

## Tugas Pemrograman 1

### Dylan dan Milya

Nama Berkas Kode Sumber : SDA18191T.java  
Batas waktu eksekusi program : 3 detik / kasus uji  
Batas memori program : 256 MB / kasus uji

*“Aku ramal, nanti kita akan bertemu di kantin”*

Milya bertemu dengan Dylan di sebuah SMA di Depok. Saat itu tahun 1999, ketika Milya pindah dari Jakarta ke Depok. Perkenalan yang tidak biasa kemudian membawa Milya mulai mengenal keunikan Dylan lebih jauh. Dylan pintar, baik hati, dan romantis namun dianggap sebagai berandal karena aktif di geng motor terkenal di Depok sebagai panglima tempur. Cara Dylan mendekati Milya tidak sama dengan teman-teman pria lainnya. Cara Dylan berbicara yang terdengar baku, lambat laun membuat Milya kerap merindukannya jika sehari saja tak mendengar suara itu. Hari demi hari hubungan Dylan dan Milya semakin akrab hingga akhirnya mereka berpisah ketika lulus SMA. Sebelum berpisah, mereka berjanji akan bertemu kembali setelah lima tahun.

Lima tahun yang berat telah mereka lalui, tak satu hari pun mereka melupakan masa-masa indah ketika SMA. Hari yang ditunggu pun tiba, Dylan bersiap untuk menemui Milya di tempat yang mereka janjikan lima tahun silam. Ternyata, Milya telah tiba terlebih dahulu di tempat perjanjian. Karena terlambat, Dylan bergegas berangkat menuju Milya. Dylan dan Milya saat ini terpisah sejauh **S meter**. Dylan berpindah dengan menggunakan salah satu dari **N transportasi yang ada tiap detiknya (waktu pergantian transportasi diabaikan)**. Transportasi ke  $i$  memiliki kecepatan  $v[i]$  meter/detik dan tiap transportasi memiliki kecepatan yang **unik**. Dylan meminta Milya untuk menunggu di tempat perjanjian. Namun, karena Milya tidak sabar untuk bisa bertemu dengan Dylan, **ia juga berjalan menuju Dylan dengan kecepatan U meter/detik (jika  $U = 0$ , maka Milya diam)**.

Dylan dan Milya dikatakan **bertemu jika dan hanya jika mereka berada di titik yang sama pada detik yang sama (detik merupakan bilangan bulat)**. Perhatikan bahwa mereka dapat melewati satu sama lain tanpa bertemu di titik yang sama sehingga mereka tidak akan pernah bertemu.

Kalian (sebagai sahabat Dylan dan Milya) yang sedang kuliah di Fasilkom UI, penasaran dan ingin memodelkan beberapa hal, yaitu:

- **Ada berapa cara berbeda Dylan dan Milya menggunakan alat transportasi sehingga mereka bisa bertemu?**
- **Berapa waktu minimal yang dapat ditempuh Dylan untuk bertemu Milya?**

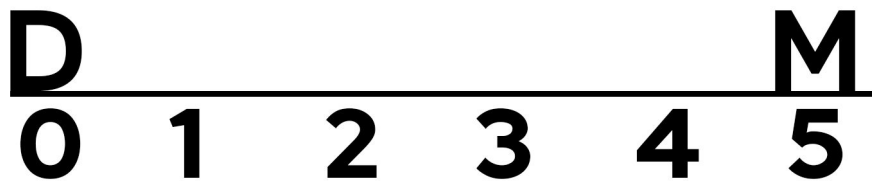
## Ilustrasi

### Contoh Kasus 1

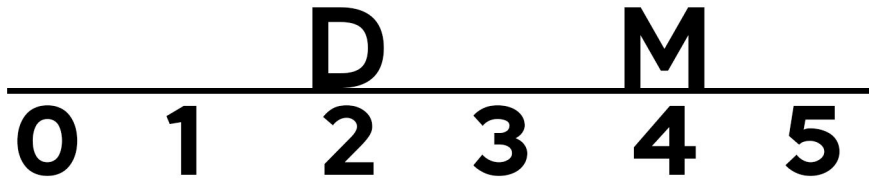
Dylan dan Milya terpisah sejauh 5 meter. Dylan dapat berjalan dengan kecepatan 1 meter/detik, naik ojek dengan kecepatan 2 meter/detik, atau naik angkot dengan kecepatan 3 meter/detik. Milya dapat berjalan mendekati Dylan dengan kecepatan 1 meter/detik.

Pada contoh tersebut,  $S = 5$ ,  $N = 3$ ,  $V[1] = 1$ ,  $V[2] = 2$ ,  $V[3] = 3$ ,  $U = 1$ .

Kondisi Awal



Pada detik pertama, misalkan Dylan berkendara sejauh 2 meter



Kemudian detik kedua, Dylan berjalan sejauh 1 meter



Pada ilustrasi di atas, detik kedua menunjukkan bahwa Dylan dan Milya bertemu pada titik yang sama, yaitu titik tiga. Cara yang ditempuh oleh Dylan untuk bisa mencapai titik tersebut adalah (2, 1) dibaca dua kemudian satu.

Adapun dengan kondisi pada ilustrasi di atas, banyak cara yang dapat ditempuh Dylan untuk menemui Milya adalah **DUA** cara yaitu (1, 2) dan (2, 1).

Dylan tidak akan berkendara sejauh 3 karena ia tidak akan bertemu dengan Milya pada titik yang sama apabila ia berkendara secepat itu.

## Contoh Kasus 2

Dylan dan Milya terpisah sejauh 12 meter. Dylan dapat berjalan dengan kecepatan 1 meter/detik, naik ojek dengan kecepatan 3 meter/detik, atau naik angkot dengan kecepatan 4 meter/detik. Milya dapat berjalan mendekati Dylan dengan kecepatan 1 meter/detik.

Pada contoh tersebut,  $S = 12$ ,  $N = 3$ ,  $V[1] = 1$ ,  $V[2] = 3$ ,  $V[3] = 4$ ,  $U = 1$ .

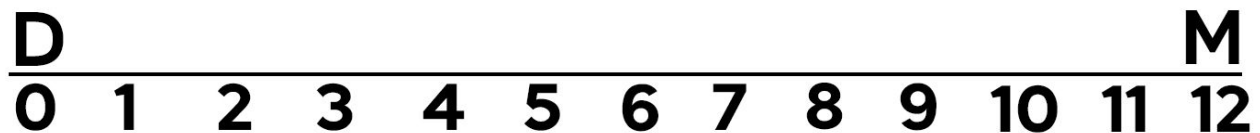
Ada **16** cara berbeda bagi Dylan untuk bisa menemui Milya pada titik yang sama.

1. (1, 4, 4)

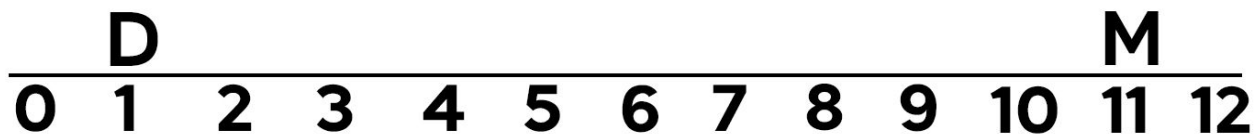
Dylan berjalan di detik pertama, kemudian naik angkot selama dua detik.

Cara ini dideskripsikan dalam visualisasi berikut

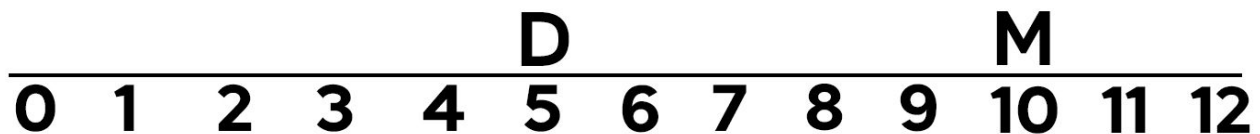
Kondisi Awal



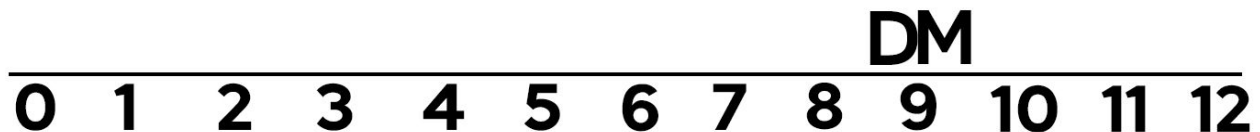
Pada detik pertama, misalkan Dylan berjalan sejauh 1 meter



Kemudian detik kedua, Dylan berkendara sejauh 4 meter



Kemudian detik ketiga, Dylan berkendara sejauh 4 meter



Pada ilustrasi di atas, detik ketiga menunjukkan bahwa Dylan dan Milya bertemu pada titik yang sama, yaitu titik sembilan. Cara yang ditempuh oleh Dylan untuk bisa mencapai titik tersebut adalah (1, 4, 4) dibaca Dylan berjalan selama satu detik, kemudian naik angkot selama dua detik sampai bertemu Milya.

Selain (1,4,4) Dylan juga bisa bertemu dengan Milya pada titik yang sama dengan langkah:

2. (3, 3, 3)

Dylan naik ojek dengan kecepatan konstan 3 meter/detik. 3 detik kemudian, Dylan berhasil menemui Milya.

3. (4, 1, 4)  
Dylan naik angkot selama satu detik, kemudian berjalan selama satu detik, kemudian naik angkot selama satu detik hingga akhirnya Dylan berhasil menemui Milya.
4. (4, 4, 1)  
Dylan naik angkot selama dua detik, kemudian dilanjutkan berjalan selama satu detik hingga akhirnya Dylan berhasil menemui Milya.
5. (1, 1, 3, 3)  
Dylan naik berjalan selama dua detik selama dua detik, kemudian dilanjutkan naik ojek selama dua detik hingga akhirnya Dylan berhasil menemui Milya.
6. (1, 3, 1, 3)  
Dylan berjalan selama satu detik, kemudian dilanjutkan naik ojek selama satu detik, kemudian berjalan selama satu detik. Setelah itu Dylan naik ojek selama satu detik hingga akhirnya Dylan berhasil menemui Milya.
7. (1, 3, 3, 1)  
Dylan berjalan di detik pertama, kemudian dilanjutkan dengan naik ojek di dua detik berikutnya, lalu berjalan selama satu detik hingga akhirnya bertemu dengan Milya.
8. (3, 1, 1, 3)  
Dylan naik ojek selama satu detik, kemudian dilanjutkan dengan berjalan selama dua detik, lalu naik ojek selama satu detik hingga akhirnya bertemu dengan Milya.
9. (3, 1, 3, 1)  
Dylan naik ojek selama satu detik, kemudian dilanjutkan dengan berjalan selama satu detik, lalu naik ojek selama satu detik, kemudian dilanjutkan dengan berjalan selama satu detik hingga akhirnya bertemu dengan Milya.
10. (3, 3, 1, 1)  
Dylan naik ojek selama dua detik, kemudian dilanjutkan dengan berjalan selama dua detik hingga akhirnya bertemu dengan Milya.
11. (1, 1, 1, 1, 3)  
Dylan berjalan selama empat detik, kemudian dilanjutkan dengan naik ojek sampai menemui Milya.
12. (1, 1, 1, 3, 1)  
Dylan berjalan selama tiga detik, kemudian naik ojek selama satu detik, lalu berjalan selama satu detik sampai menemui Milya.
13. (1, 1, 3, 1, 1)  
Dylan berjalan selama dua detik, kemudian naik ojek selama satu detik, kemudian berjalan selama dua detik sampai menemui Milya.
14. (1, 3, 1, 1, 1)  
Dylan berjalan selama satu detik, kemudian naik ojek selama satu detik, kemudian berjalan selama tiga detik sampai menemui Milya.
15. (3, 1, 1, 1, 1)  
Dylan naik ojek selama satu detik, kemudian berjalan selama empat detik sampai menemui Milya.

16. (1, 1, 1, 1, 1, 1)

Dylan berjalan selama enam detik sampai menemui Milya.

Dari 16 cara yang dipaparkan di atas, waktu tempuh yang dibutuhkan oleh Dylan bervariasi. Ada cara yang membutuhkan waktu 3 detik (cara no 1 - 4), 4 detik (cara no 5 - 10), 5 detik (cara no 11 - 15), dan 6 detik (cara no 16).

Jika Anda ingin membantu Dylan menemui Milya dalam waktu seminimal mungkin, cara yang diusulkan adalah cara no 1, 2, 3, atau 4.

## Definisi Problem

Setiap testcase mengandung salah satu dari dua permasalahan berikut :

- A. Menghitung banyaknya cara Dylan menggunakan alat transportasi sehingga ia dapat bertemu dengan Milya.

Karena banyaknya cara mereka bertemu dapat bernilai sangat besar, **outputkan banyaknya cara setelah dimodulo dengan 1000000007. Hati-hati dengan kasus penjumlahan yang melewati batas integer sebelum di-modulo.** Mohon untuk mengingat kembali materi aritmatika modulo agar tidak salah dalam mengoperasikannya.

Contoh : Jika banyaknya cara bernilai 1000000010, maka jawaban yang dioutputkan adalah 3.

- B. Menghitung waktu minimal yang dapat ditempuh Dylan untuk bertemu dengan Milya dan memberikan langkah-langkah yang dilakukan setiap detiknya. Apabila terdapat lebih dari satu konfigurasi langkah dengan waktu minimal, cetak konfigurasi langkah dengan **leksikografis terkecil.**

Misal terdapat dua konfigurasi langkah A dan B dengan  $A[i]$  dan  $B[i]$  masing-masing menyatakan langkah pada detik ke  $i$  pada konfigurasi langkah A dan B. Maka, A lebih kecil secara leksikografis dari B, jika dan hanya jika terdapat  $j$ , sehingga  $A[i] = B[i]$  untuk setiap  $i < j$  dan  $A[j] < B[j]$ .

Contoh : Pada ilustrasi ke-2 yang telah dipaparkan di atas, terdapat 4 konfigurasi langkah dengan waktu minimal yaitu  $= \{(1, 4, 4), (4, 1, 4), (4, 4, 1), (3, 3, 3)\}$ . Dengan definisi leksikografis di atas,  $(1, 4, 4)$  merupakan konfigurasi langkah dengan leksikografis terkecil

Grading system di Aren akan menyediakan sejumlah test case dengan tingkat kesulitan beragam (Easy, Medium, Hard). Gambaran kasar: 80% testcase memiliki kesulitan Easy atau Medium, sisanya 20% kesulitan Hard.

## Input Program

Masukan akan diberikan dengan format :

```
S N U
V[1] V[2] ... V[N]
P
```

Dengan catatan:

- $S$  = Jarak antara Dylan dengan Milya.
- $N$  = Jumlah transportasi berbeda yang dapat Dylan gunakan untuk berpindah tempat tiap detiknya.
- $U$  = Besar kecepatan yang dapat Milya tempuh untuk berpindah.
- $V[i]$  = Besar kecepatan yang dapat Dylan tempuh untuk berpindah pada transportasi tertentu.
- $P$  = Berisi 'A' atau 'B' yang menyatakan jenis permasalahan sesuai dengan penjelasan pada bagian Definisi Problem.

Batas masukan untuk test case dengan tingkat kesulitan easy / medium:

- $1 \leq S \leq 20$
- $1 \leq N \leq 4$
- $N \leq S$
- $0 \leq U < S$
- $0 < V[1] < V[2] < \dots < V[N] \leq S$

Batas masukan untuk test case dengan tingkat kesulitan hard:

- $1 \leq S \leq 3000$
- $1 \leq N \leq 100$
- $N \leq S$
- $0 \leq U < S$
- $0 < V[1] < V[2] < \dots < V[N] \leq S$

## Output Program

Untuk setiap testcase,

Jika  $P = 'A'$ , cetak:

$X$

- $X =$  Banyaknya cara Dylan berada pada titik yang sama dengan Milya. Apabila tidak ada cara yang memungkinkan bagi Dylan untuk bertemu dengan Milya maka tuliskan  $'0'$

Jika  $P = 'B'$ , cetak:

$Y$

$V[1] \ V[2] \ \dots \ V[Y]$

- $Y =$  Waktu minimal Dylan bertemu dengan Milya. Apabila tidak ada cara yang memungkinkan bagi Dylan untuk bertemu dengan Milya maka cetak  $'NA'$ .
- $V[1] \ V[2] \ \dots \ V[Y] =$  Konfigurasi langkah yang dapat diambil oleh Dylan dengan Waktu minimal. Apabila terdapat lebih dari satu konfigurasi langkah dengan waktu minimal, cetak konfigurasi langkah dengan **leksikografis terkecil** (seperti yang dijelaskan pada bagian Definisi Problem).



### Contoh Masukan 1

```
5 3 1
1 2 3
A
```

### Contoh Keluaran 1

```
2
```

### Penjelasan Contoh 1

Contoh sesuai dengan ilustrasi pertama yang telah dipaparkan sebelumnya.

Karena  $P = 'A'$  maka cukup output banyaknya cara yang memungkinkan bagi Dylan untuk bertemu dengan Milya. Dalam hal ini, artinya cukup cetak '2'

## Contoh Masukan 2

```
12 3 1  
1 3 4  
B
```

## Contoh Keluaran 2

```
3  
1 4 4
```

## Penjelasan Contoh 2

Contoh ini sesuai dengan ilustrasi kedua yang telah dipaparkan sebelumnya.

Dari keempat cara dengan waktu minimal yaitu :  $\{(1,4,4), (4, 1, 4), (4, 4, 1), (3, 3, 3)\}$ , program mencetak langkah  $(1,4,4)$  karena merupakan langkah dengan leksikografis terkecil.

### Contoh Masukan 3

6 4 2
1 2 3 4
B

### Contoh Keluaran 3

1
4

### Penjelasan Contoh 3

Karena  $P = 'B'$  maka kita perlu output waktu minimal yang dapat ditempuh Dylan untuk bisa bertemu dengan Milya di titik yang sama beserta variasi langkah yang digunakan dengan leksikografis terkecil. Untuk lebih rincinya adalah sebagai berikut:

1. Pada detik 0, Dylan dan Milya terpisah sejauh 6 meter.
2. Pada detik 1, misalkan Dylan berpindah sejauh 4 meter dan Milya berpindah sejauh 2 meter.
3. Ternyata, pada detik tersebut Dylan bertemu dengan Milya pada titik yang sama.
4. Dengan langkah lain, Dylan bisa saja bertemu dengan Milya pada titik 2 apabila ia berpindah dengan urutan (1, 1) yang terhitung sebagai 2 langkah. Namun, itu bukan waktu minimal karena terdapat cara dimana Dylan dapat bertemu dengan Milya hanya dengan 1 detik yaitu dengan langkah (4).

#### Contoh Masukan 4

6 1 3  
4  
A

#### Contoh Keluaran 4

0

#### Contoh Masukan 5

6 1 3  
4  
B

#### Contoh Keluaran 5

NA

#### Penjelasan Contoh 4 dan 5

Karena mereka tidak mungkin bertemu, sehingga output adalah '0' pada contoh masukan keempat dan 'NA' pada contoh masukan kelima.

# **Submission Tugas Pemrograman 1**

Dalam menyelesaikan Tugas Pemrograman ini, Anda bebas mengajukan solusi / algoritma apapun. Akan tetapi, agar dapat memperoleh nilai sempurna (dan nilai bonus):

- Menjawab test case dengan benar, dan
- Solusi Anda diimplementasikan dengan fungsi **rekursif**

Untuk menyelesaikan test case dengan tingkat kesulitan “hard”, solusi Anda mesti diimplementasikan dengan efisien (review prinsip-prinsip desain rekursif yang dipelajari di kelas, seperti **Divide and Conquer**, dan / atau **Dynamic Programming**).

Pada kode sumber, Mahasiswa diwajibkan menuliskan **dokumentasi** berupa *comment* pada tiap *method* dan *class* yang menjelaskan fungsi dari *method* dan *class* yang ada pada kode sumber.

Hal-hal yang dijadikan dasar penilaian pada Tugas Pemrograman ini terdiri dari:

- Kebenaran dan efisiensi logika dari solusi / algoritma
- Struktur penulisan program
- Automatic grader
- Dokumentasi atau penjelasan program

## **Batas waktu pengumpulan kode sumber:**

Senin, 8 Oktober 2018 pukul 23.55 Waktu Scele / Aren

## **Jangan menunda-nunda Pengerjaan Tugas!**

Tujuan dari Lab ini adalah melatih Anda agar menguasai bahan kuliah yang diajarkan di kelas. Mahasiswa diperbolehkan untuk berdiskusi, tetapi Anda tetap harus **menuliskan sendiri** solusi program tanpa bantuan orang lain. Belajarlah menjadi mahasiswa yang mematuhi integritas akademik. **Sikap Jujur merupakan sebuah sikap yang dimiliki mahasiswa Fasilkom UI.**

Mekanisme pengumpulan kode sumber akan diumumkan di SceLe. Kode sumber yang dikumpulkan selain melalui mekanisme yang disepakati akan **diabaikan** dan **dianggap tidak mengumpulkan**.

**Peringatan:** Jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!