Problem E. Hamburgers

Time limit 1000 ms
Mem limit 262144 kB

Polycarpus loves hamburgers very much. He especially adores the hamburgers he makes with his own hands. Polycarpus thinks that there are only three decent ingredients to make hamburgers from: a bread, sausage and cheese. He writes down the recipe of his favorite "Le Hamburger de Polycarpus" as a string of letters 'B' (bread), 'S' (sausage) и 'C' (cheese). The ingredients in the recipe go from bottom to top, for example, recipe "BSCBS" represents the hamburger where the ingredients go from bottom to top as bread, sausage, cheese, bread and sausage again.

Polycarpus has n_b pieces of bread, n_s pieces of sausage and n_c pieces of cheese in the kitchen. Besides, the shop nearby has all three ingredients, the prices are p_b rubles for a piece of bread, p_s for a piece of sausage and p_c for a piece of cheese.

Polycarpus has r rubles and he is ready to shop on them. What maximum number of hamburgers can he cook? You can assume that Polycarpus cannot break or slice any of the pieces of bread, sausage or cheese. Besides, the shop has an unlimited number of pieces of each ingredient.

Input

The first line of the input contains a non-empty string that describes the recipe of "Le Hamburger de Polycarpus". The length of the string doesn't exceed 100, the string contains only letters 'B' (uppercase English B), 'S' (uppercase English S) and 'C' (uppercase English C).

The second line contains three integers n_b, n_s, n_c $(1 \le n_b, n_s, n_c \le 100)$ — the number of the pieces of bread, sausage and cheese on Polycarpus' kitchen. The third line contains three integers p_b, p_s, p_c $(1 \le p_b, p_s, p_c \le 100)$ — the price of one piece of bread, sausage and cheese in the shop. Finally, the fourth line contains integer r $(1 \le r \le 10^{12})$ — the number of rubles Polycarpus has.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print the maximum number of hamburgers Polycarpus can make. If he can't make any hamburger, print `0`.

Sample 1

Input	Output
BBBSSC 6 4 1 1 2 3 4	2

Sample 2

Input	Output
BBC 1 10 1 1 10 1 21	7

Sample 3

Input	Output
BSC 1 1 1 1 1 3 1000000000000	200000000001