blocks needed to build it.

For example, given array H containing N = 9 integers:

H[0] = 8 H[1] = 8 H[2] = 5

H[3] = 7 H[4] = 9 H[5] = 8

H[6] = 7 H[7] = 4 H[8] = 8

the function should return 7. The figure shows one possible arrangement of seven blocks.

Write an efficient algorithm for the following assumptions:

N is an integer within the range [1..100,000];

each element of array H is an integer within the range [1..1,000,000,000].