



## B. 爬山

时间限制：1.0 s 内存限制：512 MB

小 C 喜欢爬山，但他的体能一般，不能连续爬升太高。

这天小 C 决定去爬比特峰。比特峰共有  $n$  个景点，第  $i$  个景点的海拔为  $h_i$  米。

景点之间共有  $m$  条双向线路，第  $i$  条线路连通景点  $u_i$  和景点  $v_i$ ，小 C 从任意方向走过第  $i$  条线路所需的时长均为  $t_i$  分钟。

小 C 在爬山过程中会累积疲劳值，初始的疲劳值为 0，并且小 C 能够忍受的最大疲劳值为  $H$ 。

假设当前小 C 位于某个景点  $x$ ，且存在一条从景点  $x$  到景点  $y$  的线路：

- 如果景点  $x$  比景点  $y$  的海拔高，那么这是一段轻松的下坡路，小 C 如果选择走这条线路，体力得到了恢复，疲劳值被清空。
- 否则，如果小 C 选择走这条线路，疲劳值会累积  $h_y - h_x$ 。小 C 只能在保证积累的疲劳值不超过  $H$  的前提下走这条线路。

现在小 C 从 1 号景点出发，只能通过这些线路在景点间移动。对于每个景点，小 C 想知道他最快需要多少分钟可以到达该景点，或永远无法到达？

### Input

第一行包含三个正整数  $n, m, H$  ( $2 \leq n \leq 10^4$ ,  $0 \leq m \leq 2 \times 10^4$ ,  $1 \leq H \leq 100$ )，分别表示景点的个数，线路的条数，以及小 C 能忍受的最大疲劳值。

第二行包含  $n$  个正整数  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 10^9$ )，代表每个景点的海拔。

接下来  $m$  行，每行三个正整数  $u_i, v_i, t_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq t_i \leq 10^9$ )，代表一条线路连接的两个景点和所需的时长。

### Output

输出一行  $n - 1$  个整数，第  $i$  个整数代表小 C 到达第  $i + 1$  号景点所需的最短时间（以分钟为单位）。如果小 C 永远无法到达该景点，输出  $-1$ 。

#### Sample Input 1

```
6 7 3
1 2 3 4 5 6
1 2 1
2 4 10
1 3 8
2 3 2
3 5 9
4 3 100
2 6 2
```

#### Sample Output 1

```
1 3 11 16 -1
```