PREPARE CERTIFY COMPETE

Q Search

⊐ Û

junaediakbar8 Y

All Contests > Praktikum Modul 4 - Struktur Data B 2021 / 2022 > Server Utas Tunggal

# Server Utas Tunggal

locked

Problem

Submissions

Sebagai seorang calon programmer handal Anda diminta mensimulasikan alur kerja Single-Threaded Server. Ada n query untuk diproses, ke-i akan diterima pada waktu  $t_i$  dan perlu diproses dalam satuan waktu. Semua  $t_i$  dijamin beda.

Saat *query* muncul, server dapat bereaksi dalam tiga cara yang mungkin:

1. Jika server kosong dan antrian query kosong, maka server segera mulai memproses query ini.

Discussions

- 2. Jika server sibuk dan ada kurang dari b query dalam antrian, maka query baru ditambahkan ke akhir antrian.
- 3. Jika server sedang sibuk dan sudah ada b query yang menunggu dalam antrian, maka query baru tersebut ditolak dan tidak akan pernah diproses.

Segera setelah server selesai memproses beberapa *query*, ia akan mengambil *query* baru dari antrian (tentu saja jika tidak kosong). Jika *query* baru tiba pada waktu x, dan server selesai memproses *query* lain pada waktu yang sama persis, kita dapat berasumsi bahwa *query* pertama diambil dari antrian dan baru kemudian *query* baru muncul.

Untuk setiap query, cari kapan server akan selesai memprosesnya atau cetak -1 jika query tersebut akan ditolak.

#### NOTE

Diwajibkan menggunakan STL yang telah ada Modul STL

# **Input Format**

Baris pertama dari input berisi dua bilangan bulat n dan b — jumlah query dan ukuran maksimum yang mungkin dari antrian query.

Kemudian ikuti n baris dengan deskripsi query (dalam urutan kronologis). Setiap deskripsi terdiri dari dua bilangan bulat  $t_i$  dan  $d_i$  ( $1 \le t_i, \ d_i \le 10^9$ ), di mana  $t_i$  adalah waktu saat query ke-i muncul dan  $d_i$  adalah waktu yang dibutuhkan server untuk memprosesnya. Dijamin bahwa  $t_{i-1} < t_i$  untuk semua i > 1.

# Constraints

- $(1 \le n, b \le 200000)$
- $(1 \le t_i, d_i \le 10^9)$

#### **Output Format**

Cetak urutan n bilangan bulat  $e_1, e_2, \ldots, e_n$ , di mana  $e_i$  adalah saat server akan selesai memproses *query* ke-i (*query* diberi nomor sesuai urutan kemunculannya di input) atau -1 jika *query* yang sesuai akan ditolak.

# Sample Input 0

- 5 1
- 2 9
- 4 8
- 10 9

15 2 19 1

#### Sample Output 0

11 19 -1 21 22

#### **Explanation 0**

Catatan Pertimbangkan sampel pertama.

- 1. Server akan mulai memproses kueri pertama pada saat 2 dan akan selesai memprosesnya pada saat 11.
- 2. Saat ini kueri 4 detik muncul dan melanjutkan ke antrian.
- 3. Saat ini 10 permintaan ketiga muncul. Namun, server masih sibuk dengan permintaan 1, b = 1 dan sudah ada permintaan 2 yang tertunda dalam antrian, sehingga permintaan ketiga ditolak saja.
- 4. Saat ini 11 server akan selesai memproses query pertama dan akan mengambil query kedua dari antrian.
- 5. Saat ini 15 kueri keempat muncul. Karena server sedang sibuk, ia melanjutkan ke antrian.
- 6. Saat ini 19 dua peristiwa terjadi secara bersamaan: server selesai untuk melanjutkan permintaan kedua dan permintaan kelima muncul. Seperti yang dikatakan dalam pernyataan di atas, server pertama akan selesai memproses kueri kedua, kemudian akan memilih kueri keempat dari antrian dan baru kemudian kueri kelima akan muncul. Karena antrian kosong, permintaan kelima dilanjutkan di sana.
- 7. Server selesai memproses kueri nomor 4 pada saat 21. Nomor kueri 5 diambil dari antrian.
- 8. Server selesai memproses permintaan nomor 5 pada saat 22.

# Sample Input 1

4 1

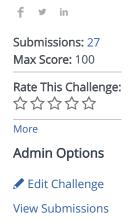
2 8

4 8 10 9

15 2

# Sample Output 1

10 18 27 -1





```
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

/* Enter your code here. Read input from STDIN. Print output to STDOUT */
return 0;
}

Line: 1 Col: 1
```

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature