



Perang Watatsumi

Deskripsi

Watatsumi adalah negeri air yang dipimpin oleh Dewi laut, **Sanagomiya Kokomi**. Saat ini sedang berlangsung perang 3 arah di negeri ini, yaitu antara pasukan laut yang dipimpin Kokomi, pasukan bumi yang dipimpin oleh Raja bebatuan, **Zhongli**, dan pasukan petir yang dipimpin oleh Dewi Awet Muda, **Raiden Shogun**. Saat ini, semua pemimpin perang yaitu Kokomi, Zhongli, dan Raiden Shogun sedang berada di Watatsumi. **Anda pengamat perang** yang akan menulis tentang peperangan ini.

Watatsumi terdiri dari N pulau. Setiap pulau terdiri dari beberapa dataran yang berjajar dari kiri ke kanan. Pulau ke- i bernama P_i dan memiliki D_i dataran, dengan dataran ke- j memiliki tinggi $A_{i,j}$ meter. Terdapat sebuah kuil di dataran paling kiri setiap pulau. Kuil ke- i memiliki **nama yang sama dengan nama awal pulau**. Misal nama pulauanya "SEIRAI", maka nama kuilnya "KUIL SEIRAI". Sebagai Presiden Watatsumi, Kokomi memiliki kontrol penuh terhadap semua pulau-pulau yang ada di Watatsumi. Kokomi dapat memberikan dua buah perintah yaitu **UNIFIKASI** dan **PISAH**. Dengan detail sebagai berikut:

1. **UNIFIKASI U V**: Menggabungkan pulau bernama **U** dan pulau bernama **V**
 - a. Gerakkan pulau **V** agar berada di belakang pulau **U**
 - b. Satukan pulau **U** dengan pulau **V**. Pulau baru tersebut bernama **U**.
 - c. Meski pulau **V** sudah tidak ada lagi, **nama kuil-kuil yang sebelumnya ada di V tidak berubah nama**. Misalkan **U** = "KANNAZUKA" dan di **V** ada kuil "SEIRAI" dan "NARUKAMI", maka setelah **V** digabung ke **U**, kuil-kuil yang sebelumnya ada di **V** tetap bernama "SEIRAI" dan "NARUKAMI".
2. **PISAH U**: Pisahkan pulau yang memiliki kuil **U** di bagian kiri kuil **U**.
 - a. Misalkan dataran pada **U** = [10, 20, 30, 40, 50, 60, 70] dengan kuil **U** berada di dataran dengan tinggi 40. Maka pulau ini dipisah menjadi 2, yaitu pulau dengan dataran [10, 20, 30], dan pulau dengan dataran [40, 50, 60, 70]
 - b. Nama pulau sebelah kiri sama seperti sebelumnya dan **nama pulau kanan yang baru terbentuk adalah U**.

Anda mencatat, untuk setiap **UNIFIKASI**, berapa banyak dataran yang dimiliki pulau baru dan untuk setiap **PISAH**, berapa banyak dataran yang dimiliki oleh pulau yang tidak memiliki kuil **U** dan pulau yang memiliki kuil **U**.

Raiden Shogun mulanya berada di dataran ke-**E** di pulau **R**. Raiden Shogun berpartisipasi secara aktif mengontrol alur perang agar pasukannya awet muda. Dia memiliki 3 buah pergerakan kilat, yaitu:

1. **GERAK [KIRI | KANAN] S**:

Raiden Shogun bergerak sebanyak **S** langkah ke kiri atau kanan dari dataran yang sekarang dipijak. Anda mencatat **tinggi dataran yang dipijak Raiden Shogun saat ini** setelah pergerakan ini dilakukan.

2. **TEBAS [KIRI|KANAN] S:**

Dengan pedangnya, Raiden Shogun bergerak secara lurus instan ke kiri/kanan, menebas semua musuh dan berhenti di **dataran terdekat ke-S dengan tinggi yang sama** seperti dataran sebelum menebas. Anda mencatat **tinggi dataran sebelumnya dari lokasi akhir Raiden Shogun bergerak**. Yakni, cetak dataran di kanan apabila **TEBAS KIRI**, dan cetak dataran di kiri jika **TEBAS KANAN**. Secara khusus, **cetak 0 apabila Raiden Shogun tidak bergerak dari tempat awal**.

3. **TELEPORTASI V:**

Raiden Shogun berpindah ke kuil **V**. Anda mencatat **tinggi dataran yang dipijak Raiden Shogun** setelah pergerakan ini.

Pada **GERAK** dan **TEBAS**, Raiden Shogun tidak harus bergerak sebanyak **S** langkah. Jika setelah bergerak beberapa langkah dia tidak dapat melanjutkan pergerakannya, misalnya:

- Saat ini berada di ujung kiri pulau dan sedang dilakukan GERAK KIRI, setelah sebelumnya melangkah ke kiri dua kali.
- Tinggi dataran posisi awal Raiden Shogun sebelum **TEBAS KANAN** 10 adalah 5 meter. Setelah menebas tiga kali, tidak ada lagi dataran dengan tinggi 5 meter di kanan Raiden Shogun.

Maka Raiden Shogun akan **memberhentikan pergerakannya**.

Zhongli, memiliki kontrol semua bebatuan di dunia, secara aktif mengontrol dataran-dataran di pulau-pulau Watatsumi untuk meningkatkan mobilitas pasukannya. Dia dapat melakukan empat buah manipulasi geografi yang luar biasa, yaitu:

1. **RISE U H X:** Semua dataran di pulau **U** yang **lebih tinggi dari H** akan dinaikkan tingginya sebanyak **X** meter. Anda mencatat **berapa dataran yang terdampak** dari manipulasi ini.
2. **QUAKE U H X:** Semua dataran di pulau **U** yang **lebih rendah dari H** akan diturunkan tingginya sebanyak **X**. Anda mencatat **berapa dataran yang terdampak** dari manipulasi ini.
3. **CRUMBLE:** Menghancurkan dataran tempat Raiden Shogun berada sekarang. Raiden Shogun kemudian pindah ke dataran **di sebelah kirinya**. Anda mencatat **tinggi dataran yang dihancurkan Zhongli**. Jika penghancuran dataran menyebabkan pulau terpisah, **pulau akan disatukan kembali** dengan menggabungkan dua dataran yang terpisah.
4. **STABILIZE:** Bandingkan tinggi dataran yang saat ini dipijak Raiden Shogun dan tinggi dataran **di kirinya**. Anggap dataran dengan tinggi lebih rendah memiliki tinggi **X**. Buat dataran baru dengan tinggi **X** di kanan posisi Raiden Shogun. Anda mencatat **tinggi dataran baru yang dibuat Zhongli**.

Zhongli sadar pasukannya kuat di darat, **sehingga setiap RISE dan QUAKE dijamin tidak akan membuat sebuah dataran tenggelam** (tinggi 0 atau lebih rendah). Setidak-sukanya Zhongli terhadap Raiden Shogun, dia sangat menghormati peninggalan sejarah sehingga **tidak akan ada aksi apapun jika CRUMBLE atau STABILIZE dilakukan jika Raiden Shogun sedang berada di kuil** (catat 0)

Sebagai keturunan langsung Sun Tzu, Kokomi aktif memanfaatkan kondisi bahwa Watatsumi adalah rumahnya. Sebagai Dewi laut, **dia dapat memanipulasi tinggi laut dan berjalan di air**. Jika pasukannya sedang terpojokkan, Kokomi menggunakan strategi penyerangan last resort yaitu **SWEEPING U L**. Kokomi menaikkan tinggi air laut menjadi **L** meter, menenggelamkan semua dataran dengan yang lebih rendah (dan pasukan musuh), lalu *sweeping* dari dataran terkiri hingga dataran terkanan pulau U untuk menghabiskan sisa-sisa pasukan musuh. Kokomi dapat leluasa berjalan di dataran, namun memerlukan 1 unit sihir untuk berjalan di air. Anda mencatat **berapa unit**

sihir yang diperlukan untuk setiap **SWEEPING**. Tinggi lautan kembali normal setelah suatu SWEEPING.

Selama peperangan terjadi Q aksi oleh ketiga pemimpin perang (Kokomi, Zhongli, dan Raiden Shogun). Anda akan mendokumentasikan semua kejadian ini agar peperangan ini untuk kemudian dipublikasikan. Harapannya, peperangan ini dikenang sepanjang masa!

Format Masukan

- Baris pertama berisi N , banyaknya pulau mula-mula di Watatsumi.
- N Baris berikutnya mendeskripsikan setiap pulau. Baris ke- i berisi sebuah string P_i , bilangan D_i , lalu diikuti dengan D_i buah bilangan $A_{i,j}$. Secara berturut-turut variabel tersebut menyatakan nama pulau, banyaknya dataran, dan tinggi setiap dataran secara berturut-turut di pulau ke- i .
- Baris berikutnya berisi R dan E yang secara berturut-turut merepresentasikan nama pulau dan posisi dataran (posisi kuil merupakan dataran ke-1) mula-mula Raiden Shogun berada.
- Baris pertama berisi Q , banyaknya kejadian selama peperangan ini
- Q baris berikutnya masing-masing berisi sebuah kejadian di antara 10 kemungkinan: **PISAH, UNIFIKASI, RISE, QUAKE, CRUMBLE, STABILIZE, LANGKAH, TEBAS, TELEPORTASI, SWEEPING**. Setiap kejadian diikuti dengan masukan valid sesuai yang dijelaskan pada deskripsi soal.

Format Keluaran

Keluarkan hasil catatan anda untuk setiap kejadian sesuai yang dijelaskan pada bagian deskripsi soal.

Batasan

- **UMUM**
 - $1 \leq N \leq 100.000$
 - $1 \leq Q \leq 100.000$
 - $1 \leq |P_i| \leq 10$, nama pulau unik dan hanya terdiri dari huruf alfabet kapital
 - $1 \leq D_i$ dan $\sum D_i \leq 300.000$
 - $1 \leq A_{i,j} \leq 10^9$
 - Dijamin pulau atau kuil yang disebutkan pada **UNIFIKASI, PISAH, RISE, QUAKE, TELEPORTASI**, dan **SWEEPING** ada.
 - Dijamin posisi awal Raiden Shogun valid. Pulau R ada dan memiliki setidaknya E dataran.
- **UNIFIKASI, PISAH, dan SWEEPING**
 - Sebuah pulau maksimal memiliki 20 kuil dalam suatu waktu
 - Pada pisah, dijamin U bukan merupakan kuil paling kiri pulau
 - $1 \leq L \leq 10^{12}$
- **RISE, QUAKE, STABILIZE, dan CRUMBLE**
 - $1 \leq H, X \leq 10^9$, dijamin tidak ada dataran dengan tinggi non positif setelah **RISE** atau **QUAKE**.
- **GERAK, TEBAS, dan TELEPORTASI**

- $1 \leq S \leq 100$

Contoh Masukan 1

```

3
SEIRAI 3 6 7 8
KANNAZUKA 5 10 20 10 4 10
NARUKAMI 2 7 8
NARUKAMI 2
7
CRUMBLE
CRUMBLE
UNIFIKASI SEIRAI KANNAZUKA
TELEPORTASI KANNAZUKA
QUAKE SEIRAI 8 3
TEBAS KANAN 2
TEBAS KIRI 3

```

Contoh Keluaran 1

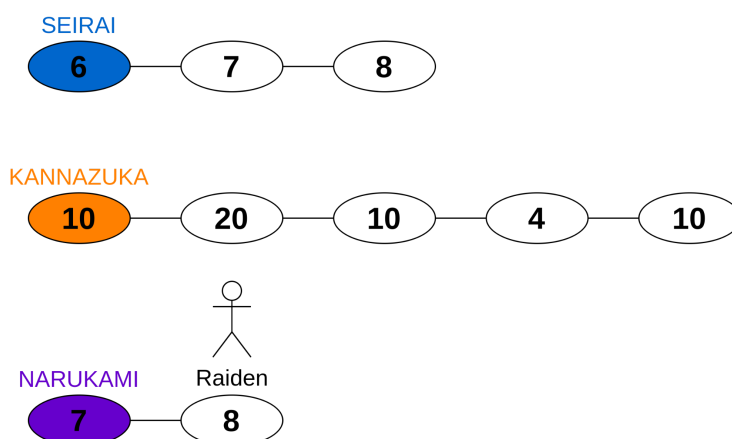
```

8
0
8
10
3
1
20

```

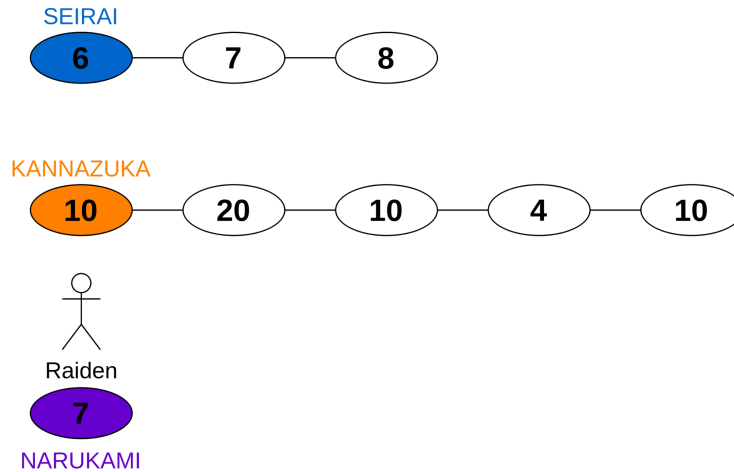
Penjelasan 1

Pada mulanya, Raiden SHogun berada di dataran setinggi 8 di pulau **NARUKAMI**. Kondisi pulau-pulau dan posisi Raiden saat ini sebagai berikut.



1. CRUMBLE

Dataran setinggi 8 yang dipijak Raiden Shogun dihancurkan. Raiden Shogun berpindah ke kirinya (dataran setinggi 7). Tercetak **8** karena itulah tinggi dataran yang dihancurkan.

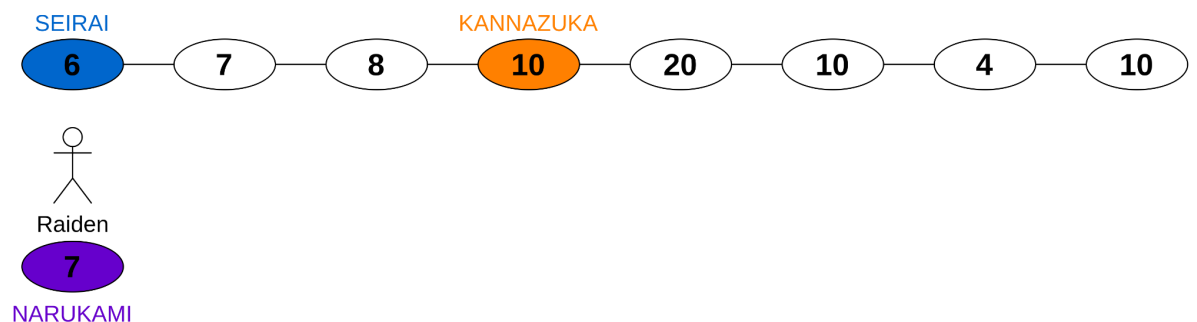


2. CRUMBLE

Dataran tidak dihancurkan karena berisi kuil. Tercetak **8** (secara khusus, lihat deskripsi soal).

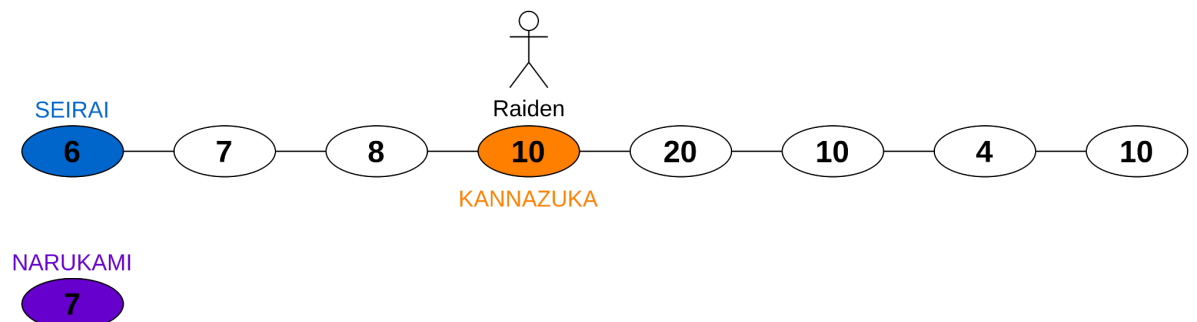
3. UNIFIKASI SEIRAI KANNAZUKA

Kini dataran di pulau **SEIRAI** = [6, 7, 8, 10, 20, 10, 4, 10]. Pulau **KANNAZUKA** dianggap tidak ada, namun **kuil KANNAZUKA** kini berada di Pulau **SEIRAI**. Tercetak **8** karena pulau baru memiliki 8 dataran. Kondisi pulau-pulau dan posisi Raiden saat ini dapat dilihat pada gambar di bawah ini.



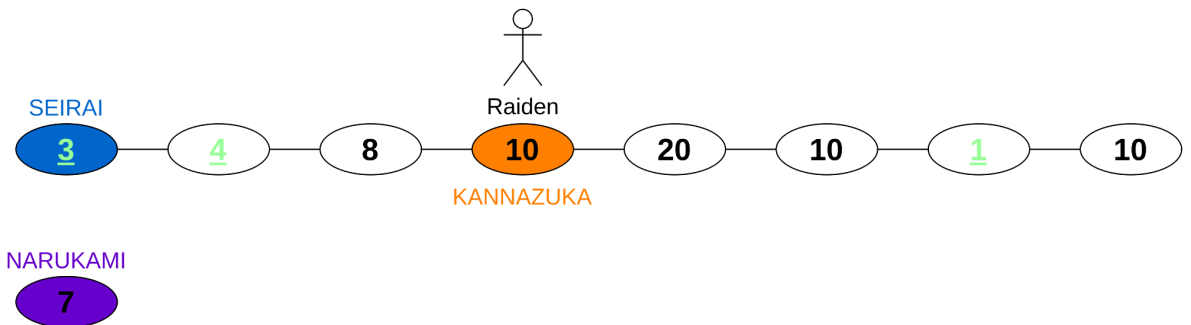
4. TELEPORTASI KANNAZUKA

Raiden teleportasi ke kuil **KANNAZUKA** (yang kini berada di pulau **SEIRAI**). Tercetak **10** karena itulah tinggi dataran tersebut saat ini. Kondisi pulau-pulau dan posisi Raiden saat ini dapat dilihat pada gambar di bawah ini.



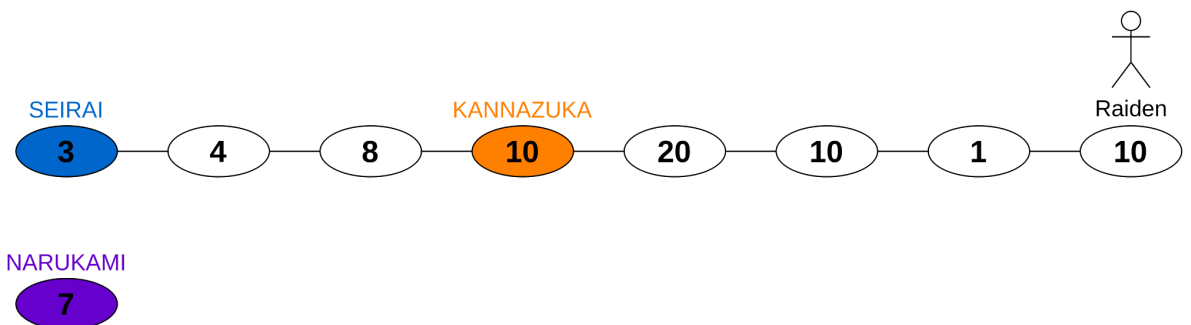
5. QUAKE SEIRAI 8 3

Semua dataran dengan tinggi di dibawah 8 diturunkan tingginya sebanyak 3. Ada **3** dataran yang memenuhi, yakni dengan tinggi 6, 7, 4. Kini dataran di pulau **SEIRAI** = [3, 4, 8, 10, 20, 10, 1, 10]. Kondisi pulau-pulau dan posisi Raiden saat ini dapat dilihat pada gambar di bawah ini.



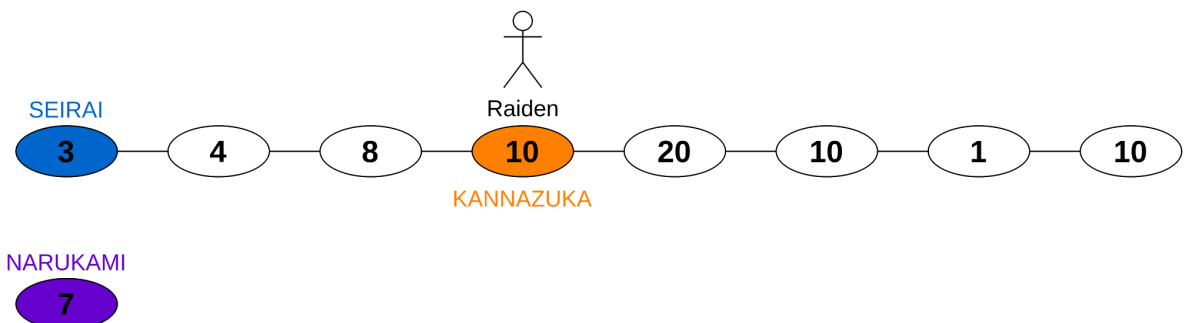
6. TEBAS KANAN 2

Saat ini Raiden Shogun berada di dataran dengan tinggi 10 dan menebas dua langkah. Ini berarti Raiden Shogun melewati dua buah dataran dengan tinggi 10, yakni dataran yang diapit tinggi (20, 1) dan dataran terkanan. Raiden Shogun berhenti di dataran terkanan. Karena **TEBAS KANAN**, maka dicetak tinggi dataran di kirinya saat berhenti, yaitu **1**.



7. TEBAS KIRI 3

Raiden Shogun mencoba menebas 3 ke KIRI. Pertama melewati dataran tinggi 10 yang diapit (20, 1), kemudian melewati dan berhenti dataran yang diapit (8, 20). Raiden Shogun baru menebas dua langkah, namun **karena tidak ada dataran dengan tinggi 10 lagi di kirinya, Raiden Shogun berhenti**. Karena **TEBAS KIRI**, Raiden Shogun mencetak dataran di kanannya, yaitu **20**.



Contoh Masukan 2

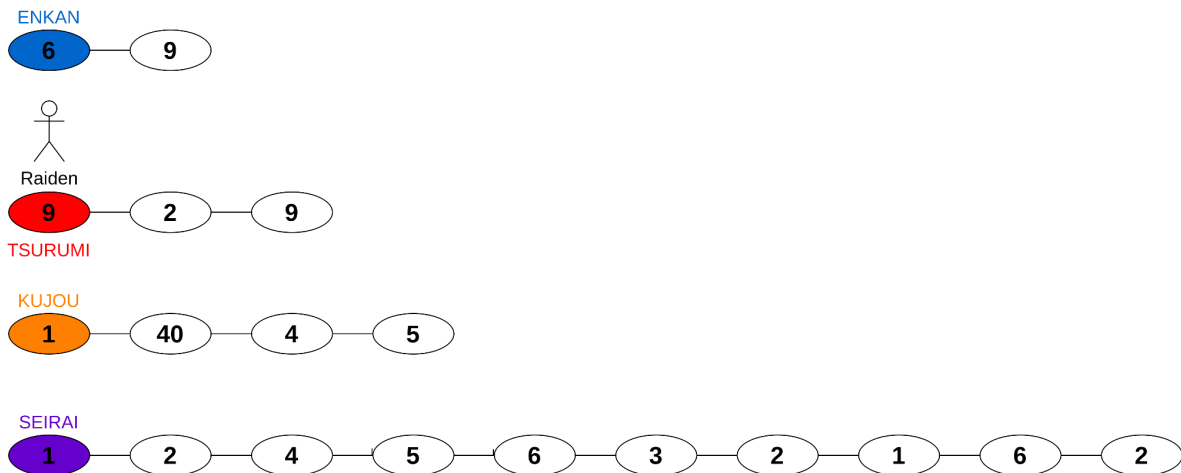
```
4
ENKAN 2 6 9
TSURUMI 3 9 2 9
KUJOU 4 1 40 4 5
SEIRAI 10 1 2 4 5 6 3 2 1 6 2
TSURUMI 1
18
UNIFIKASI ENKAN TSURUMI
UNIFIKASI ENKAN KUJOU
TEBAS KANAN 1
TEBAS KIRI 1
TELEPORTASI KUJOU
PISAH KUJOU
GERAK KIRI 1
CRUMBLE
STABILIZE
GERAK KANAN 1
CRUMBLE
GERAK KANAN 1
STABILIZE
SWEEPING ENKAN 3
RISE ENKAN 1 2
QUAKE ENKAN 7 1
SWEEPING ENKAN 3
SWEEPING SEIRAI 4
```

Contoh Keluaran 2

```
5
9
2
2
1
5 4
1
0
0
40
40
4
1
1
5
1
0
6
```

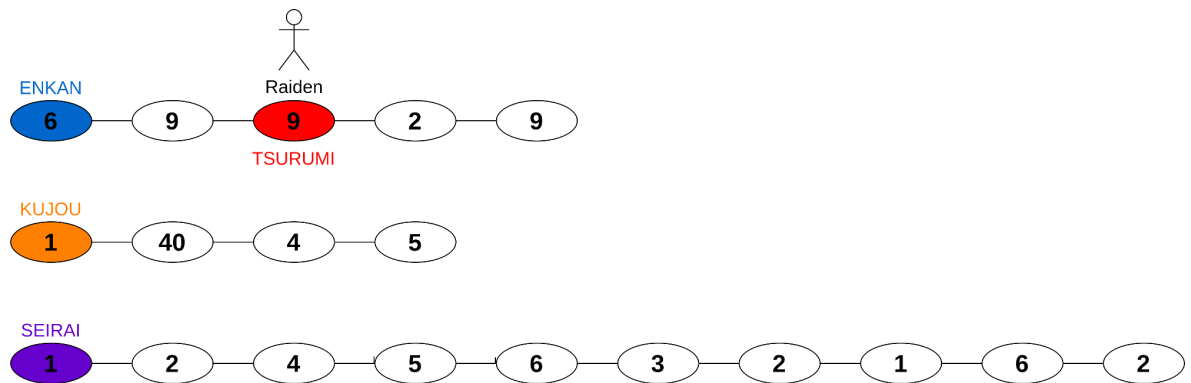
Penjelasan 2

Kita punya 3 pulau sebagai berikut



1. UNIFIKASI ENKAN TSURUMI

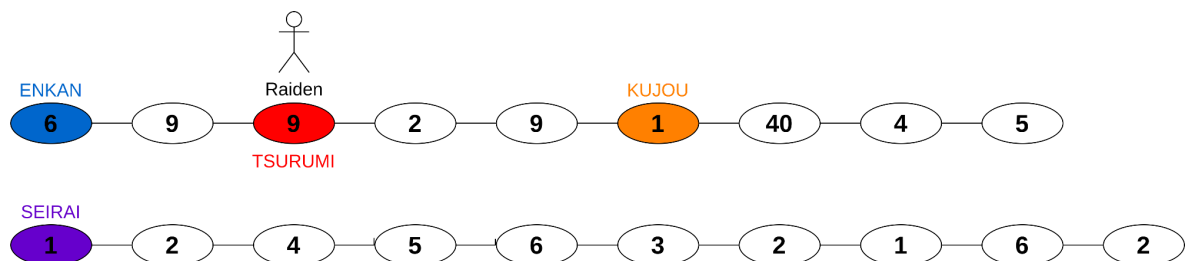
Setelah kita menjalankan operasi pertama, bentuk pulau yang terbentuk sebagai berikut.



ENKAN kini memiliki 5 dataran, sehingga tercetak **5**.

2. UNIFIKASI ENKAN KIJOU

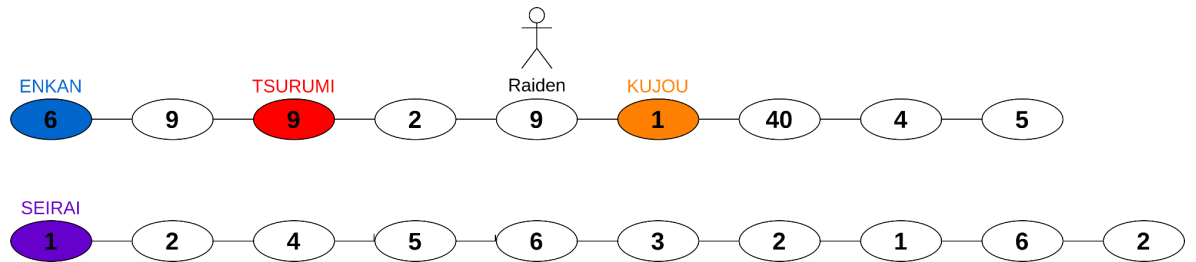
Setelah kita menjalankan operasi kedua, bentuk pulau yang terbentuk sebagai berikut



ENKAN kini memiliki 9 dataran, sehingga tercetak **9**.

3. TEBAS KANAN 1

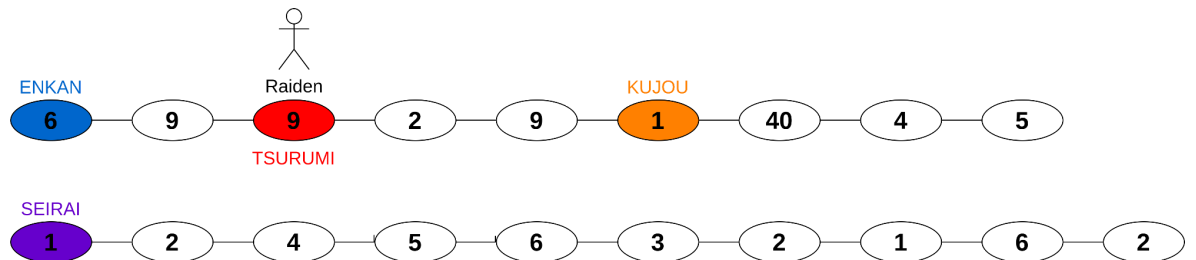
Setelah operasi **TEBAS KANAN 1**, Raiden bergerak ke dataran tinggi 9 terdekat pertama di sebelah kanan. Sehingga posisi akhir Raiden sebagai berikut.



Setelah menjalankan operasi ini, program mencetak keluaran **2** sebagai dataran yang terakhir dilewati sebelum mencapai posisi saat ini.

4. **TEBAS KIRI 1**

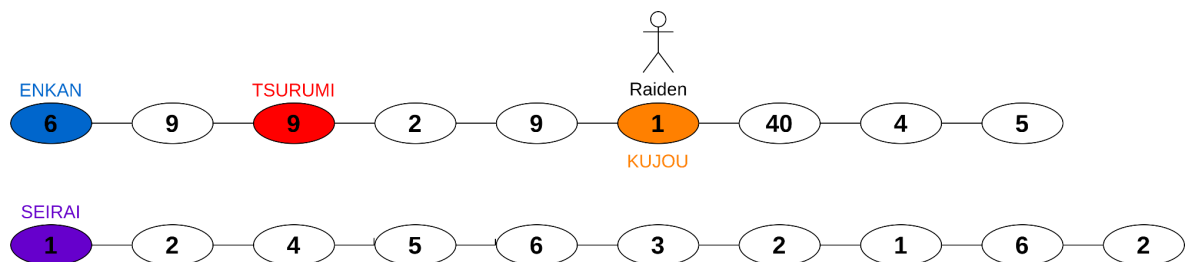
Setelah operasi **TEBAS KIRI 1**, Raiden bergerak ke dataran tinggi 9 terdekat pertama di sebelah kiri. Ini sama artinya dengan Raiden kembali ke posisi sebelumnya. Berikut ilustrasi posisi Raiden setelah melakukan **TEBAS KIRI 1**.



Setelah operasi ini dijalankan, program mencetak keluaran **2** sebagai dataran yang terakhir dilewati sebelum mencapai posisi saat ini.

5. **TELEPORTASI KUJOU**

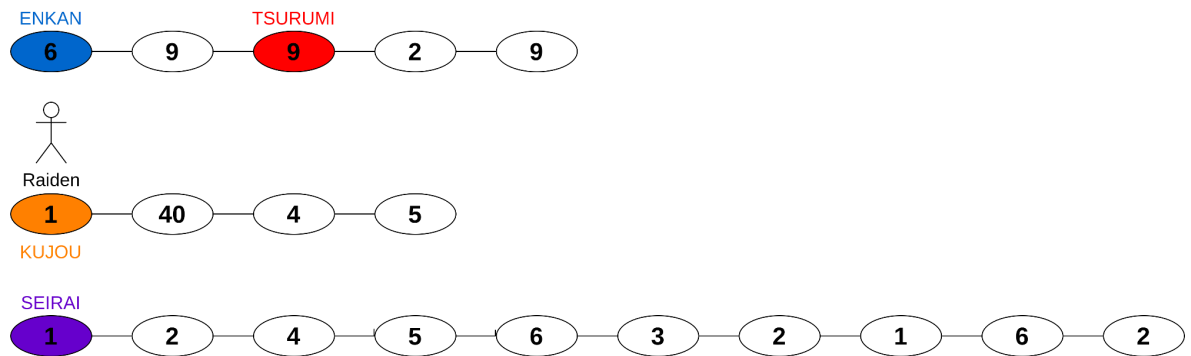
Ketika melakukan **TELEPORTASI KUJOU**, Raiden akan berpindah posisi tepat di kuil **KUJOU** berada. Berikut ilustrasi posisi Raiden saat ini.



Setelah operasi ini selesai, program mencetak keluaran **1** yang merepresentasikan ketinggian pulau saat ini.

6. **PISAH KUJOU**

Kokomi kemudian menginstruksikan untuk memecah pulau yang memiliki kuil **KUJOU** menjadi sebuah pulau independen. Berikut ilustrasi pulau dan posisi Raiden saat ini.



Kedua pulau terpisah, pulau yang tanpa kuil **KUJOU** memiliki dataran sebanyak 5 dan pulau yang memiliki kuil **KUJOU** terdapat dataran sebanyak 4. Oleh karena itu, setelah operasi ini program mencetak keluaran **5 4**.

7. GERAK KIRI 1

Karena kita tau bahwa Raiden berada di ujung pulau, Raiden Shogun tidak dapat bergerak. Meski demikian, program tetap mencetak dataran yang saat ini dipijak Raiden Shogun yaitu **1**.

8. CRUMBLE

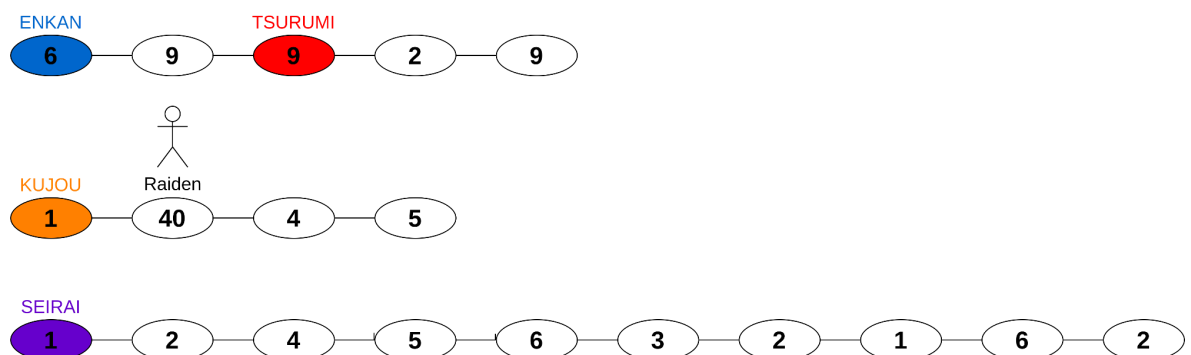
Kita tau bahwa Raiden berada di kuil **KUJOU** saat ini. Karena posisi saat ini adalah kuil, Raiden tidak dapat melakukan operasi **CRUMBLE**. Program mencetak keluaran setelah **0** operasi dilakukan.

9. STABILIZE

Sama seperti kasus sebelumnya, karena Raiden berada di kuil untuk saat ini, Raiden tidak dapat melakukan operasi **STABILIZE**. Setelah operasi ini, program mencetak keluaran **0**.

10. GERAK KANAN 1

Setelah melakukan operasi **GERAK KANAN 1**, Raiden berada di posisi sebagai berikut.

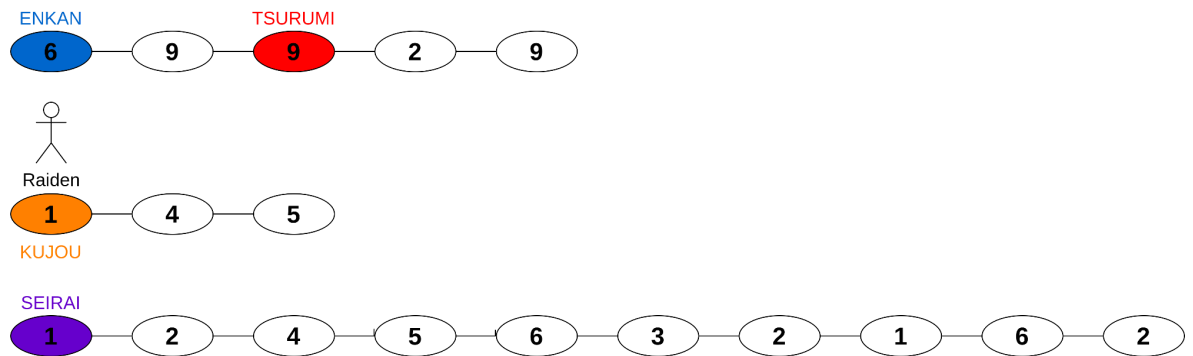


Setelah operasi dijalankan, program mencetak keluaran **40** yang merepresentasikan tinggi dataran yang dipijak Raiden saat ini.

11. CRUMBLE

Pada saat Zhongli melakukan operasi **CRUMBLE**, Zhongli menghancurkan dataran yang dipijak Raiden saat ini lalu menggabungkan kedua dataran yang terpisah. Raiden kemudian

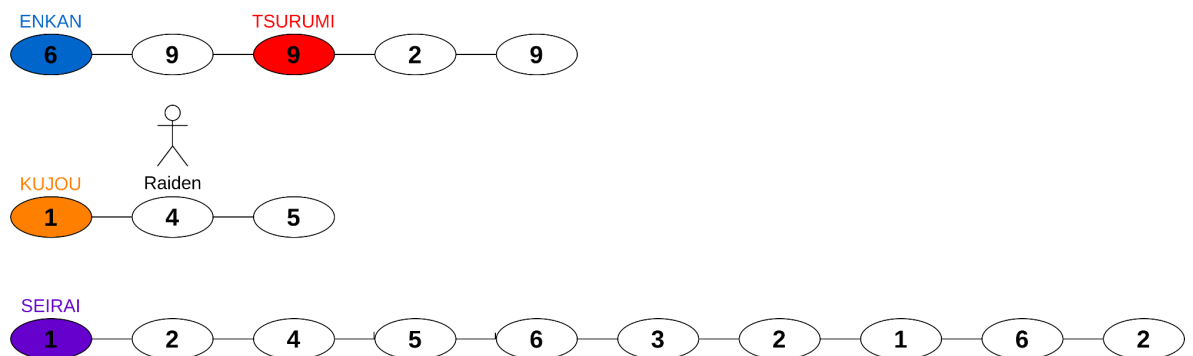
mundur berpindah posisi ke dataran sebelah kiri dataran yang sebelum dihancurkan oleh Zhongli. Berikut ilustrasi pulau yang terbentuk.



Setelah operasi ini dijalankan, program mencetak keluaran **40** yang merepresentasikan tinggi dataran yang dihancurkan oleh Zhongli.

12. GERAK KANAN 1

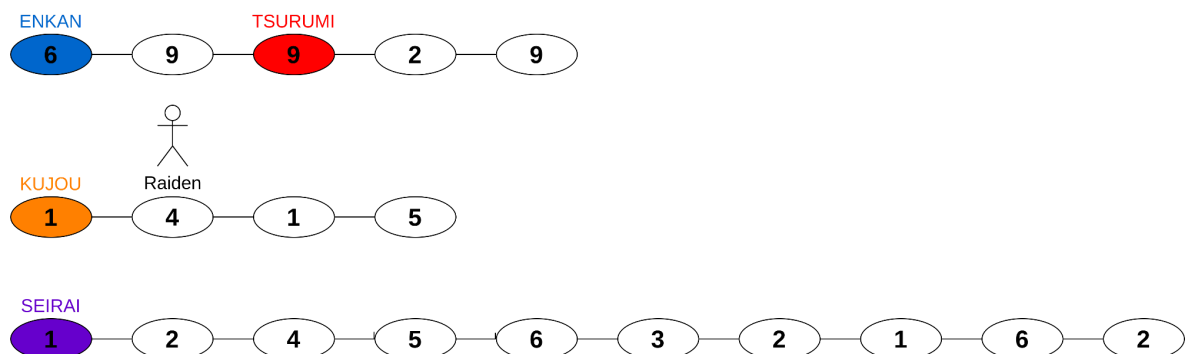
Raiden kemudian melakukan **GERAK KANAN 1** sehingga Raiden bergerak ke kanan sejauh satu langkah. Berikut ilustrasi posisi Raiden beserta pulau-pulau ada saat ini.



Setelah melakukan operasi ini, program mengeluarkan keluaran **4** yang merepresentasikan tinggi dataran yang dipijak oleh Raiden saat ini.

13. STABILIZE

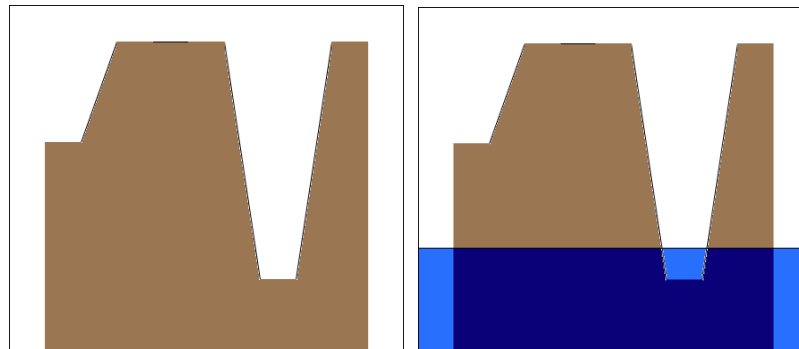
Setelah melakukan operasi **STABILIZE**, Zhongli membangun dataran setinggi 1 di sebelah kanan Raiden dikarenakan dataran sebelah kiri Raiden saat ini lebih rendah dibandingkan dengan yang Raiden pijak. Berikut ilustrasi pulau yang terbentuk.



Setelah operasi ini selesai, program mengeluarkan keluaran **1** yang menyatakan ketinggian dataran baru yang dibuat oleh Zhongli.

14. SWEEPING ENKAN 3

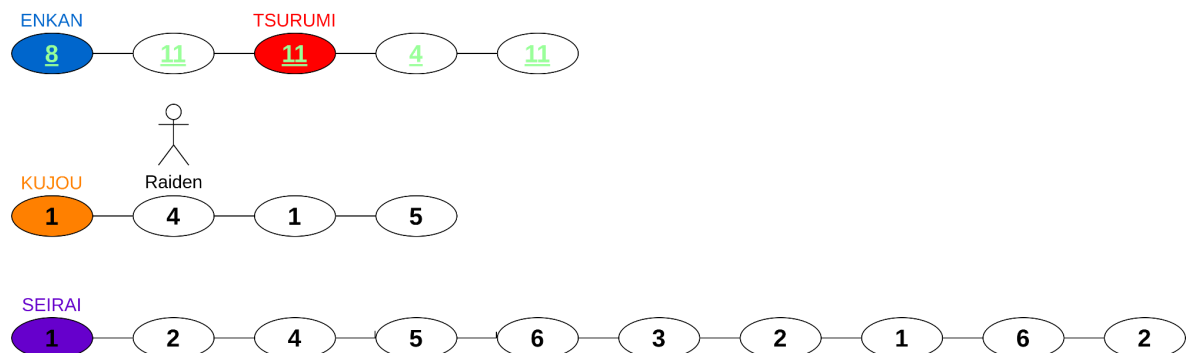
Kokomi menaikkan ketinggian air di **ENKAN** menjadi 3. Berikut visualisasi sebelum dan sesudah Kokomi meninggikan air:



Kokomi mulai sweeping dari kiri hingga kanan. Kokomi dengan mudah mendaki bukit atau menuruni lembah. Terlihat bahwa dari topologi tersebut, Kokomi hanya memerlukan **1 unit sihir**, yakni untuk menyeberang dari dataran ke-3 ke dataran ke-5

15. RISE ENKAN 1 2

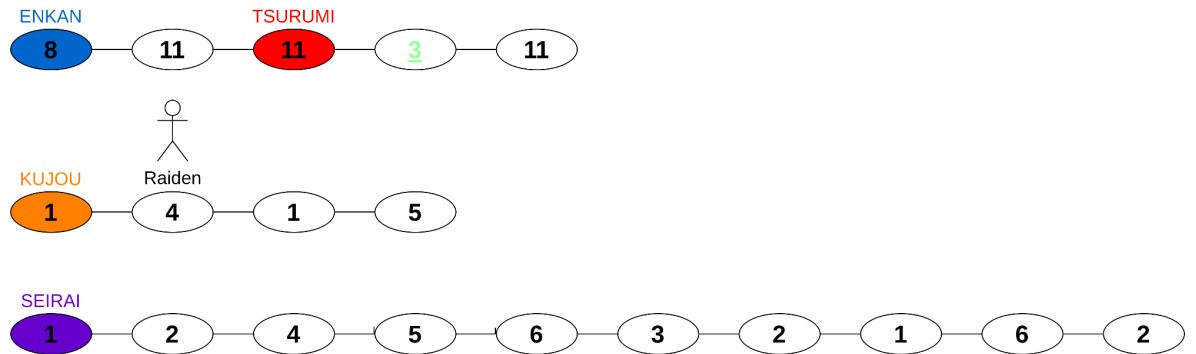
Operasi **RISE ENKAN 1 2** membuat Zhongli menaikkan semua dataran yang tingginya lebih dari 1 sebanyak dua unit pada **ENKAN**. Berikut ilustrasi yang terbentuk setelah operasi ini dijalankan.



Setelah operasi ini dieksekusi, program mengeluarkan keluaran **5** yang merepresentasikan banyaknya dataran yang terdampak setelah operasi ini dilakukan. Dalam kasus ini semua dataran di **ENKAN** memiliki tinggi lebih dari 1 sehingga kelima dataran terdampak.

16. QUAKE ENKAN 7 1

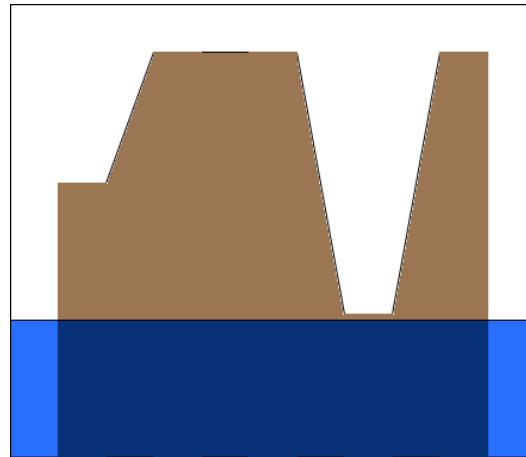
Operasi **QUAKE ENKAN 7 1** membuat Zhongli menurunkan semua dataran yang tingginya kurang dari 7 sebanyak satu unit. Berikut ilustrasi yang terbentuk setelah operasi ini dijalankan.



Setelah operasi ini dieksekusi, program mengeluarkan keluaran **1** yang merepresentasikan banyaknya dataran yang terdampak setelah operasi ini dilakukan. Dalam kasus ini hanya ada sebuah dataran di ENKAN memiliki tinggi kurang dari 7.

17. SWEEPING ENKAN 3

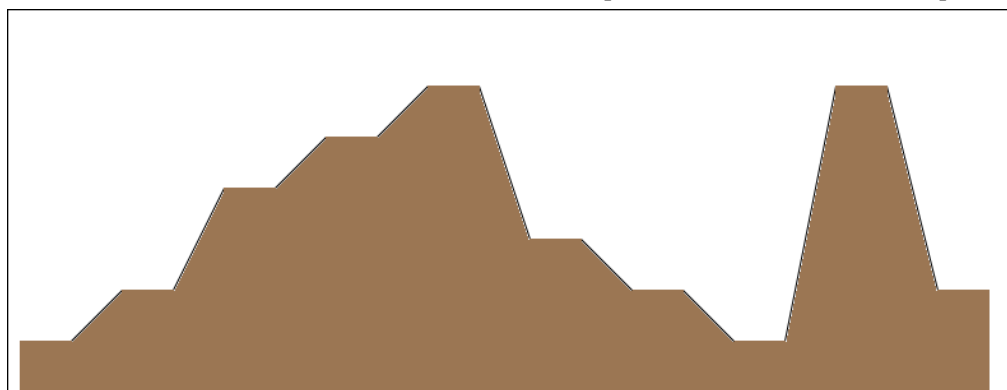
Kokomi menaikkan ketinggian air di **ENKAN** menjadi 3. Berikut visualisasi setelah air dinaikkan:



Perhatikan bahwa **dataran dengan tinggi 3 tidak tenggelam**. Kokomi dapat sweeping pulau ini **tanpa menggunakan sihir** karena tidak ada keperluan untuk berjalan di air.

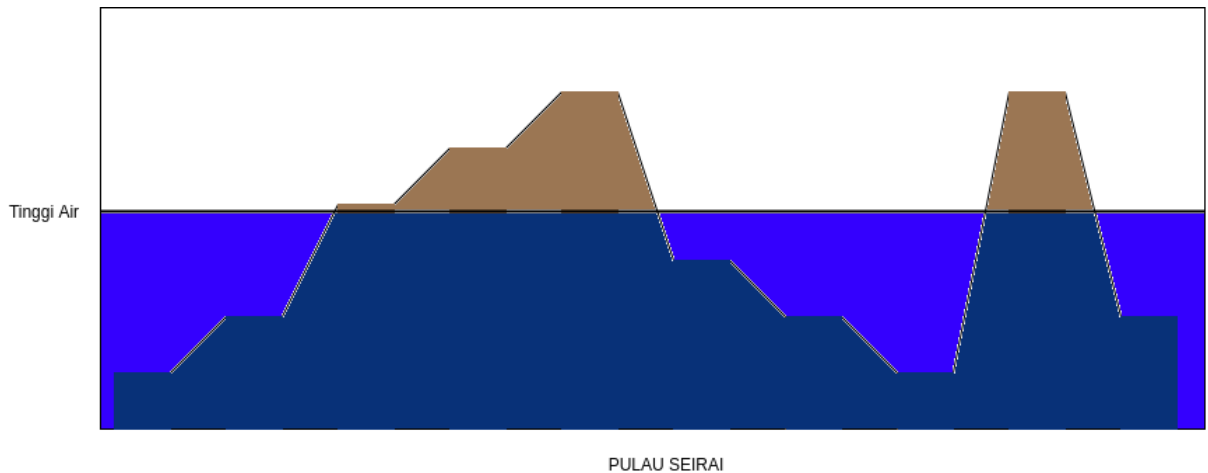
18. SWEEPING SEIRAI 4

Berikut kondisi awal Pulau **SEIRAI** dengan dataran = [1, 2, 4, 5, 6, 3, 2, 1, 6, 2]



PULAU SEIRAI

Berikut kondisi Pulau setelah tinggi air dinaikkan. Perhatikan bahwa dataran dengan tinggi 4 tidak tergenang:



Kokomi selalu mulai dari dataran terkiri, meski pada kasus ini terendam air. Berikut deskripsi yang dilakukan Kokomi:

- Dari dataran terkiri (yang terendam air), kokomi memerlukan **dua unit sihir** untuk pergi ke dataran tidak terendam berikutnya.
- Setelah sweeping dataran tersebut, Kokomi memerlukan **tiga unit sihir** untuk pergi ke dataran tidak terendam berikutnya
- Terakhir, Kokomi memerlukan **satu unit sihir** untuk pergi ke kanan pulau. **Ini tetap harus dilakukan meskipun dataran terendam air.**

Total, Kokomi memerlukan **6 unit sihir** untuk menyelesaikan sweeping ini. Perhatikan jika semua dataran tergenang, **Kokomi tetap harus pergi dari dataran terkiri hingga terkanan**. Salah satu alasannya untuk membersihkan pasukan-pasukan musuh yang tenggelam.

Keterangan Tambahan

Scope TP ini adalah LinkedList, Sorting, dan BST. **Tidak diperbolehkan menggunakan struktur data tingkat lanjut yang belum atau tidak diajarkan di SDA.** Termasuk, tetapi tidak terbatas pada: Hash Table implementasi sendiri, Minimum Spanning Tree, Segment Tree, Fenwick Tree dan SQRT Decomposition.

Jika solusi anda menggunakan LinkedList, **wajib** untuk mengimplementasikan LinkedList tersebut sendiri. Tidak diperbolehkan menggunakan library LinkedList bawaan Java seperti `LinkedList<>()`, `LinkedListDeque<>()`, atau `LinkedHashMap<>()`.

Jika solusi anda menggunakan sorting, **wajib** untuk mengimplementasikan semua proses sorting anda **tanpa menggunakan bantuan library sorting java**, seperti `Collections.sort()`.

Tidak diperlukan AVL tree atau balancing algorithm BST lain untuk menyelesaikan TP ini (diluar scope). Meski demikian, anda **diperbolehkan** memakai AVL Tree atau *balancing algorithm* implementasi sendiri dalam implementasi anda.

Informasi Tambahan Test-case

Kejadian yang mungkin	Nomor TC
UNIFIKASI, PISAH	1-15
RISE, QUAKE	16-25
CRUMBLE, STABILIZE, GERAK, TEBAS, TELEPORTASI	26-41
UNIFIKASI, PISAH, RISE, QUAKE	42-46
GERAK, TEBAS, TELEPORTASI	47-53
GERAK, TEBAS, TELEPORTASI, RISE, QUAKE	54-55
SWEEPING	56-61
SWEEPING, UNIFIKASI, PISAH, RISE, QUAKE	62-67
GERAK, TEBAS, TELEPORTASI, UNIFIKASI, PISAH	68-77
GERAK, TEBAS, TELEPORTASI, UNIFIKASI, PISAH, CRUMBLE, STABILIZE	78-79
All Query	80-100