

B - Foreign Exchange Editorial by en_translator

For $i = 1, 2, \dots, N - 1$ in this order, exchange the currency of country i for that of country $i + 1$ as many times as possible.

Since exchange can be done a plethora of times, processing them one by one does not finish in the execution time limit. However, if you have A_i units of the currency of country i , you can exchange the currency of country i for that of country $(i + 1)$ $\lfloor A_i/S_i \rfloor$ times, so one can perform the exchange $\lfloor A_i/S_i \rfloor$ at once in order to obtain $\lfloor A_i/S_i \rfloor \times T_i$ units of the currency of country $(i + 1)$.

Therefore, the problem can be solved by receiving $A = (A_1, A_2, \dots, A_N)$ as input, adding $\lfloor A_i/S_i \rfloor \times T_i$ to A_{i+1} for $i = 1, 2, \dots, N - 1$ in this order, and then printing the value A_N .

In order to perform some process “for $i = 1, 2, \dots, N - 1$ in this order,” one can use the loop feature (like a `for` statement), which is a standard feature in a programming language. Also check out the specification of division in your language to correctly obtain the result of division A_i/S_i rounded down.

The following is sample code in C++ language.

```
#include <iostream>
using namespace std;
typedef long long ll;
5. ll n;
   ll a[200001], s[200001], t[200001];
   int main(void)
   {
10.  cin >> n;
      for(int i = 1; i <= n; i++) cin >> a[i];
      for(int i = 1; i <= n-1; i++) cin >> s[i] >> t[i];

      for(int i = 1; i <= n-1; i++) a[i+1] += a[i]/s[i] * t[i];
15.  cout << a[n] << endl;

      return 0;
   }
```

[Copy](#)