

We love cakes, how about you? Let's have cake-focused homework! >3

Note: For all these problems, you do not need to actually print the result(s). You just need to put your solution accordingly in the functions presented in hw8.cpp in starter_code.

(We give you some example inputs in example_input.txt, where the example_output.txt has all corresponding example outputs)

1. Cake Cutting

Suppose you are a cake smuggler, who buys cake in a local store, cuts it into pieces, and sells them to your customers. Different customers have different preferences: some of them would pay a certain price for a small piece of cake, while others might pay higher prices for a larger piece of cake. As a smuggler, you want to make sure that you always earn the highest profit possible.

For simplicity, we are going to uniform the size of a cake or a piece of cake to 1. Also, we are going to ignore the price you pay for buying the cake.

Here is an example:

Size (of piece) of cake	1	2	3	4
Price	1	3	5	4

Now suppose you just bought a cake from a local store, which has a size of 4. According to the information above, the highest profit you can make is 6.

Your program should take one input string argument, which is something like this: "1, 3, 5, 4", which represents all possible prices of the above table. The size of your cake (s) is determined by the number of possible prices, where in this case, the size of your cake is 4.

Here is another example:

Passing in "1, 3, 5, 3, 8, 9" means your cake has a size of 6:

Size (of piece) of cake	1	2	3	4	5	6
Price	1	3	5	3	8	9

Your program should print a single number to the console, which is the highest profit you can get. In this case, it should be: "10".

Hint: Do not use naive solutions!

Spoiler:

2. Cake Carrying

Suppose that you are a cake smuggler who used to buy cake from a local store. You have found another store which provides a better deal. But this store is one hour away. To save your driving time, you decide to buy multiple cakes at a time. You have a fridging box to store these cakes. However, Your fridging box has a fixed capacity in terms of weight. You have several cakes that you want to buy. Each cake has a weight. Unfortunately, your fridging box does not have the capacity to store all the cakes. You have to select a few cakes to take. To help you choose, every cake has been assigned a potential profit score. You can either take a full cake or you have to leave it. You cannot take a partial cake. You want to maximize the total potential profit score of the selected cakes. You also have to ensure that the total weight of the selected cakes do not exceed the capacity of the fridging box.

Here is an example of the cakes:

Item	Weight	Potential Profit Score
Vanilla Cake	2	3
Chocolate Cake	3	4
Lemon Cheese Cake	4	5
Red Velvet Cake	5	6

Suppose that the capacity of your fridging box is 5. According to the information above, you should select Vanilla cake and Chocolate cake to buy. Their combined weight is 5 which is within the capacity of the fridging box. Their combined Potential Profit Score is 7 which is highest among all possible valid combinations.

Your program should take three input string arguments. First string represents the capacity (C) of the fridging box. Second and third strings are something like this: "2, 3, 4, 5" and "3, 4, 5, 6". The second string represents the weights of cakes and the third string represents the Potential Profit Score of the items. Second and third strings would have the same number of elements (n).

Your program should print a single number to the console, which is the highest Potential Profit Score you can get. In this case, it should be: "7".

Here is another example:

You have inputs: "50", "10, 20, 30" and "60, 100, 120".

Your program will print "220" to the console as it is the highest possible potential profit score among valid combinations.

Hint: Do not use naive solutions!

Spoiler:

3. Cake Offering

Suppose now you have made enough money and are no longer a cake smuggler. As you have cut so many cakes, you are now a professional cook, who specializes in cutting cakes! Before actually cutting the cake, you are wondering about how many ways you can cut a cake.

For simplicity, we are going to uniform the size of a cake or a piece of cake to 1.

Here is an example:

You have a cake with size of 4, where you can cut it into three different sizes: {1, 2, 3}. The solution of this is that you can cut it in a total of 4 ways: {1,1,1,1}, {1,1,2}, {2,2}, {1,3}.

Your program needs to take two string arguments as input. The first argument will be the size of the cake, where the second argument will be something like this: "1, 2, 3". You will want your program to print the number of ways of cutting that certain cake into the console.

Here is another example:

You have inputs: "10" and "2, 4, 6, 3, 5".

Your program will print "9" to the console as there are 9 ways to cut this size of 10's cake.

Hint: Do not use naive solutions!

Spoiler:

-