



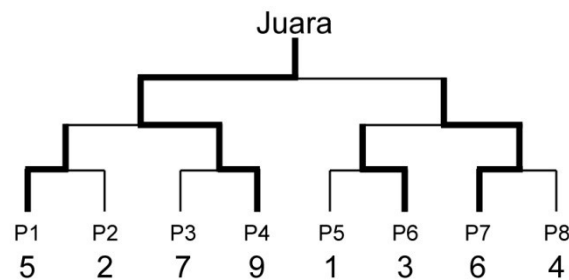
Problem A

Turnamen Panco

Author

Suhendry Effendy

Pak Kumis menyelenggarakan sebuah turnamen panco dengan jumlah peserta 2^N . Peserta pertama (P1) akan bertanding dengan peserta kedua (P2). P3 akan bertanding dengan P4, P5 akan bertanding dengan P6, dst. Pemenang dari P1 dan P2 akan bertanding dengan pemenang dari P3 dan P4, pemenang dari P5 dan P6 akan bertanding dengan pemenang dari P7 dan P8, dst (lihat bagan di bawah).



Pak Kumis sudah mengetahui kekuatan setiap peserta yang mengikuti turnamen ini dan ia yakin tidak ada dua peserta yang memiliki kekuatan yang sama. Jika ada dua orang peserta yang bertanding, maka yang kuat lah yang menang. Bantu pak Kumis untuk memprediksi siapa yang akan memenangkan turnamen ini. Pada contoh di atas, turnamen ini dimenangkan oleh peserta ke 4 (P4) yang memiliki kekuatan 9 (Ia mengalahkan P3 yang mempunyai kekuatan 7 pada babak pertama, mengalahkan P1 yang memiliki kekuatan 5 pada babak kedua, dan mengalahkan P7 yang memiliki kekuatan 6 pada babak final).

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi sebuah bilangan bulat N ($1 \leq N \leq 10$). Baris berikutnya berisi 2^N bilangan bulat P_i ($1 \leq P_i \leq 5.000$) yang merepresentasikan kekuatan dari peserta ke 1 hingga peserta ke 2^N secara berurutan. Tidak ada dua peserta yang memiliki kekuatan yang sama.

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan nomor peserta yang diprediksi akan memenangkan turnamen ini.

Contoh Input	Output Untuk Contoh Input
1 3 5 2 7 9 1 3 6 4	4



Problem B

Mencari Donat

Author

Ricky Winata

Pak Buncit ingin membuka toko kue yang menjual donat. Karena ia memiliki ambisi untuk menjual donat dalam jumlah besar, maka proses pembuatan donat diserahkan kepada mesin. Untuk menjaga bentuk donat yang dihasilkan tetap seperti donat, maka pak Buncit memasang alat pemindai (*scanner*) di tahap akhir proses pembuatan donat. Alat pemindai ini mampu memindai donat-donat yang dihasilkan oleh mesin tersebut. Sayangnya, pak Buncit tidak memiliki program yang mampu mengolah informasi yang dihasilkan dari alat pemindai itu. Pak Buncit ingin mengetahui dari setiap gambar hasil pemindaian, ada berapa donat yang terdapat dalam gambar tersebut.

Untuk mendapatkan informasi tersebut, pak Buncit menggunakan cara ini: Gambar hasil pemindaian terdiri dari karakter '1' dan '0'. Sekumpulan karakter '1' yang saling terhubung secara horisontal, vertikal maupun diagonal disebut sebagai komponen (bagian roti dari donat). Sedangkan kumpulan karakter '0' yang dikelilingi oleh sebuah komponen dan tidak mengelilingi komponen apapun adalah lubang (bagian lubang dari donat). Sebuah donat adalah sebuah komponen yang mengelilingi tepat satu lubang.

```
11111
10001
10001
10001
11111
```

Satu donat, satu komponen dengan satu lubang

```
11111
10001
10101
10001
11110
```

Bukan donat, dua komponen dan tidak ada lubang.

```
00111
00101
11111
10100
11100
```

Bukan donat, satu komponen dengan dua lubang.

```
001111
000101
010011
101000
111000
```

Dua buah donat.

Tugas anda adalah untuk membuat program untuk menghitung jumlah donat yang ada pada setiap gambar hasil pemindaian.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus. Setiap kasus dimulai dengan dua buah bilangan bulat N dan M ($1 \leq N, M \leq 100$) yang menyatakan jumlah baris dan kolom secara berurutan. N baris berikutnya masing-masing berisi M karakter ('0' atau '1') yang merepresentasikan gambar hasil pemindaian

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan jumlah donat yang ada pada gambar tersebut.



BNPC

The 2010
Bina Nusantara Programming Contest
for High School Students



Contoh Input	Output Untuk Contoh Input
5	1
3 5	0
11111	0
11001	2
11111	2
5 5	
11111	
10001	
10101	
10001	
11110	
5 5	
00111	
00101	
11111	
10100	
11100	
3 7	
0100010	
1010101	
0100110	
5 6	
001111	
000101	
010011	
101000	
111000	



Problem C

Tetangga Bidang Persegi

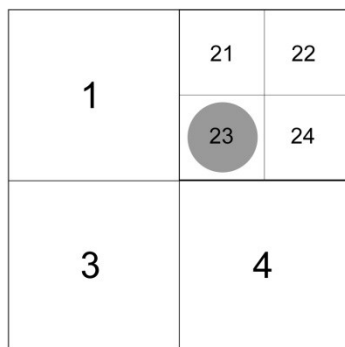
Author

Panji Kharisma

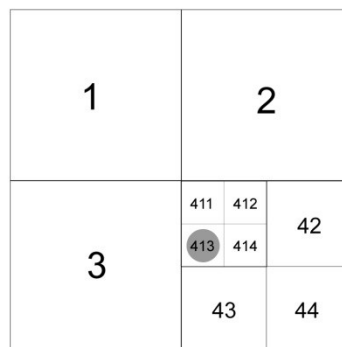
Sebuah persegi bisa dibagi menjadi bidang-bidang dengan ketentuan sebagai berikut:

1. Bidang kiri atas diberi nomor 1.
2. Bidang kanan atas diberi nomor 2.
3. Bidang kiri bawah diberi nomor 3.
4. Bidang kanan bawah diberi nomor 4.
5. Sistem penomoran ini berlaku rekursif.

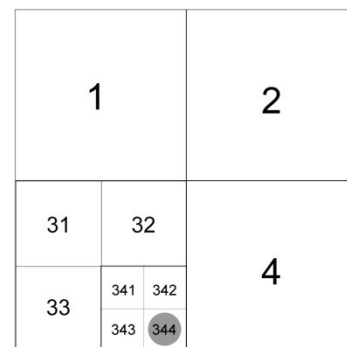
Berikut ini beberapa contoh bidang dan lokasinya pada persegi.



Lokasi bidang "23"



Lokasi bidang "413"



Lokasi bidang "344"

Bidang "23" di atas memiliki 4 tetangga, yaitu bidang 1 (di kiri), 21 (di atas), 24 (di kanan) dan 4 (di bawah). Bidang "413" di atas juga memiliki 4 tetangga, yaitu bidang 3 (di kiri), 411 (di atas), 414 (di kanan) dan 43 (di bawah). Sedangkan bidang "344" memiliki 3 tetangga, yaitu bidang 343 (di kiri), 342 (di atas) dan 4 (di kanan), tidak ada tetangga yang berada di bawah.

Diberikan sebuah bidang, tentukan siapa saja tetangga dari bidang tersebut.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi sebuah string S yang merupakan nomor bidang yang diberikan. S panjangnya antara 1 dan 100, inklusif, dan hanya terdiri dari karakter 1..4.

Format Output

Untuk setiap kasus, output "Case #X:" (tanpa tanda kutip) di mana X adalah nomor kasus, berurutan dimulai dari nomor 1. Kemudian output tetangga dari bidang S dengan satu bidang pada satu baris diurutkan secara leksikograf (urutan kamus).

Contoh Input	Output Untuk Contoh Input
4 23 413 344 211	Case #1: 1 21 24 4 Case #2: 3 411 414 43 Case #3: 342 343 4 Case #4: 1 212 213



Problem D

Mari Berhitung

Author

Harryanto

Diberikan empat buah bilangan bulat A , B , N dan M , hitunglah jumlah dari semua bilangan yang berada di antara A dan B , inklusif, yang habis dibagi oleh N namun tidak habis dibagi oleh M .

Contoh. $A = 4$, $B = 20$, $N = 3$, $M = 5$. Bilangan dari 4 hingga 20 yang habis dibagi 3 namun tidak habis dibagi 5 adalah: 6, 9, 12 dan 18 (15 tidak diikutkan karena 15 habis dibagi 5). Sehingga jumlahnya adalah $6 + 9 + 12 + 18 = 45$.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 500$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi empat buah bilangan bulat A , B , N dan M ($1 \leq A, N, M \leq B \leq 20.000.000$).

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan jumlah bilangan bulat yang berada di antara A dan B , inklusif, yang habis dibagi oleh N namun tidak habis dibagi oleh M .

Hint: gunakan tipe data integer 64-bit untuk outputnya.

Contoh Input	Output Untuk Contoh Input
3	45
4 20 3 5	60
2 20 4 7	18
1 10 3 8	

Problem E

Reduksi String

Author

Winardi Kurniawan

Diberikan sebuah string yang terdiri dari tiga karakter ('a', 'b', dan 'c'), anda diminta untuk mereduksi string tersebut dengan aturan:

- aaa, aba dan aca bisa direduksi menjadi a.
- bab, bbb dan bcb bisa direduksi menjadi b.
- cac, cbc dan ccc bisa direduksi menjadi c.

Contoh.

S = abcbabcba

abcbabcba bisa direduksi menjadi abcbcbba

abcbcbba bisa direduksi menjadi abcbba

abcbba bisa direduksi menjadi abba

abba bisa direduksi menjadi a

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi sebuah string S dengan panjang antara 1 hingga 10.000. S hanya tersusun atas karakter 'a', 'b', atau 'c'.

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan panjang minimum string setelah direduksi.

Contoh Input	Output Untuk Contoh Input
7	1
a	2
ab	1
aba	2
abab	2
aaaaaaaaa	1
abcbabcba	1
cac	

Problem F
Susun Balok

Author

Eko Wibowo

Pak Kumis sedang bermain-main dengan N buah balok yang masing-masing memiliki ketinggian H_i . Beliau ingin menumpuk beberapa balok sedemikian sehingga tinggi tumpukan balok tersebut tepat M . Pak Kumis kebingungan apakah ia bisa memilih menumpuk beberapa balok sedemikian sehingga tingginya tepat M , disinilah tugas anda untuk membantu pak Kumis.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 20$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi dua buah bilangan bulat N ($1 \leq N \leq 250.000$) dan M ($1 \leq M \leq 1.000$) yang menyatakan jumlah balok yang tersedia dan tinggi tumpukan balok yang diinginkan secara berurutan. Baris berikutnya terdiri dari N buah bilangan bulat positif H_i ($1 \leq H_i \leq 1.000.000$) yang menyatakan tinggi masing-masing balok.

Format Output

Untuk setiap kasus, output dalam sebaris kata "YA" (tanpa kutip) jika pak Kumis bisa menumpuk beberapa balok sehingga tingginya tepat M , atau "TIDAK" (tanpa kutip) jika tidak bisa.

Contoh Input	Output Untuk Contoh Input
2	YA
4 6	TIDAK
1 2 2 3	
4 10	
3 3 3 3	

Penjelasan contoh input pada kasus 1.

Pak Kumis bisa memilih balok pertama, kedua dan keempat ($1 + 2 + 3$) sehingga total tingginya adalah 6.

Problem G

Panda Sehat

Author

Indra

Panda Pak Buncit sehat-sehat! Rahasiannya adalah mereka sering berolahraga (ya! panda Pak Buncit tidak malas) dan diberi makan makanan yang bergizi. Setiap hari panda Pak Buncit diberi makanan yang mengandung N jenis nutrisi yang masing-masing sejumlah W_i miligram. Nutrisi-nutrisi ini bisa didapatkan dari dua jenis makanan. Satu paket makanan A mengandung T_i miligram masing-masing nutrisi yang diperlukan, sedangkan satu paket makanan B mengandung S_i miligram masing-masing nutrisi yang diperlukan.

Pak Buncit perlu membeli satu atau beberapa paket makanan guna memenuhi kebutuhan minimal panda-pandanya setiap hari. Paket yang dijual bisa dibeli sebagian. Jika Pak Buncit hanya membutuhkan $1/x$ isi paket (tentunya dengan komposisi nutrisi proporsional), maka ia hanya perlu membayar $1/x$ dari harga satu paket utuh. Bantu Pak Buncit untuk menghitung biaya minimal yang ia perlukan.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 150$) yang menyatakan banyaknya kasus. Setiap kasus dimulai dengan sebuah bilangan bulat N ($1 \leq N \leq 300$). Baris kedua berisi N buah bilangan bulat W_i ($1 \leq W_i \leq 200.000$) yang secara berurutan menyatakan jumlah nutrisi ke- i yang diperlukan panda. Baris ketiga berisi N buah bilangan bulat T_i ($1 \leq T_i \leq 2.000$) yang secara berurutan menyatakan jumlah miligram nutrisi ke- i yang terdapat pada satu paket makanan A. Baris keempat berisi N buah bilangan bulat S_i ($1 \leq S_i \leq 2.000$) yang secara berurutan menyatakan jumlah miligram nutrisi ke- i yang terdapat pada satu paket makanan B. Baris kelima berisi dua buah bilangan bulat C_A dan C_B yang menyatakan harga satu paket makanan A dan B secara berurutan.

Format Output

Untuk setiap kasus, output dalam sebaris biaya minimum yang perlu dikeluarkan pak Buncit untuk membeli makanan yang memenuhi kebutuhan nutrisi pandanya. Output dengan dua angka di belakang koma.

Contoh Input	Output Untuk Contoh Input
2	500.00
2	1466.67
20 16	
2 1	
4 8	
1000 100	
3	
20 6 10	
5 1 1	
2 1 5	
300 200	



BNPC

The 2010
Bina Nusantara Programming Contest
for High School Students



Penjelasan contoh input pada kasus 1.

Pak Buncit bisa membeli 5 paket B sehingga nutrisi-1 yang didapatkan adalah $5 * 4 = 20$ dan nutrisi-2 adalah $5 * 8 = 40$. Harga yang harus dibayar adalah $5 * 100 = 500.00$.

Penjelasan contoh input pada kasus 2.

Pak Buncit bisa membeli $8/3$ paket A dan $10/3$ paket B sehingga total nutrisi yang ia dapatkan untuk nutrisi-1 adalah $8/3 * 5 + 10/3 * 2 = 20$, nutrisi-2 adalah $8/3 * 2 + 10/3 * 1 = 8.67$ dan nutrisi-3 adalah $8/3 * 1 + 10/3 * 5 = 19.33$, mencukupi kebutuhan panda (20 untuk nutrisi-1, 6 untuk nutrisi-2 dan 10 untuk nutrisi-3). Harga yang harus dibayar adalah $8/3 * 300 + 10/3 * 200 = 1466.67$.

Problem H

Basis Bilangan Fibonacci

Author

Felix Jingga

Deret Fibonacci didefinisikan sebagai berikut:

$$F_n = \begin{cases} n, & n < 2 \\ F_{n-1} + F_{n-2}, & n \geq 2 \end{cases}$$

Beberapa deret Fibonacci pertama ($n = 1..7$) adalah 1, 1, 2, 3, 5, 8, 13.

Setiap bilangan bulat positif dapat direpresentasikan sebagai bilangan berbasis-fibonacci. Dalam bilangan basis-fibonacci, bit ke- i dari kanan jika bernilai 1 artinya bilangan tersebut ditambah dengan F_i (fibonacci ke- i). Contoh:

$$\begin{aligned} 17 &= 1 && 1 && 1 && 0 && 1 && 0 \\ &= 1 * F_6 &+& 1 * F_5 &+& 1 * F_4 &+& 0 * F_3 &+& 1 * F_2 &+& 0 * F_1 \\ &= 8 &+& 5 &+& 3 &+& 0 &+& 1 &+& 0 \\ &= 17 \end{aligned}$$

Jadi 17 dapat direpresentasikan sebagai 111010 dalam basis-fibonacci. Namun sayangnya, ternyata representasi dari suatu bilangan terhadap basis-fibonacci tidaklah unik. Contohnya bilangan 17 di atas bisa direpresentasikan dalam bentuk lain:

$$\begin{aligned} 17 &= 111010 = F_6 + F_5 + F_4 + F_2 \\ &= 111001 = F_6 + F_5 + F_4 + F_1 \\ &= 110111 = F_6 + F_5 + F_3 + F_2 + F_1 \\ &= 1000111 = F_7 + F_3 + F_2 + F_1 \\ &= 1001010 = F_7 + F_4 + F_2 \\ &= 1001001 = F_7 + F_4 + F_1 \end{aligned}$$

Ada 6 cara untuk merepresentasikan 17 dalam bilangan basis-fibonacci. Diberikan sebuah bilangan bulat positif N , tentukan ada berapa cara untuk merepresentasikan N dalam bilangan basis-fibonacci.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 200$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi sebuah bilangan bulat N ($1 \leq N \leq 2.000.000$).

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan banyaknya cara merepresentasikan N ke dalam basis-fibonacci.

Contoh Input	Output Untuk Contoh Input
3 1 5 17	2 3 6



Problem I

Misteri Rekursi Fibonacci

Author

Welly Dwi Putra

Fibonacci adalah deret bilangan yang sering digunakan ketika kita belajar membuat fungsi rekursi. Deret Fibonacci ke N yaitu F_N didefinisikan sebagai penjumlahan dari dua deret sebelumnya (F_{N-1} dan F_{N-2}) dengan F_0 dan F_1 sebagai basis.

Berikut adalah fungsi rekursi yang digunakan untuk menghitung bilangan fibonacci ke N dalam C/C++ maupun Pascal.

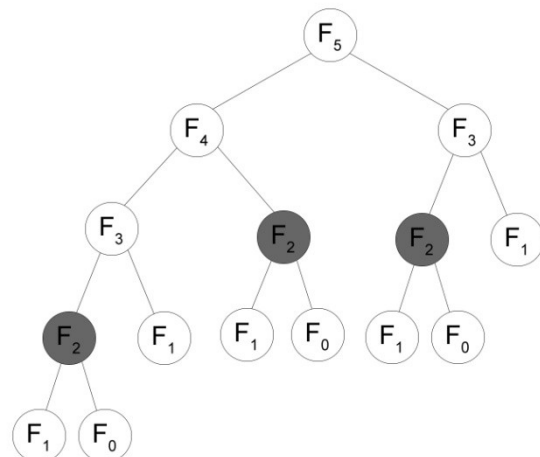
C/C++

```
int f(int n) {
    if ( n <= 1 ) return n;
    return f(n-1) + f(n-2);
}
```

Pascal

```
function f(n: integer) : integer;
begin
    if ( n <= 1 ) then f := n
    else f := f(n-1) + f(n-2);
end;
```

Di samping adalah gambar pohon pemanggilan fungsi rekursi ketika menghitung F_5 . Tampak bahwa untuk menghitung F_5 , F_2 akan dipanggil sebanyak tiga kali.



Diberikan N dan M , tugas anda adalah menghitung berapa kali F_M akan dipanggil untuk menghitung F_N . Karena angka ini bisa besar, modulo hasilnya dengan 10.000.

Format Input

Input dimulai dengan sebuah bilangan bulat T ($T \leq 1.000$) yang menyatakan banyaknya kasus. Setiap kasus terdiri dari sebaris yang berisi dua buah bilangan bulat N dan M ($0 \leq N, M \leq 10.000$).

Format Output

Untuk setiap kasus, output dalam sebaris sebuah bilangan bulat yang menyatakan banyaknya F_M dipanggil ketika menghitung F_N . Hasilnya dimodulo dengan 10.000.

Contoh Input	Output Untuk Contoh Input
3	3
5 2	2
3 1	3
6 3	



Problem J

Ikuti Iramanya!

Author

Suhendry Effendy

Ricat sedang merancang sebuah permainan komputer yang memanfaatkan ritme musik dan pergerakan *mouse*.

Sejumlah persegi akan muncul di layar secara berkala, pemain harus mengklik persegi yang muncul tersebut untuk mendapatkan poin. Setiap persegi berpusat di koordinat (X_i, Y_i) dan merentang sejauh S ke empat arah (atas, bawah, kiri kanan). Contoh, koordinat di $(20, 50)$ dengan $S = 10$ akan membentuk sebuah persegi dengan keempat koordinat sudutnya: $(10,40)$, $(10,60)$, $(30,40)$, dan $(30,60)$.

Poin yang diperoleh pemain dihitung dengan cara:

- Jika pemain melakukan klik di area persegi tepat pada saat persegi tersebut muncul, maka ia mendapatkan 300 poin.
- Jika pemain melakukan klik di area persegi 1 detik setelah persegi tersebut muncul, maka ia mendapatkan 100 poin.
- Jika pemain melakukan klik di area persegi 2 detik setelah persegi tersebut muncul, maka ia mendapatkan 50 poin.
- Jika persegi yang muncul tidak juga diklik setelah 2 detik berlalu, maka persegi tersebut akan hilang.
- Setiap persegi hanya bisa diklik satu kali (persegi tersebut akan hilang setelah diklik).
- Semua klik yang tidak berada di area persegi akan diabaikan.

Bantu Ricat membuat program untuk menghitung total poin yang didapatkan oleh pemain.

Format Input

Baris pertama berisi sebuah bilangan bulat T ($T \leq 100$) yang menyatakan jumlah kasus.

Setiap kasus dimulai dengan tiga buah bilangan bulat N , M dan S ($1 \leq N, M \leq 100$; $1 \leq S \leq 30$) yang menyatakan banyaknya persegi yang akan muncul, banyaknya klik yang dilakukan pemain, dan panjang rentang setiap persegi dari koordinat pusatnya.

N baris berikutnya masing-masing berisi tiga buah bilangan bulat T_i , X_i dan Y_i ($1 \leq T_i \leq 1.000$; $50 \leq X_i, Y_i \leq 950$) yang menyatakan waktu kemunculan persegi ke- i dan koordinat (x,y) dari pusat persegi ke- i secara berurutan. Data persegi disajikan terurut menaik berdasarkan T .

M baris berikutnya masing-masing berisi tiga buah bilangan bulat P_i , R_i dan C_i ($1 \leq P_i \leq 1.000$; $1 \leq R_i, C_i \leq 1000$) yang menyatakan waktu kapan pemain melakukan klik dan koordinat (x, y) klik tersebut secara berurutan. Data klik disajikan terurut menaik berdasarkan P .

Anda boleh mengasumsikan bahwa tidak ada koordinat pada layar yang memuat lebih dari satu persegi pada waktu yang sama.



BNPC

The 2010
Bina Nusantara Programming Contest
for High School Students



Format Output

Untuk setiap kasus, output dalam satu baris sebuah bilangan bulat yang menyatakan total poin yang didapatkan oleh pemain.

Contoh Input	Output Untuk Contoh Input
3	400
2 3 15	150
5 100 200	100
6 200 70	
5 110 190	
6 350 120	
7 190 80	
3 5 20	
10 500 500	
20 500 500	
30 500 500	
11 490 485	
12 500 500	
22 505 503	
29 500 500	
30 400 400	
1 2 10	
100 200 200	
100 220 210	
101 210 190	

Penjelasan contoh input pada kasus 1

- Klik pertama mengenai persegi-1 dan mendapatkan 300 poin
- Klik kedua tidak mendapatkan poin.
- Klik ketiga mengenai persegi-2 dan mendapatkan 100 poin.

Penjelasan contoh input pada kasus 2

- Klik pertama mengenai persegi-1 dan mendapatkan 100 poin
- Klik kedua tidak mendapatkan poin (persegi-1 sudah diklik dan hilang).
- Klik ketiga mengenai persegi-2 dan mendapatkan 50 poin.
- Klik keempat dan kelima tidak mendapatkan poin (tidak mengenai persegi manapun).