

## Zadanie „Kasjer”



### #zmienne\_lokalne

W zadaniu *Kasjer* mamy wyznaczyć, ile papierosów może wypalić kasjer podczas przerw w obsłudze klientów. W danych wejściowych znajduje się liczba klientów  $n$ , całkowity czas pracy kasjera  $L$  oraz czas potrzebny na wypalenie jednego papierosa  $a$ . Ponadto podane jest  $n$  par liczb  $(t_i, l_i)$  określających odpowiednio moment pojawienia się klienta  $t_i$  oraz czas jego obsługi  $l_i$  ( $i = 1, 2, \dots, n$ ).

Główny problem polega na wyznaczeniu długości kolejnych przerw pomiędzy klientami. Przerwa trwa od momentu zakończenia obsługi poprzedniego klienta (oznaczymy tę chwilę przez  $t_p$ ), aż do rozpoczęcia obsługi następnego klienta, czyli  $t_i$ ,  $i = 1, 2, \dots, n$ . Przechodząc do następnej przerwy zmiennej  $t_p$  nadajemy wartość  $t_i + l_i$ .

Aby opis układu przerw był kompletny, musimy dodać jeszcze przerwę *przed* obsłużeniem pierwszego klienta (wystarczy na początku przyjąć  $t_p = 0$ ) oraz przerwę *po* obsłużeniu ostatniego klienta. Układ przerw przedstawia się zatem następująco:

- $t_p := 0$ ;
- Przerwa pierwsza:  $(t_p, t_1)$   
Długość =  $t_1 - t_p$ ,  $t_p := t_1 + l_1$ ;
- Przerwa druga:  $(t_p, t_2)$   
Długość =  $t_2 - t_p$ ,  $t_p := t_2 + l_2$ ;
- Przerwa trzecia:  $(t_p, t_3)$   
Długość =  $t_3 - t_p$ ,  $t_p := t_3 + l_3$ ;
- ...
- Przerwa  $n$ -ta:  $(t_p, t_n)$   
Długość =  $t_n - t_p$ ,  $t_p := t_n + l_n$ ;
- Przerwa  $(n + 1)$ -sza:  $(t_p, L)$   
Długość =  $L - t_p$ .

Użyliśmy tutaj symbolu  $:=$ , który w literaturze algorytmicznej jest często używany do oznaczenia instrukcji przypisania (nadania wartości).\*

Długości przerw podzielone przez czas palenia jednego „szluga” (a dokładniej: *podłoga* z tego dzielenia) dają po zsumowaniu ilość wypalonych papierosów  $n_a$ .

---

\*Symbol ten używany był na przykład w językach programowania Algol i Pascal.

## Kod programu

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n, L, a;
    cin >> n >> L >> a;
    int t_p{ }, n_a{ };
    while(n--)
    {
        int t, l;
        cin >> t >> l;
        n_a += (t - t_p) / a;
        t_p = t + l;
    }
    n_a += (L - t_p) / a;
    cout << n_a << '\n';
    return 0;
}
```

Zmienne  $t$  oraz  $l$  w programie nie mają indeksów, gdyż ich wartość jest potrzebna tylko podczas danego obiegu pętli – dlatego też zadeklarowane są lokalnie (wewnątrz pętli `while`). Po zakończeniu pętli musimy jeszcze dodać ilość wypalonych papierosów podczas ostatniej przerwy: po ostatnim kliencie. Zwróćmy też uwagę na warunek kontynuacji pętli – liczba obiegów pętli jest równa wczytanej wartości zmiennej  $n$ , ponieważ wartość różna od zera jest traktowana w tym kontekście jako wartość prawdy, a wartość zero oznacza fałsz.