

Cat Rumah

Batas Waktu	1 detik
Batas Memori	256 MB

Deskripsi

Hari ini Pak Dengklek ingin mengecat rumahnya dengan warna favorit bebek-bebeknya. Karena Pak Dengklek hanya memiliki kaleng-kaleng cat dengan warna dasar (merah, kuning, dan biru), maka Pak Dengklek menyiapkan sebuah ember dengan kapasitas maksimal 1.000 cc untuk mencampurkan ketiga warna dasar tersebut menjadi warna favorit bebek-bebeknya.

Pada suatu waktu, Pak Dengklek dapat menambahkan suatu warna dasar sebanyak sejumlah cc ke dalam embernya kemudian mencampurkannya. Perlu diperhatikan bahwa campuran warna tersebut tidak boleh melebihi kapasitas ember. Karena cat yang dipakai Pak Dengklek mudah mengering, maka Pak Dengklek hanya dapat melakukan proses menambahkan dan mencampurkan ini maksimal sebanyak 100 kali. Setelah melakukan proses menambahkan dan mencampurkan ini, campuran warna di ember akan dipakai untuk mengecat rumah.

Warna favorit bebek-bebek Pak Dengklek didapat dengan mencampurkan M cc warna Merah, K cc warna Kuning, dan B cc warna Biru dengan ketentuan $M + K + B = 1.000$. Perlu diperhatikan bahwa nilai M , K , dan B tidak dijamin berupa bilangan bulat.

Sayangnya, bebek-bebek Pak Dengklek tidak mengetahui nama dari warna favorit mereka sehingga mereka hanya dapat memberi tahu nilai kemiripan antara warna favorit mereka dengan warna yang berada pada ember Pak Dengklek. Nilai kemiripan tersebut dihitung dengan ketentuan berikut:

- Dimisalkan campuran warna di ember Pak Dengklek saat ini yaitu X cc warna merah, Y cc warna Kuning, dan Z cc warna Biru.
- Jarak warna antara warna di ember Pak Dengklek dengan warna favorit Bebek (W) dihitung dengan meminimalkan nilai $(|kX - M| + |kY - K| + |kZ - B|)$ untuk suatu nilai riil k .
- Sebagai catatan, pasti setidaknya ada satu dari ketiga nilai $|kX - M|$, $|kY - K|$, atau $|kZ - B|$ yang sama dengan nol untuk mendapatkan nilai W .
- Nilai kemiripan (F) dihitung dengan formula:

Sebagai contoh, apabila nilai M , K , dan B berturut-turut bernilai 200, 600, dan 200; serta nilai X , Y , dan Z berturut-turut bernilai 125, 200, dan 50; maka:

- Untuk mendapatkan nilai W , perhatikan bahwa salah satu dari ketiga nilai adalah 0.
 - Apabila nilai $|kX - M| = 0$, maka $k = 1,6$. Sehingga nilai $(|kX - M| + |kY - K| + |kZ - B|) = 400$.

- Apabila nilai $|kY - K| = 0$, maka $k = 3$. Sehingga nilai $(|kX - M| + |kY - K| + |kZ - B|) = 225$.
- Apabila nilai $|kZ - B| = 0$, maka $k = 4$. Sehingga nilai $(|kX - M| + |kY - K| + |kZ - B|) = 500$.
- Nilai W adalah nilai minimum dari ketiganya, sehingga $W = 225$.
- Nilai F dihitung dengan formula $F = 40$.

Sehingga pada setiap Pak Dengklek melakukan proses menambahkan dan mencampurkan (yang telah dijelaskan pada paragraf 2), bebek-bebek Pak Dengklek akan memberi tahu nilai F kepada Pak Dengklek.

Bantulah Pak Dengklek untuk mengecat rumahnya dengan warna semirip mungkin dengan warna favorit bebek-bebeknya, yaitu mendapatkan nilai F sebesar mungkin!

Format Interaksi

Pada awalnya, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' jika bukan.
- Untuk setiap nilai i di antara 1 hingga banyaknya subsoal, berlaku:
 - jika kasus uji tersebut memenuhi batasan subsoal ke- i , maka karakter ke- i berisi i , atau
 - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- i , maka karakter ke- i berisi karakter '.'

Sebagai contoh apabila label sebuah kasus uji sebuah soal adalah 0 . . 345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Selama interaksi berlangsung, program Anda diharuskan mencetak salah satu dari dua perintah: TAMBAH [WARNA] [TAKARAN] dan SELESAI.

Cetaklah TAMBAH [WARNA] [TAKARAN], jika Anda hendak menambahkan [TAKARAN] cc cat berwarna [WARNA] ke dalam ember. Volume cairan setelah ditambahkan tidak boleh melebihi kapasitas ember yakni 1.000 cc dan merupakan bilangan riil positif dengan 4 angka di belakang koma. Warna yang boleh ditambahkan hanyalah warna MERAH, KUNING, atau pun BIRU. Perhatikan bahwa perintah ini hanya dapat dipanggil maksimal 100 kali. Setelah mencetak perintah ini, program anda akan menerima input sebuah bilangan riil F persen dengan 4 angka di belakang koma sesuai dengan deskripsi soal.

Cetaklah SELESAI, jika Anda hendak menyelesaikan proses menambahkan dan mencampurkan. Campuran cat yang berada di dalam ember akan digunakan Pak Dengklek

untuk mengecat rumahnya. Setelah mencetak perintah ini, interaksi akan selesai dan penilaian akan dilakukan.

Contoh Interaksi

Keluaran Program Peserta (Pak Dengklek)	Umpan Balik <i>Grader</i> (Bebek)	Informasi Tambahan
	0..3	
TAMBAH MERAH 100.0000		
	0.0000	W = 800.0000
TAMBAH KUNING 123.4500		
	11.8159	W = 486.0267
TAMBAH KUNING 76.5500		
	30.7180	W = 300.0000
TAMBAH BIRU 50.0000		
	51.0102	W = 150.0000
TAMBAH MERAH 25.0000		
	40.0000	W = 225.0000
SELESAI		
	(interaksi selesai)	

Penjelasan Contoh Interaksi

Pada kasus tersebut, nilai M, K, dan B berturut-turut bernilai 200, 600, dan 200.

Pada akhirnya, warna yang digunakan Pak Dengklek untuk mengecat rumahnya merupakan campuran dari :

- X = 125 cc warna Merah,
- Y = 200 cc warna Kuning, dan
- Z = 50 cc warna Biru.

Nilai yang didapatkan peserta adalah nilai F terakhir yaitu 40.

Subsoal

Tidak ada subsoal pada soal ini. Poin total yang Anda peroleh adalah akumulasi nilai dari 100 buah kasus uji. Semua kasus uji akan termasuk di dalam setidaknya 1 dari 3 himpunan kasus uji berikut.

Himpunan Kasus Uji 1 (20 kasus uji)

- $M = K$
- Nilai M, K, dan B adalah sebuah bilangan bulat.

Himpunan Kasus Uji 2 (30 kasus uji)

- $B = 500$
- Nilai M , K , dan B tidak dipastikan sebuah bilangan bulat.

Himpunan Kasus Uji 3 (50 kasus uji)

- Nilai M , K , dan B tidak dipastikan sebuah bilangan bulat.

Untuk membantu Anda memahami interaksi, disediakan game yang dapat diakses [di sini](#). Kasus uji yang diberikan pada game ini hanyalah untuk visualisasi dan tidak termasuk dalam penilaian seperti pada game di soal interaktif sebelumnya.

Catatan

Yang perlu diperhatikan adalah bahwa untuk tipe soal interaktif seperti ini, Anda harus selalu memberikan perintah `"fflush(stdout);"` (bagi pengguna C/C++) atau `"flush(output);"` (bagi pengguna PASCAL) setiap kali Anda mencetak keluaran (dengan kata lain, setiap kali ada perintah mencetak keluaran misalnya `write`, `writeln`, `printf`, `cout`, atau `puts`, tepat di bawahnya harus ada perintah `fflush/flush`).

Sebagai contoh, berikut adalah contoh *source code* dalam bahasa Pascal yang selalu hanya menambahkan 100 cc cat untuk masing-masing warna ke dalam ember.

```
var subsoal: string;
    F: double;
    m, k, b: double;

begin
    m := 100;
    k := 100;
    b := 100;

    readln(subsoal);

    writeln('TAMBAH MERAH ', m:0:4);
    flush(output);

    // baca keluran dari grader, meskipun tidak digunakan
    readln(F);

    writeln('TAMBAH KUNING ', k:0:4);
    flush(output);

    // baca keluran dari grader, meskipun tidak digunakan
    readln(F);

    writeln('TAMBAH BIRU ', b:0:4);
    flush(output);

    // baca keluran dari grader, meskipun tidak digunakan
    readln(F);

    writeln('SELESAI');
end.
```

Dan berikut adalah contoh *source code* dalam bahasa C dan C++.

```
#include <stdio.h>
#include <string.h>

char subsoal[100];
double F;
double m, k, b;

int main() {
    m = 100;
    k = 100;
    b = 100;

    scanf("%s", subsoal);

    printf("TAMBAH MERAH %.4lf\n", m);
    fflush(stdout);

    /* baca keluran dari grader, meskipun tidak digunakan */
    scanf("%lf", &F);

    printf("TAMBAH KUNING %.4lf\n", k);
    fflush(stdout);

    /* baca keluran dari grader, meskipun tidak digunakan */
    scanf("%lf", &F);

    printf("TAMBAH BIRU %.4lf\n", b);
    fflush(stdout);

    /* baca keluran dari grader, meskipun tidak digunakan */
    scanf("%lf", &F);

    printf("SELESAI\n");
    fflush(stdout);

    return 0;
}
```

Peringatan

Apabila program Anda melakukan salah satu dari hal-hal di bawah ini:

- melakukan tindakan tidak sesuai format sehingga tidak dikenali oleh *grader*,
- memanggil perintah TAMBAH lebih dari 100 kali,
- menambahkan cat hingga melebihi kapasitas ember, atau
- menambahkan cat dengan takaran bilangan negatif.

maka program Anda akan dihentikan secara otomatis dan Anda memperoleh nilai 0 pada kasus uji yang bersangkutan. Meskipun memperoleh nilai 0, perhatikan bahwa Anda **tetap mendapatkan *verdict Accepted*** untuk kasus uji tersebut.

Reduksi Pesan

Batas Waktu	1 detik
Batas Memori	32 MB

Deskripsi

Pak Dengklek bersama tim risetnya sedang mengembangkan program untuk mereduksi pesan yang berupa sederetan huruf alfabet kapital (A – Z). Program ini sudah selesai dikembangkan, dan kini memasuki tahap pengujian. Sayangnya, Pak Dengklek dan tim kebingungan bagaimana memastikan bahwa program yang mereka kembangkan bekerja dengan baik. Akhirnya ia meminta bantuan Anda untuk membuat program yang dapat membantu mengoreksi pekerjaan Pak Dengklek.

Program yang diinginkan Pak Dengklek sederhana. Anda akan diberikan sebuah untai pesan berisi simbol A – Z (semua kapital) tanpa spasi. Setiap huruf akan dinomori 1 sampai N, dengan N adalah panjang dari pesan itu sendiri. Kemudian Pak Dengklek akan memberikan beberapa pertanyaan yang dinyatakan oleh pasangan bilangan i dan j. Program anda harus mengeluarkan hasil reduksi *substring* dari posisi ke-i hingga posisi ke-j. Adapun aturan reduksi yang diinginkan adalah sebagai berikut:

- Untuk setiap sekumpulan huruf yang saling bersebelahan dan memiliki simbol yang sama, hapus semua kecuali satu. Dengan kata lain, pada hasil reduksi pesan, tidak ada dua huruf yang sama akan saling bersebelahan. Contoh: HAALLLLOOOOO → HALO, MATARAM → MATARAM, dan AAAAA → A.
- Hitung panjang pesan setelah dilakukan proses reduksi. Keluarkan angka ini, karena ini merupakan informasi yang penting untuk Pak Dengklek. Contoh: HAALLLLOOOOO → HALO (4 huruf).
- Jika panjang pesan yang tereduksi **lebih kecil dari 10**, cetak juga pesan hasil reduksi tersebut.

Bisakah Anda menyelesaikan program yang diinginkan Pak Dengklek? Sebagai catatan, Pak Dengklek hanya memberikan Anda sedikit waktu karena sebentar lagi hasil penelitian mereka akan dipublikasikan.

Format Masukan

Pada baris pertama, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' jika bukan.
- Untuk setiap nilai i di antara 1 hingga banyaknya subsoal, berlaku:

- jika kasus uji tersebut memenuhi batasan subsoal ke-i, maka karakter ke-i berisi i, atau
- jika kasus uji tersebut tidak memenuhi batasan subsoal ke-i, maka karakter ke-i berisi karakter '.'

Sebagai contoh apabila label sebuah kasus uji sebuah soal adalah 0..345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris kedua berisi 2 bilangan, N dan Q, dimana N adalah panjang pesan dan Q adalah banyaknya pertanyaan.

Baris berikutnya adalah sebuah pesan yang terdiri dari simbol A – Z, sepanjang N.

Q baris berikutnya berisi 2 buah bilangan i dan j.

Format Keluaran

Untuk setiap pertanyaan, keluarkan panjang pesan setelah melakukan reduksi potongan pesan dari posisi ke-i hingga posisi ke-j. Apabila panjang pesan lebih kecil dari 10, maka tampilkan hasil reduksinya di baris yang sama dipisahkan oleh spasi.

Contoh Masukan

```
0..345
20 5
AAABBCCCCCQWERTYUIOP
1 1
1 3
2 9
1 20
18 20
```

Contoh Keluaran

```
1 A
1 A
3 ABC
13
3 IOP
```

Subsoal

Subsoal 1 (7 poin)

- $N = 15$
- $Q = 10$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 2 (11 poin)

- $N = 100$
- $Q = 10$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 3 (21 poin)

- $1 \leq N \leq 100$
- $1 \leq Q \leq 100$

Subsoal 4 (27 poin)

- $1 \leq N \leq 100$
- $1 \leq Q \leq 100.000$

Subsoal 5 (34 poin)

- $1 \leq N \leq 100.000$
- $1 \leq Q \leq 100.000$

Peringatan

Bagi pengguna C/C++ diwajibkan untuk menggunakan perintah `scanf()` (bukan `cin()`) untuk membaca masukan pada soal ini agar tidak terjebak *Time Limit Exceeded*.

Pelontar Bebek

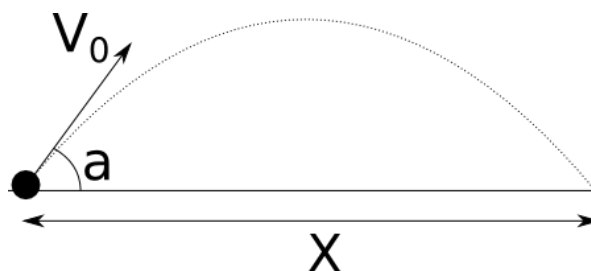
Batas Waktu	1 detik
Batas Memori	32 MB

Deskripsi

Kwek dan Kwak, seperti layaknya bebek lain sedang seru-serunya bermain permainan “Rocket Duck”. Pada Rocket Duck, pemain menjalankan seekor bebek bernama “Rocky” yang ingin mencari arti hidup, yang diyakini berada diujung belahan dunia. Di luar dugaan, Rocky mencari arti hidup dengan cara yang tidak biasa dan ironis: melempar dirinya sejauh mungkin dengan alat pelontar (dan berharap mencapai ujung belahan dunia). Tentu ini cara yang ekstrim dan berbahaya. Tapi siapa peduli, Rocky hanyalah sebuah tokoh imajiner dalam sebuah *game*, yang terpenting bahwa permainan ini berhasil memikat banyak bebek muda.

Semua bebek ambisius mengejar *high-score* di Rocket Duck, begitu pula Kwek dan Kwak. Mereka membayangkan namanya terpampang pada tabel *leaderboard*, yang menjadi sebuah kepuasan tersendiri bagi mereka. Namun Kwek dan Kwak punya kehidupan lain, mereka tidak bisa terus menerus bermain Rocket Duck. Sehingga kemampuan mereka tidak sejago pemain lainnya. Untungnya Kwek dan Kwak kenal dengan Anda, sang programmer handal yang konon dapat menyelesaikan berbagai masalah hanya dengan hentakan jari-jemari di atas *keyboard*. Kini mereka memerlukan bantuan kalian untuk mengejar *high-score* pada permainan Rocket Duck.

Rocket Duck adalah permainan 2D, di mana pemain berusaha mencapai jarak terjauh dengan menggunakan alat pelontar bebek. Alat ini dapat melontarkan bebek pada derajat dan kecepatan awal tertentu. Bebek kemudian bergerak secara parabola, hingga akhirnya jatuh ke tanah. Nilai yang didapat dari pemain adalah jarak yang dicapai. Makin jauh jarak yang dicapai, makin tinggi nilai yang didapat. Mekanisme lontaran roket mengikuti mekanisme dasar gerak parabolik. Berikut ini adalah ilustrasi mekanisme dari gerak parabolik:



Untuk Pemain, akan disediakan beberapa pilihan roda gigi untuk merakit pelontar bebek. Setiap roda gigi memiliki dua properti, yaitu nilai kecepatan K dan nilai sudut S . Sebuah pelontar bebek nantinya akan terdiri dari beberapa roda gigi yang terpasang pada mesin. Sudut akhir dari mesin pelontar ini adalah total nilai sudut dari semua roda gigi yang dipasang. Kecepatan dari mesin pelontar ini adalah kecepatan maksimum dari semua roda gigi yang dipasang.

Sebagai tambahan, mesin pelontar ini tidak bisa menembak dengan sudut yang lebih besar dari 180 derajat. Artinya konfigurasi mesin yang menyebabkan hal tersebut terjadi tidak diperbolehkan. Perlu diperhatikan pula bahwa nilai pemain didapat dari jauhnya jarak. Artinya arah tembak dapat diabaikan (sudut kurang dari 90 derajat memiliki arah tembak berlawanan dengan sudut lebih dari 90 derajat). Jumlah roda gigi yang ada bersifat terbatas, artinya pemain hanya dapat memakai tiap roda gigi maksimal sekali saja.

Untuk membantu, berikut adalah rumus untuk menghitung jarak tembak X pada permainan Rocket Duck, diberikan nilai sudut a dan kecepatan awal V_0 .

$$X = |2 \times V_0^2 \times \sin a \times \cos a \times 9.8|$$

Bantulah Kwek dan Kwak memilih konfigurasi roda gigi optimal, untuk mendapatkan jarak terjauh.

Format Masukan

Pada baris pertama, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' jika bukan.
- Untuk setiap nilai i di antara 1 hingga banyaknya subsoal, berlaku:
 - jika kasus uji tersebut memenuhi batasan subsoal ke- i , maka karakter ke- i berisi i , atau
 - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- i , maka karakter ke- i berisi karakter '.'

Sebagai contoh apabila label sebuah kasus uji sebuah soal adalah 0 . . 345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris kedua adalah bilangan N yang menyatakan banyaknya pilihan roda gigi yang tersedia.

N baris berikutnya terdiri dari dua angka K_i dan S_i , yang menyatakan nilai kecepatan dan nilai sudut dari roda gigi yang bersesuaian. Nilai sudut diberikan dengan skala perbesaran $10\times$ lipat sehingga nilai sudut 900 maksudnya adalah 90 derajat, nilai sudut 135 maksudnya adalah 13,5 derajat, dan sebagainya.

Format Keluaran

Untuk tiap kasus uji, keluarkan sebuah bilangan riil hingga 12 angka di belakang penanda desimal yang menyatakan jarak terjauh yang dapat dicapai dengan konfigurasi roda gigi

optimal. Perbedaan perhitungan absolut atau perhitungan relatif di bawah 0,001 akan diterima.

Contoh Masukan

```
0..3.567
3
50 160
105 320
101 290
```

Contoh Keluaran

```
107453.118185065090
```

Subsoal

Pada semua subsoal, berlaku:

- $1 \leq S_i \leq 1.800$
- K_i dan S_i adalah bilangan bulat.

Subsoal 1 (8 poin)

- $N = 5$
- $1 \leq K_i \leq 100$
- Nilai K tiap roda gigi adalah sama.
- Kasus uji dapat diunduh [di sini](#).

Subsoal 2 (10 poin)

- $N = 5$
- $1 \leq K_i \leq 100$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 3 (13 poin)

- $1 \leq N \leq 10$
- $1 \leq K_i \leq 100$

Subsoal 4 (14 poin)

- $1 \leq N \leq 50$
- $1 \leq K_i \leq 100$
- Nilai K tiap roda gigi adalah sama.

Subsoal 5 (15 poin)

- $1 \leq N \leq 50$
- $1 \leq K_i \leq 100$

Subsoal 6 (19 poin)

- $1 \leq N \leq 1.000$
- $1 \leq K_i \leq 1.000$

Subsoal 7 (21 poin)

- $1 \leq N \leq 500.000$
- $1 \leq K_i \leq 1.000$

Catatan

Untuk melakukan perhitungan $\sin(a)$ atau $\cos(a)$, terdapat fungsi yang sudah ada dan bisa Anda gunakan baik pada Pascal maupun pada C/C++. Namun, kedua fungsi itu menerima sudut bukan dalam derajat, melainkan dalam radian (satuan yang lain untuk sudut). Anda dapat menggunakan fungsi berikut untuk merubah derajat menjadi radian.

Jika menggunakan Pascal, Anda **wajib** menuliskan 'uses math' sebelum mendeklarasikan variabel. Contoh program:

```
uses math;
var
    sudut:double;

function toRadian(a:double):double;
begin
    toRadian := (a * arccos(-1)) / 180;
end;

begin
    sudut := 20; // perhatikan bahwa sudut ini tidak dalam perbesaran 10x
    lipat

    // menghitung sin(20)
    writeln(sin(toRadian(sudut))); // hasilnya: 3.4202014332566871E-0001

    // menghitung cos(20)
    writeln(cos(toRadian(sudut))); // hasilnya: 9.3969262078590839E-0001
end.
```

Jika menggunakan C++, Anda **wajib** menulis '#include <cmath>' pada bagian deklarasi *library* yang akan digunakan. Contoh program:

```
#include <cstdio>
#include <cmath>

double sudut;

double toRadian(double a){
    return (a * acos(-1)) / 180;
}

int main() {
    sudut = 20; /* perhatikan bahwa sudut ini tidak dalam perbesaran
    10x lipat */
```

```
/* menghitung sin(20) */
printf("%lf\n", sin(toRadian(sudut))); /* hasil: 0.342020 */

/* menghitung cos(20) */
printf("%lf\n", cos(toRadian(sudut))); /* hasil: 0.939693 */

return 0;
}
```

Jika menggunakan C, Anda **wajib** menulis '#include <math.h>' pada bagian deklarasi *library* yang akan digunakan. Contoh program:

```
#include <stdio.h>
#include <math.h>

double sudut;

double toRadian(double a){
    return (a * acos(-1)) / 180;
}

int main() {
    sudut = 20; /* perhatikan bahwa sudut ini tidak dalam perbesaran
10x lipat */

    /* menghitung sin(20) */
    printf("%lf\n", sin(toRadian(sudut))); /* hasil: 0.342020 */

    /* menghitung cos(20) */
    printf("%lf\n", cos(toRadian(sudut))); /* hasil: 0.939693 */

    return 0;
}
```

Peringatan

Gunakan tipe data **double** untuk menggantikan tipe data real/float agar ketelitian bilangan riil menjadi lebih tinggi.

Perhatikan potongan-potongan kode program di atas sebagai contoh penggunaannya.

Perang Dunia Ketiga

Batas Waktu	0,5 detik
Batas Memori	64 MB

Deskripsi

Perang dunia ketiga sudah pecah! Kedamaian di Negara Bebek pun terancam. Pak Dengklek yang sangat cinta Negara Bebek pun membuat sebuah senjata canggih yang menggunakan teknologi laser dan diyakini mampu menghancurkan musuh sekuat apapun.

Senjata canggih Pak Dengklek memiliki sebuah layar dan 3 tombol:

1. Layar: Untuk menampilkan kekuatan laser saat ini (harus selalu berupa bilangan asli).
2. Tombol "Tembak": Untuk menembakkan laser ke musuh.
3. Tombol "Kali 2": Membuat laser menjadi 2 kali lebih kuat dari sebelumnya.
4. Tombol "Bagi 2": Membuat kekuatan laser menjadi setengah dari sebelumnya. Jika kekuatan laser sebelum dilakukan operasi ini berupa bilangan ganjil, maka tidak terjadi apa-apa.

Nilai awal kekuatan laser hanya dapat diatur sekali sebelum pasukan musuh datang, karena untuk mengganti kekuatan laser membutuhkan waktu yang sangat lama (kecuali menggunakan tombol "Kali 2" dan tombol "Bagi 2"). Perhatikan bahwa nilai awal kekuatan laser ini harus lebih dari 0.

Senjata laser buatan Pak Dengklek ini tentunya menggunakan daya yang sangat besar. Pak Dengklek berkata bahwa untuk setiap satuan kekuatan laser yang ditembakkan menggunakan 1 unit energi. Tentunya kekuatan setiap pasukan musuh berbeda-beda, sehingga kekuatan laser yang diperlukan untuk menghancurkan mereka pun bisa jadi berbeda-beda. Untungnya, BINB (Badan Intelijen Negara Bebek) sudah mengetahui kekuatan musuh yang akan menyerang Negara Bebek dan sudah menghitung kekuatan laser yang diperlukan untuk menghancurkan masing-masing musuh.

Diberikan kekuatan laser minimum yang dibutuhkan untuk menghancurkan pasukan-pasukan negara musuh, berapa minimal unit energi yang harus terbuang untuk menghancurkan seluruh pasukan musuh? Anda harus menentukan nilai awal kekuatan laser, yang boleh ditentukan secara bebas.

Unit energi yang terbuang untuk menghancurkan satu pasukan musuh didefinisikan sebagai selisih dari unit energi minimal untuk menghancurkan pasukan tersebut dan unit energi yang digunakan Negara Bebek untuk menghancurkan pasukan tersebut. Sedangkan unit energi yang terbuang untuk menghancurkan seluruh musuh adalah jumlah dari unit energi yang terbuang untuk setiap pasukan musuh.

Format Masukan

Pada awalnya, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' jika bukan.
- Untuk setiap nilai i di antara 1 hingga banyaknya subsoal, berlaku:
 - jika kasus uji tersebut memenuhi batasan subsoal ke- i , maka karakter ke- i berisi i , atau
 - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- i , maka karakter ke- i berisi karakter '.'

Sebagai contoh apabila label sebuah kasus uji sebuah soal adalah 0..345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris kedua pada berkas masukan berisi sebuah bilangan bulat N yang menyatakan banyaknya pasukan musuh yang akan menyerang Negara Bebek.

Baris ketiga berisi N buah bilangan bulat non-negatif X_i yang menyatakan kekuatan laser minimum untuk menghancurkan pasukan ke- i .

Format Keluaran

Sebuah baris berisi sebuah bilangan bulat yang menyatakan unit energi yang terbuang minimal yang diperlukan untuk menghancurkan semua pasukan musuh.

Contoh Masukan 1

```
0..3456
2
7 31
```

Contoh Keluaran 1

```
2
```

Contoh Masukan 2

```
0..3456
3
4 8 2
```

Contoh Keluaran 2

Penjelasan

Untuk contoh masukan nomor 1, kekuatan awal laser dapat diset sama dengan 8. Saat pasukan musuh nomor 1 datang, pasukan nomor 1 tersebut dapat langsung dihancurkan dengan menggunakan 8 unit energi (sehingga $8 - 7 = 1$ unit energi terbang). Saat pasukan musuh nomor 2 datang, Pak Dengklek bisa menekan tombol "Kali 2" sebanyak 2 kali dan menghancurkannya dengan menggunakan 32 unit energi (sehingga $32 - 31 = 1$ unit energi terbang). Total 2 unit energi terbang untuk menghancurkan pasukan musuh.

Untuk contoh masukan nomor 2, kekuatan awal laser diset sama dengan 4. Saat pasukan musuh nomor 1 datang, pasukan nomor 1 tersebut dapat langsung dihancurkan dengan menggunakan 4 unit energi (sehingga $4 - 4 = 0$ unit energi terbang). Saat pasukan nomor 2 datang, Pak Dengklek bisa menekan tombol "Kali 2" sebanyak 1 kali dan menghancurkannya dengan menggunakan 8 unit energi (sehingga $8 - 8 = 0$ unit energi terbang). Saat pasukan musuh nomor 3 datang, kita bisa menekan tombol "Bagi 2" sebanyak 2 kali dan menghancurkannya dengan menggunakan 2 unit energi (sehingga $2 - 2 = 0$ unit energi terbang). Total 0 unit energi terbang untuk menghancurkan pasukan musuh.

Subsoal

Subsoal 1 (11 poin)

- $N = 10$
- $X_i = i$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 2 (7 poin)

- $N = 12$
- $0 \leq X_i \leq 10$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 3 (23 poin)

- $1 \leq N \leq 100$
- $0 \leq X_i \leq 100$

Subsoal 4 (12 poin)

- $1 \leq N \leq 3.000$
- $0 \leq X_i \leq 3.000$

Subsoal 5 (31 poin)

- $1 \leq N \leq 100.000$
- $0 \leq X_i \leq 100.000$

Subsoal 6 (16 poin)

- $1 \leq N \leq 1.000.000$
- $0 \leq X_i \leq 1.000.000$

Peringatan

Bagi pengguna C/C++ diwajibkan untuk menggunakan perintah `scanf()` (bukan `cin()`) untuk membaca masukan pada soal ini agar tidak terjebak *Time Limit Exceeded*.