

## Mendeteksi Molekul

Petr bekerja pada sebuah perusahaan yang membuat sebuah mesin untuk mendeteksi himpunan molekul. Tiap molekul mempunyai bobot berupa bilangan integer positif. Mesin tersebut mempunyai suatu *rentang pendeteksian*  $[l, u]$ , dengan  $l$  dan  $u$  berupa bilangan integer positif. Mesin dapat mendeteksi suatu himpunan molekul jika dan hanya jika himpunan ini memuat sebuah subhimpunan molekul dengan total bobot sesuai dengan rentang pendeteksian tersebut.

Secara formal, terdapat  $n$  molekul dengan bobot berupa bilangan integer positif  $w_0, \dots, w_{n-1}$ . Pendeteksian dikatakan berhasil jika terdapat sebuah himpunan dengan indeks-indeks yang saling berbeda  $I = \{i_1, \dots, i_m\}$  sedemikian sehingga  $l \leq w_{i_1} + \dots + w_{i_m} \leq u$ .

Berdasarkan spesifikasi mesin di atas, maka selisih antara  $l$  dan  $u$  dijamin lebih besar atau sama dengan selisih bobot antara molekul terberat dan teringan. Secara formal,  $u - l \geq w_{max} - w_{min}$ , dengan  $w_{max} = \max(w_0, \dots, w_{n-1})$  dan  $w_{min} = \min(w_0, \dots, w_{n-1})$ .

Tugas Anda adalah menuliskan sebuah program untuk menentukan apakah ada sebuah subhimpunan molekul sebarang dengan total bobot berada pada rentang pendeteksian yang sudah ditentukan atau tidak.

### Rincian Implementasi

Anda harus mengimplementasikan sebuah fungsi (method):

- `int[] solve(int l, int u, int[] w)`
  - $l$  dan  $u$ : titik-titik ujung suatu rentang pendeteksian,
  - $w$ : himpunan bobot dari molekul.
  - jika terdapat subhimpunan yang sesuai dengan spesifikasi mesin, maka fungsi tersebut harus mengembalikan sebuah array berisi indeks dari molekul-molekul yang membentuk subhimpunan tersebut. Jika terdapat beberapa jawaban yang benar, maka fungsi tersebut dapat mengembalikan sebarang jawaban.
  - jika tidak terdapat satupun subhimpunan yang memenuhi spesifikasi tersebut, maka fungsi tersebut harus mengembalikan sebuah array kosong.

Program Anda dapat menuliskan indeks dengan urutan sebarang pada array pengembalian.

Gunakan file template yang sudah disediakan untuk implementasi rinci dari bahasa

pemrograman yang Anda pakai.

## Contoh

### Contoh 1

```
solve(15, 17, [6, 8, 8, 7])
```

Pada contoh diatas terdapat empat molekul dengan bobot 6, 8, 8 dan 7. Mesin dapat mendeteksi subhimpunan dengan total bobot antara 15 and 17, inklusif. Catatan,  $17 - 15 \geq 8 - 6$ . Total bobot molekul 1 dan 3 is  $w_1 + w_3 = 8 + 7 = 15$ , sehingga fungsi dan mengembalikan `[1, 3]`. Kemungkinan jawaban benar yang lain adalah `[1, 2]` ( $w_1 + w_2 = 8 + 8 = 16$ ) dan `[2, 3]` ( $w_2 + w_3 = 8 + 7 = 15$ ).

### Contoh 2

```
solve(14, 15, [5, 5, 6, 6])
```

Pada contoh diatas terdapat empat molekul dengan bobot 5, 5, 6 dan 6, dan kita akan mencari subhimpunan dengan total bobot antara 14 dan 15, inklusif. Catatan,  $15 - 14 \geq 6 - 5$ . Tidak terdapat satupun subhimpunan molekul dengan total bobot antara 14 dan 15 sehingga fungsi harus mengembalikan sebuah array kosong.

### Contoh 3

```
solve(10, 20, [15, 17, 16, 18])
```

Pada contoh diatas terdapat empat molekul dengan bobot 15, 17, 16 dan 18, dan kita akan mencari subhimpunan dengan total bobot antara 10 dan 20, inklusif. Catatan,  $20 - 10 \geq 18 - 15$ . Sebarang subhimpunan yang mempunyai tepat satu elemen memiliki jumlah bobot antara 10 dan 20, sehingga kemungkinan jawaban yang benar adalah: `[0]`, `[1]`, `[2]` dan `[3]`.

## Subtask

- (9 poin):  $1 \leq n \leq 100$ ,  $1 \leq w_i \leq 100$ ,  $1 \leq u, l \leq 1000$ , semua  $w_i$  sama.
- (10 poin):  $1 \leq n \leq 100$ ,  $1 \leq w_i, u, l \leq 1000$  dan  $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$ .
- (12 poin):  $1 \leq n \leq 100$  dan  $1 \leq w_i, u, l \leq 1000$ .
- (15 poin):  $1 \leq n \leq 10000$  dan  $1 \leq w_i, u, l \leq 10000$ .
- (23 poin):  $1 \leq n \leq 10000$  dan  $1 \leq w_i, u, l \leq 500000$ .
- (31 poin):  $1 \leq n \leq 200000$  dan  $1 \leq w_i, u, l < 2^{31}$ .

## Grader

Grader membaca masukan dengan format berikut:

- baris 1 : bilangan integer  $n$ ,  $l$ ,  $u$ .
- baris 2 :  $n$  bilangan integer :  $w_0, \dots, w_{n-1}$ .