ছোট থেকে বড়! ( Lower to Upper! )

CPU: 2s  
Memory: 1024MB

তোমাকে T সংখ্যক ইংরেজিতে লেখা নাম দেওয়া আছে। নামগুলোতে ইংরেজি বড় হাতের A থেকে Z অক্ষর, ছোট হাতের a থেকে z অক্ষর এবং স্পেস দেওয়া থাকতে পারে। তোমাকে নামগুলোর সবগুলো অক্ষরগুলোকে বড় হাতের বানাতে হবে। অর্থাৎ, যেই অক্ষরগুলো আগে ছোট হাতের ছিল, সেগুলো এখন বড় হাতের হয়ে যাবে। আর যেই অক্ষরগুলো আগে থেকেই বড় হাতের ছিল, সেগুলো এখনো ঠিক আগের মতই বড় হাতেরই থেকে যাবে। স্পেসগুলো ঠিক ইনপুটে যেভাবে দেওয়া থাকবে, সেই জায়গাতেই রেখে দিতে হবে।

You are given T names in english letters. Each name will include some of the uppercase letters from A to Z, some of the lowercase letters from a to z and some spaces. You have to transform the name from lowercase to uppercase. The uppercase letters that were uppercase before will remain uppercase and the spaces will also remain in their places.

ইনপুটের বর্ণনা

ইনপুট ফাইল এর প্রথম লাইন এ থাকবে একটি পূর্ণ সংখ্যা **T (1 <= T <=১০০০০)**যা ফাইল এ কতগুলো ইনপুট সেট আছে তা বুঝাবে। এরপরের **T** সংখ্যক লাইনের প্রতিটিতে একটি করে নাম থাকবে। প্রতিটি নামে স্পেসসহ সর্বোচ্চ **১০০** টি করে অক্ষর থাকবে।

Input Specification

The first line contains a positive integer **T ( 1<=T<=10000 )**, denoting the number of test case. The next **T** lines will contain a name. The length of a name will not exceed **100** characters.

আউটপুট এর বর্ণনা

প্রথম লাইন ব্যতিত অন্য সকল ইনপুট লাইনের জন্য এক লাইন আউটপুট দিতে হবে। প্রতিটা কেসের জন্য কেসের ক্রম এবং সেই কেসটার ইনপুটে দেয়া নামটি বড় হাতের অক্ষরে প্রিণ্ট করতে হবে। সঠিক ফরম্যাট জানার জন্যে উদাহরণের ইনপুট ও আউটপুটগুলো দেখে নাও।

Output Specification

For each case, print the case number and the name in uppercase letters. See the samples for exact formatting.

Sample

| **Input** | **Output** |
| --- | --- |
| 5  Hasnain Heickal Jami  Mir Wasi Ahmed  Tarif Ezaz  Mahmud Ridwan  Md Mahbubul Hasan | Case 1: HASNAIN HEICKAL JAMI  Case 2: MIR WASI AHMED  Case 3: TARIF EZAZ  Case 4: MAHMUD RIDWAN  Case 5: MD MAHBUBUL HASAN |

#include <stdio.h>

#include <string.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

getchar();

while (current\_tc < tc) {

int i;

char name[101];

int length;

//scanf(" %[^\n]", name);

gets(name);

//printf("%s", name);

length = strlen(name);

for (i = 0; i < length; i++) {

if (name[i] >= 97 && name[i] <= 122) {

name[i] = 'A' + (name[i] - 'a');

}

}

printf("Case %d: %s\n", current\_tc + 1, name);

current\_tc++;

}

return 0;

}

সর্বোচ্চ সংখ্যা (Maximum Number)

CPU: 0.5s  
Memory: 512MB

তিনটি পূর্ণ সংখ্যা দেওয়া আছে, তোমাকে তাদের মধ্যে সর্বোচ্চ সংখ্যার মান বের করতে হবে।

Given three integers A, B and C, you have to output the maximum number between them.

ইনপুটের বর্ণনা

প্রথম লাইনে থাকবে কেসের সংখ্যা, একটি ধনাত্মক পূর্ণসংখ্যা T (T <= ১০০)। তার পরের লাইনে থাকবে তিনটি করে পূর্ণসংখ্যা। A, B এবং C-এর পরমমান ২০০ থেকে বেশি হবে না। ( |A|,|B|,|C| <= 200 )

Input Specification

The first line contains a positive integer T ( T <= 100 ), number of test case. In the following T lines, you will be given three integers. The absolute value of A and B and C will not exceed 200.

আউটপুটের বর্ণনা

প্রতিটি কেসের জন্যে কেস নাম্বার এবং তার পর সর্বোচ্চ সংখ্যার মান প্রিন্ট দিতে হবে।

Output Specification

For each case print the case number, and the maximum number between the three integers.

Sample

| **Input** | **Output** |
| --- | --- |
| 4  1 2 3  3 2 1  -3 -2 -2  2 2 2 | Case 1: 3  Case 2: 3  Case 3: -2  Case 4: 2 |

#include <stdio.h>

int maximum(int a, int b);

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int a, b, c;

int max;

scanf("%d %d %d", &a, &b, &c);

max = maximum(a, b);

max = maximum(max, c);

printf("Case %d: %d\n", current\_tc + 1, max);

current\_tc++;

}

return 0;

} // end main

int maximum(int a, int b) {

if (a > b) return a;

return b;

} // end max

তারার পিরামিড (Pyramid of Stars)

CPU: 1s  
Memory: 1024MB

মারকাস সম্প্রতি মিশর থেকে ঘুরে আসলো। এখন সে পিরামিডের প্রেমে পরেছে। এইজন্যে সে পিরামিড আঁকবে চিন্তা করেছে। যেহেতু সে এখনো নতুন, তার আঁকা পিরামিডগুলো বাম্ পাশে ঘেঁষে থাকে। নিম্নে তার দ্বারা আঁকা একটি ৫ উচ্চতার পিরামিডের নমুনা দেয়া হল।

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*

Pyramid with height = 5

এখন সে N উচ্চতার পিরামিড আঁকতে চায়। সে তোমার সাহায্য খুজছে কাজটির জন্যে। তোমাকে যদি N দেয়া হয়, অনুগ্রহ করে তার মত করে N উচ্চতার একটি পিরামিড আঁক।

Marcus recently visited Egypt. Now he has fallen in love with the pyramids. He wants to draw pyramids now. But as he is still amateur, his pyramids will be left aligned. You can see a example of pyramid with height 5 in the figure above. Now he wants to draw a pyramid with height N. He seeks your help for this. Please help him in drawing a pyramid of height N.

ইনপুটের বর্ণনা

প্রথম লাইনে থাকবে একটি ধনাত্মক পূর্ণসংখ্যা T (T <= ১০০ ), যেটি হল টেস্ট কেসের সংখ্যা। তার পরের T লাইনে থাকবে একটি পূর্ণ সংখ্যা N ( ১ <= N <= ১০০ ), যেটি হল পিরামিডের উচ্চতা।

Input Specification

The first line contains a positive integer T (T <= 100 ), number of test case. In the following T lines, you will be given a positive integers N (1<=N <= 100) which is the height of the pyramid.

আউটপুটের বর্ণনা

প্রতিটি কেসের জন্য কেস নম্বর প্রিন্ট দিয়ে, তার পরের লাইন থেকে N উচ্চতার একটি পিরামিড আঁকতে হবে।

Output Specification

For each case print the case number, and then the pyramid with height = N.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  1  2  3 | Case 1:  \*  Case 2:  \*  \*\*  \*  Case 3:  \*  \*\*  \*\*\*  \*\*  \* |

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i, j;

int height;

int row;

scanf("%d", &height);

//row = (height \* 2) - 1;

printf("Case %d:\n", current\_tc + 1);

for (i = 1; i <= height; i++) {

for (j = 0; j < i; j++) {

printf("\*");

}

printf("\n");

}

for (i = height - 1; i > 0; i--) {

for (j = 0; j < i; j++) {

printf("\*");

}

printf("\n");

}

current\_tc++;

}

return 0;

} // end main

গরিষ্ঠ সাধারণ গুণনীয়ক ( Greatest Common Divisor )

CPU: 1s  
Memory: 1024MB

পিটার ও টুরিস্ট ক্রিকেট খেলছিল। তাদের খেলার নিজেদের নিয়ম আছে। পিটার অনেক ভাল খেলে। সে টসে জিতে ব্যাটিং নিল এবং **A** রান করল। তারপর টুরিস্ট সেই রানের তাড়া শুরু করল। খেলা অনেক কঠিন ছিল এবং টুরিস্ট **B** রান করার পর হঠাৎ করে বৃষ্টি শুরু হল এবং তারা দ্বন্দ্বে পড়ে গেল এখন কি করা যায়। তাদের খেলায় বৃষ্টির জন্য কোন নিয়ম ছিল না। কিন্তু টুরিস্ট অনেক বুদ্ধিমান এবং চালাক। সে বলল বৃষ্টির পর তার নতুন টার্গেট হবে **C = B + GCD( A , B )**। পিটার এই নিয়মে খুশিই ছিল। সে দ্রুত দুইটি সংখ্যা **A** ও **B** এর গসাগু বের করার জন্য একটি প্রোগ্রাম লিখে ফেলল। আমাদের এই সমস্যাটা ঠিক একই রকম। তোমাকে **A** এবং **B** এর মান দেওয়া হবে। তোমার কাজ হবে তাদের গসাগু বের করা।

Petr and Tourist were playing cricket (They have their own rules for cricket). Petr is very good in cricket. He won the toss and scored **A** runs. Then Tourist started chasing Petr's score. It was indeed a tough match and Tourist scored **B** runs, where **B** is less than or equal to **A**. Suddenly it starts to rain and they got confused about what to do. They didn't have exact rules for raining. But as Tourist is too much talented and clever, he said after the rain stops his new target will be **C = B + GCD( A , B )**. Petr was happy about this new formula. Soon he wrote a code to find the Greatest Common Divisor ( GCD ) of **A** and **B**. This problem is also similar. You'll be given **A** and **B**. You have to find their GCD.

ইনপুটের বর্ণনা

প্রথম লাইনে ইনপুটের সংখ্যা **T (0<T<=10000)** দেওয়া থাকবে। পরবর্তী **T** লাইনে দুইটা করে পূর্ণসংখ্যা **A** এবং **B** দেয়া থাকবে। এখানে, **0<B<A<=1000000**

Input Specification

First line will have the number of test cases **T (0 < T<=10000)**. And in next **T** lines there will be two integers **A** and **B**. Here **0<B<A<=1000000**.

আউটপুট এর বর্ণনা

প্রতিটি ইনপুটের জন্য তোমাকে শুধুমাত্র **A** ও **B** এর গরিষ্ঠ সাধারণ গুণনীয়ক প্রিন্ট করতে হবে।

**৩০** শতাংশ পয়েন্টের জন্য, **0<T<=1000** এবং **0<B<A<=1000**

**১০০** শতাংশ পয়েন্টের জন্য, ইনপুট বর্ণনা দেখো।

Output Specification

For each input you just need to print the GCD of **A**,**B**.

For **30%** points, **0<T<=1000** and **0<B<A<=1000**

For **100%** points, follow input constraints.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  2 1  10 5  200 100 | 1  5  100 |

#include <stdio.h>

int gcd(int num1, int num2);

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int num1, num2;

scanf("%d %d", &num1, &num2);

printf("%d\n", gcd(num1, num2));

current\_tc++;

}

return 0;

} // end main

int gcd(int num1, int num2) {

int temp;

while (num2 != 0) {

temp = num2;

num2 = num1 % num2;

num1 = temp;

}

return num1;

} // end gcd

সর্বশেষ অংক(Last Digit)

CPU: 0.5s  
Memory: 512MB

X এবং P দেয়া থাকলে নিচের যোগফলের সর্বশেষ (সর্বডানের) অঙ্ক কি হবে?

Given X and P, what is the last (Rightmost or Least Significant) digit of the following sum?

X0 + X1 + X2 + X3 + ……. + XP-1 + XP

ইনপুটের বর্ণনা

শুধু একটি লাইন, space দ্বারা পৃথক করা X এবং P এর মান।

Input Specification

Just one line, space separated values of X and P.

For 30% cases 1<=X<=9 and 1<=P<=10

For 30% cases 1<=X<=999 and 1<=P<=100000

For 40% cases 1<=X<=999999 and 1<=P<=10^15

আউটপুটের বর্ণনা

শুধু একটি লাইন, যোগফলের সর্বশেষ (সর্বডানের) অঙ্ক

Output Specification

Just one line, the result.

Samples

| **Input** | **Output** |
| --- | --- |
| 1 5 | 6 |

| **Input** | **Output** |
| --- | --- |
| 2 6 | 7 |

| **Input** | **Output** |
| --- | --- |
| 3 7 | 0 |

| **Input** | **Output** |
| --- | --- |
| 4 8 | 1 |

#include <stdio.h>

int power(int num, int pow) {

int i;

int result = 1;

for (i = 0; i < pow; i++) {

result = result \* num;

}

return result;

}

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int i;

int x, p;

int sum = 0;

scanf("%d %d", &x, &p);

sum = (power(x, p + 1) - 1) / (x - 1);

printf("%d\n", sum % 10);

return 0;

} // end main

গুনতে পারবে কি ? ( Can You Count? )

CPU: 5s  
Memory: 1024MB

তোমরা হয়ত মি. হতাশ এর কথা শুনে থাকবে। লোকটা ছোট বেলায় শুধু বড় বড় কথা বলত আর চিন্তা করত! একদিন সে ভাবলো একদিনের জন্য সে ছুটি নিবে। কোন কাজ নেই , বেচারা আকশের দিকে তাকিয়ে আছে। হটাৎ করে সে একটা তারার দিকে তাকিয়ে ভাবল , "এটা কি আসলেই এমন? নাকি ভেজাল আছে?" যেমনি ভাবা তেমনি কাজ …… ভেজাল তাকে প্রমাণ করতেই হবে । অবশেষে অনেক সাধনা করে আইন্সটাইন এর রিলেটিভিটি ঘাটাঘাটি করতেই সে বুঝতে পারল “এই তারা আসলে সেই তারা , আমরা তো আসলে একটা টাইম মেশিনের দিকে তাকিয়ে আছি” – অসাধারন …! তাই নয় কি ? মানুষ এত চেস্টা করে টাইম মেশিন বানাতে চায় , কিন্তু টাইম মেশিন তো চোখের সামনে, দেখা যায় কিন্তু ব্যবহার করা যায় না ,আকাশ তো পুরোটাই একটা টাইম মেশিন ।

কিন্তু একটা কাজ ধরেছে মি. হতাশ, হতাশ না হয়ে বাড়ি ফিরলে কি হয়? তো কি করা যায়? হতাশ হবার আশায় সে তারা গোনা আরম্ভ করল। কিন্তু তারা গুনে কি আর শেষ হয়? সে তারা গোনে আর লিখে এ তারাটা তার কতটুকু ভালো লেগেছে। প্রত্যেক তারারই ভালো লাগার একটা মান আছে। কোন দুটো তারারই ভালো লাগার মান সমান হতে পারে না। কিন্তু মি. হতাশ বার বার ভুল করে একই তারার মান খাতায় লিখে ফেলে, আকাশ তো আর ছোট না । আকাশ দেখেই তো মানুষ অসীমকে দেখতে পায়।

তোমার কাছে এখন মি. হতাশ এর খাতার ভ্যালু গুলো আছে, তোমার বলতে হবে মি. হতাশ হতাশ হবার আগে কতগুলো আলাদা তারা গুনেছে। চিন্তা করো না, তারা গুনতে গুনতেই হয়ত তুমি একদিন গ্যালিলিও , আইন্সটাইন হয়ে যাবা। যাই হোক গ্যালিলিও, আইন্সটাইন এর কম্পিঊটার ছিলো না, তোমার আছে ।

You may have heard the name of Mr. Hotash. When he was young he used to talk like elders and ponder!!! Once he thought he'll take a-one day leave. With no work at hand, he was staring at the sky. Suddenly looking at a star he thought, Is it really like that? Or there's some problems with the star. Thus he thought it's a must to prove that the star has some problems. So doing a hell of research at last he found from the Einstein's Theory of Relativity, ”All these stars are from the past, We are looking at a cosmic time machine” - Great...!!! Isn't it?

Humans are trying so much to build one time machine, but it's right in front of their eyes, they can see it but can't use it, the whole sky is one big time machine. At last Mr. Hotash found some work, but does it suit Mr. Hotash to return to his home without being disappointed.

So thinking of that else he can do he found that counting stars will be great job to become disappointed. But is it possible to count all the stars! He started to count the stars and for each of them writes a satisfaction index that how much he liked the stars. Every star has a satisfaction index and they are unique. But Mr. Hotash did a mistake, he wrote satisfaction index of some stars multiple times since the sky is too big. Humans define infinity using the sky.

Now you are given the satisfaction indexes written in the notebook of Mr. Hotash. Can you count the number of stars Mr. Hotash noted before he became disappointed. Don't worry, by counting this values you may one day become Galileo or Einstein. Galileo or Einstein didn't have computers in their time but you have one in front of you.

ইনপুটের বর্ণনা

ইনপুট ফাইল এর প্রথম লাইন এ থাকবে একটি ধনাত্মক পূর্ণ সংখ্যা **T ( T <= 1000 )** যা ফাইল এ কতগুলো ইনপুট সেট আছে তা বুঝাবে। এর পরে একটা একটা করে প্রতিটি ইনপুট সেট থাকবে। প্রতিটা ইনপুট সেটের প্রথম লাইন এ একটি ভ্যালু **N ( N <= 10000 )** থাকবে যেটা পরর্তী লাইন এ কতগুলো ভ্যালু আছে তা জানাবে। পরবর্তী লাইনের ভ্যালু গুলোই মি. হতাশের লেখা ভ্যালু। মি. হতাশের প্রত্যেক ভ্যালুর মান ভ্যালুর মান **-10 12** থেকে **10 12** এর মধ্যে হবে।

Input Specification

The first line of input will contain an integer **T ( T <= 1000 )**, which will indicate the number of testcases in the file. Each of the testcases will start with an integer **N ( N <= 10000 )**, which will indicate the number of values in each case. The next **N** values are the written values of Mr. Hotash.

Each value written by Mr. Hotash will be within **-10 12** to **10 12**

আউটপুট এর বর্ণনা

আউটপুটে কেস নাম্বার এবং উত্তরটি লিখতে হবে। বিস্তারিত উদাহরনে দেখ।

Output Specification

Each output will show the case number and the answer of the testcase. Please see the sample cases for the exact formatting.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  3  1 2 3  5  0 1 0 2 3  2  1000 1000 | Case 1: 3  Case 2: 4  Case 3: 1 |

#include <iostream>

#include <set>

using namespace std;

int main() {

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

cin >> tc;

while (current\_tc < tc) {

int i;

set< long long int > s;

int total\_star;

long long int num;

cin >> total\_star;

for (i = 0; i < total\_star; i++) {

cin >> num;

s.insert(num);

}

cout << "Case " << current\_tc + 1 << ": " << s.size() << endl;

current\_tc++;

}

return 0;

}

যোগ করতে পার কি?(Can you sum them?)

CPU: 0.5s  
Memory: 512MB

তোমাকে পূর্ণ সংখ্যার একটি অ্যারে দেয়া হবে। তোমাকে অ্যারের প্রথম সংখ্যাটি থেকে শুরু করে, পরপর দুটি সংখ্যা না নিয়ে অ্যারেটির সঙ্খাগুলোর যোগফল বের করতে হবে।

You will be given an array of integer numbers as input. Starting with the first number, you have to find the sum of the numbers in the array such that you cannot take two successive numbers in the sum.

ইনপুটের বর্ণনা

লাইন ১ – ইন্টিজার N, অ্যারেটিতে নম্বরের সংখ্যা। (1 ≤ 𝑁 ≤ 100000)

লাইন ২ – Nটি নম্বরের একটি অ্যারে। 0 ≤ 𝑥 ≤ 2^31

Input Specification

Line 1 – Integer N, the number of integers in the series ( 1 ≤ 𝑁 ≤ 100000 )

Line 2 – A series of N integers in the range.0 ≤ 𝑥 ≤ 2^ 31

আউটপুট এর বর্ণনা

অ্যারের প্রথম নম্বরটি থেকে শুরু করে, পরপর দুটি নম্বর না নিয়ে অ্যারেটির সংখ্যাগুলোর যোগফল বের কর।

Output Specification

Starting with the first one, output the sum of the numbers in the series without taking two successive numbers.

Samples

| **Input** | **Output** |
| --- | --- |
| 2  1 0 | 1 |

| **Input** | **Output** |
| --- | --- |
| 3  1 2 3 | 4 |

#include <stdio.h>

int main()

{

int i;

int n;

scanf("%d", &n);

long long int num[n];

long long int sum;

sum = 0;

for (i = 0; i < n; i++) {

scanf("%lld", &num[i]);

}

for (i = 0; i < n; i += 2) {

sum += num[i];

}

printf("%lld\n", sum);

return 0;

}

গড় নম্বর (Average Marks)

CPU: 0.5s  
Memory: 512MB

একটি শ্রেণিতে Nটি ছাত্র আছে। কিছুদিন আগেই তারা একটি পরীক্ষা দিয়েছিল এবং আজকে তাদের পরীক্ষার খাতা দেখান হচ্ছে। এখন শিক্ষক তাদের সবার মার্কের গড় জানতে চান। তোমাকে N এবং Nটি ছাত্রের পরীক্ষার মার্ক দেওয়া হয়েছে। তোমাকে তাদের গড় মার্ক গণনা করতে হবে।

In a class, there are N students. Recently they gave an exam and today teacher is giving away their exam copies. The teacher wants to find the average score of the class. So, given N and N marks which each of the students obtained, find the average score of the class.

ইনপুটের বর্ণনা

প্রথম লাইনে একটি ধনাত্মক সংখ্যা T দেওয়া আছে, যেটি হল কতগুলো কেস আছে তার মান। তার পরে Tটি লাইন আছে, যার প্রথমে আছে একটি সংখ্যা, N ( ১ <= N <= ১০০ ), ছাত্রদের সংখ্যার মান। N-এর পরে আছে Nটি সংখ্যা যেগুলো হল ছাত্রদের মার্ক। ছাত্ররা ০ থেকে ১০০ এর ভিতরেই মার্ক পেয়েছে।

Input Specification

The first line contains a positive integer T ( T <= 100 ), number of test case. In the following T lines, the first integer will be N ( 1 <= N <= 100 ), the number of students. After that N integers will follow which are the marks of students. A student will obtain mark between 0 and 100 inclusive.

আউটপুটের বর্ণনা

প্রতিটি কেসের জন্যে প্রথমে কেস এর মান প্রিন্ট দিতে হবে, তারপর একটি পূর্ণসংখ্যা যেটি হল শ্রেণীকক্ষের গড়।

Output Specification

For each case print the case number, then a single integer which is the average score of the class.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  5 1 2 3 4 5  3 100 0 20  4 0 0 100 100 | Case 1: 3  Case 2: 40  Case 3: 50 |

বিঃদ্রঃ গড় এর মান সব কেসের জন্যে পূর্ণসংখ্যা হবে নিশ্চিত করা হয়েছে।

#include <stdio.h>

//#include <math.h>

//#include <string.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i;

int n, sum;

scanf("%d", &n);

//int num[n];

int num;

sum = 0;

for (i = 0; i < n; i++) {

scanf("%d", &num);

sum += num;

}

printf("Case %d: %d\n", current\_tc + 1, sum / n);

current\_tc++;

}

return 0;

} // end main

সংখ্যা গণনা (Counting Numbers)

CPU: 0.5s  
Memory: 512MB

একটি শ্রেণিতে Nটি ছাত্র আছে। কিছুদিন আগেই তারা একটি পরীক্ষা দিয়েছিল এবং আজকে তাদের পরীক্ষার খাতা দেখান হচ্ছে। এখন শিক্ষক জানতে চান কোন নম্বরটি সবচেয়ে বেশি সংখ্যক ছাত্র পেয়েছে এবং কত জন পেয়েছে।

In a class, there are N students. Recently they gave an exam and today teacher is giving away their exam copies. The teacher wants to find the number which is obtained by most of the students and the number of students who obtained that number.

ইনপুটের বর্ণনা

প্রথম লাইনে একটি ধনাত্মক সংখ্যা T দেওয়া আছে, যেটি হল কতগুলো কেস আছে তার মান। তার পরে Tটি লাইন আছে, যার প্রথমে আছে একটি সংখ্যা, N, ছাত্রদের সংখ্যার মান। N-এর পরে আছে Nটি সংখ্যা যেগুলো হল ছাত্রদের মার্ক। ছাত্ররা ০ থেকে ১০০ এর ভিতরেই মার্ক পেয়েছে।

সাবটাস্ক ১: 1<= N <= 1000, ৫০ পয়েন্টের জন্য।

সাবটাস্ক ২: 1<= N <= 100000, ৫০ পয়েন্টের জন্য।

Input Specification

The first line contains a positive integer T ( T <= 100 ), number of test case. In the following T lines, the first integer will be N, the number of students. After that N integers will follow which are the marks of students. A student will obtain mark between 0 and 100 inclusive.

Subtask 1: 1 <= N <= 1000, for 50 points.

Subtask 1: 1 <= N <= 100000, for 50 points.

আউটপুটের বর্ণনা

প্রতিটি কেসের জন্যে প্রথমে কেস এর মান প্রিন্ট দিতে হবে, তারপর দুইটি পূর্ণসংখ্যা যেটি হল কোন নাম্বারটি সবচেয়ে বেশি সংখ্যক ছাত্র পেয়েছে এবং কতজন ছাত্র ঐ নম্বরটি পেয়েছে। যদি এরকম দুইটি নাম্বার থাকে তাহলে বড়টি উত্তর হিসাবে দিতে হবে।

Output Specification

For each case print the case number, then two integers where first one is the number with maximum frequency and then the frequency of that marks. If several such outputs are possible, output the one with maximizes marks.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  5 1 2 3 4 5  3 100 20 20  4 0 0 100 100 | Case 1: 5 1  Case 2: 20 2  Case 3: 100 2 |

Large input file. Use scanf/printf.

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i;

int total;

int index, max;

scanf("%d", &total);

int marks[total];

int marks\_count[101];

for (i = 0; i < total; i++) {

scanf("%d", &marks[i]);

}

for(i = 0; i < 101; i++) {

marks\_count[i] = 0;

}

for(i = 0; i < total; i++) {

marks\_count[marks[i]]++;

}

max = marks\_count[0];

index = 0;

for(i = 0; i < 101; i++) {

if ((marks\_count[i] >= max) && (i > index)) {

max = marks\_count[i];

index = i;

}

}

printf("Case %d: %d %d\n", current\_tc + 1, index, max);

current\_tc++;

}

return 0;

} // end main

শব্দগুলো গুণে ফেলো ( Count All Words )

CPU: 2s  
Memory: 1024MB

দুই বন্ধু সাকিব এবং তামিম স্ট্রিং অনেক পছন্দ করে। একদিন তারা rap song শুনছিল। আমরা সবাই জানি rap song এর প্রতিটা শব্দ বোঝা অনেক কঠিন। এখান থেকে তারা একটা খেলার ধারণা পেল। খেলাটা অনেক সহজ, rap song শুনে তাদেরকে শব্দ গুলোকে খাতায় লিখতে হবে। গান শেষ হওয়ার সাথে সাথে খেলাও শেষ হবে। খেলা শেষ হওয়ার পর তারা দেখল তারা প্রত্যেকেই কিছু শব্দ লিখতে পারেনি বা একই শব্দ একাধিক বার লিখেছে। তাই মোট কতগুলো ভিন্ন শব্দ তারা লিখেছে এটা গণনা করা তাদের জন্য কঠিন হয়ে দাঁড়িয়েছে। যেহেতু তুমি সাকিব ও তামিমের এর শহরের সবচেয়ে বড় প্রোগ্রামার, তাই তারা তোমার কাছে সাহায্যের জন্য এসেছে।

এই প্রবলেমে তোমাকে **N** সংখ্যক শব্দ দেওয়া থাকবে। তোমাকে এর মদ্ধে কয়টা ভিন্ন শব্দ আছে তার সংখ্যা বের করতে হবে।

Two friends Shakib and Tamim love strings. One day they were listening to a rap song. We all know that it's very difficult to understand every words of a rap song. Soon they found an idea of a game. The game is simple, they will note as many words as possible listening to the song. The game ends when the song finishes. When the game ended they noticed that each of them may have missed some words or may have written the same word multiple times. So they found it difficult to count all words which are unique combining both Shakib and Tamim's results. As you are the great programmer of there city, they asked for your help.

In this problem you'll be given **N** words. How have two find the number of unique words.

ইনপুটের বর্ণনা

ইনপুট এর প্রথম লাইনে টেস্ট কেস সংখ্যা **T (0<T<=100)** দেওয়া থাকবে। প্রত্যেকটি কেস একটা ধনাত্মক সংখ্যা **N (0<N<=1000)** দিয়ে শুরু হবে। পরের **N** সংখ্যক লাইনে একটা করে শব্দ দেওয়া থাকবে। শব্দ গুলো কেবল ইংরেজি ছোট হাতের অক্ষরে হবে যার দৈর্ঘ্য **100** থেকে কম বা সমান কিন্তু **1** থেকে কম নয়।

Input Specification

Input starts with the number of test cases **T (0<T<=100)**. Each case starts with a positive integer **N (0<N<=1000)**. Each of the next **N** lines will contain one word. Words will be consisting of only lower case English alphabets and its length will be not more than **100** and not less than **1**.

আউটপুট এর বর্ণনা

আউটপুটঃ প্রত্যেক টেস্ট কেস এর জন্য একক শব্দের সংখ্যা প্রিন্ট করতে হবে।

Output Specification

For each case print the number of unique words.

টেস্টকেসের বর্ণনা

* ৫০ ভাগ মার্ক এর জন্য **(0<T<=20)**, **0<N<=70** এবং শব্দের দৈর্ঘ্য **৭০** এর কম কিন্তু **১** এর কম নয়।
* **১০০** ভাগ মার্ক এর জন্য মূল ইনপুট এর সীমা প্রযোজ্য।

Constraints

* For 50% marks, **(0<T<=20)**, **0<N<=70** and string length will be less than **70** but not empty.
* For **100%** marks, follow the input constraint.

Sample

| **Input** | **Output** |
| --- | --- |
| 1  5  qwerty  ytrewq  asdf  asdf  zxcv |  |

include <iostream>

#include <set>

using namespace std;

int main() {

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

cin >> tc;

while (current\_tc < tc) {

int i;

set< string > s;

int total;

string word;

cin >> total;

for (i = 0; i < total; i++) {

cin >> word;

s.insert(word);

}

cout << s.size() << endl;

current\_tc++;

}

return 0;

}

A এবং B এবং অপারেটর ( A and B and Operators )

CPU: 2s  
Memory: 1024MB

তোমাকে দুইটি সংখ্যা A এবং B এবং একটি অপারেটর দেওয়া আছে। অপারেটরগুলো {+,­-,%,/,\*} হতে পারে অর্থাৎ যোগ, বিয়োগ, মডুলাস, ভাগ কিংবা গুণ অপারেটর হতে পারে। তোমাকে A {operator} B বের করে উত্তর দিতে হবে। মনে করো, যদি তোমাকে 5 এবং 3 এই দুইটি সংখ্যা এবং + অপারেটরটি দেয়া থাকে, তাহলে তোমাকে 5 + 3 এর উত্তর বের করে প্রিণ্ট করতে হবে।

Given two integers A and B and one of the following operators {+,­-,%,/,\*}. You have to perform A {operator} B and output its result. For example, if you are given 5 and 3 as two numbers and + as the operators, then you have to give the answer of 5 + 3.

ইনপুটের বর্ণনা

ইনপুটের প্রথম লাইনে একটি ধনাত্মক পূর্ণসংখ্যা T ( T <= 100 ) দেওয়া থাকবে। এরপরের T লাইনের প্রতিটিতে দুটি করে সংখ্যা A ও B এবং একটি ক্যারেক্টার দেয়া থাকবে। A ও B এর পরমমান কখনো 10^9 অতিক্রম করবে না।

Input Specification

The first line contains a positive integer T ( T <= 100 ). In the following T lines, you will be given two integers A, B and a character. The absolute value of A and B will not exceed 10^9.

আউটপুট এর বর্ণনা

প্রতিটা কেসের জন্য শুরুতে কেস নাম্বার প্রিণ্ট করতে হবে এবং তারপর A {অপারেটর} B এর উত্তর দিতে হবে। সঠিক ফরম্যাটটি জানার জন্য Sample বা উদাহরণের ইনপুট এবং আউটপুট কেসগুলো দেখে নিতে পারো।

Output Specification

For each case print the case number, and then A {operator} B result. Check the sample input and output for further details.

Sample

| **Input** | **Output** |
| --- | --- |
| 5  10 4 +  10 4 ­-  10 4 %  10 4 /  10 4 \* | Case 1: 14  Case 2: 6  Case 3: 2  Case 4: 2  Case 5: 40 |

দ্রষ্টব্য

**'%' ও '/' এই দুটি অপারেটারের জন্য ইনপুটে B এর মান কখনো ০ দেয়া হবে না। ভাগ করার বেলায়, অর্থাৎ '/' অপারেটরের বেলায় শুধুমাত্র পূর্ণসংখ্যাটুকু প্রিন্ট করতে হবে। অর্থাৎ 5 / 2 করলে শুধুমাত্র 2 প্রিন্ট করলেই চলবে, ভগ্নাংশ প্রিন্ট করলে উত্তর ভুল হবে।**

Notes

**For operator ‘%’ and ‘/’, B will not be 0. For the division operator '/', you just need to print the whole part. That is, in case of 5 / 2, you just need to print 2, not the fraction part.**

#include <stdio.h>

//char sign[6] = {'+', '-', '\*', '/', '%'};

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

long long int a, b;

char sign;

long long int result;

scanf("%lld %lld %c", &a, &b, &sign);

if (sign == '+') result = a + b;

if (sign == '-') result = a - b;

if (sign == '\*') result = a \* b;

if (sign == '/') result = a / b;

if (sign == '%') result = a % b;

printf("Case %d: %lld\n", current\_tc + 1, result);

current\_tc++;

}

return 0;

} // end main

গ্রেডিং পদ্ধতি ( Grading Method )

CPU: 1s  
Memory: 1024MB

মনে করো তোমার শিক্ষককে তুমি সাহায্য করছো। তোমার শিক্ষক তোমাকে একটি গ্রেডার বানাতে বলেছে। তিনি তোমাকে N জন শিক্ষার্থীর প্রাপ্ত নম্বর দিবেন এবং তোমাকে N জন ছাত্রের প্রত্যেকের প্রাপ্ত গ্রেড বলতে হবে। শিক্ষার্থীরা কত নাম্বার পেলে তাদেরকে কি গ্রেড দেওয়া হবে তার বিস্তারিত একটা তালিকা নিচে দেওয়া আছে।

Suppose You are helping your teacher. So, your teacher asked you to make a grader. He will give you obtained marks of N students. You have to tell him what is the grade for each of the N students. The detailed grading policy is given in the following chart:

Marks Grade

-------------------------

90 - 100 A+

80 - 89 A

70 ­- 79 A­-

60 ­- 69 B+

50 ­- 59 B­-

40 ­- 49 C

35 ­- 39 D

0 ­- 34 F

ইনপুটের বর্ণনা

ইনপুটের শুরুতে একটি সংখ্যা **N ( 1<=N<=10000 )**দেয়া থাকবে যেটি গ্রেডপ্রাপ্ত ছাত্রদের সংখ্যা নির্দেশ করে। পরবর্তী **N** সংখ্যক লাইনের প্রতিটিতে একটি করে সংখ্যা **X ( 0 <= X <= 100 )**থাকবে যেটিতে প্রতিটি শিক্ষার্থীর প্রাপ্ত নম্বর লেখা থাকবে।

Input Specification

Input starts with an integer **N ( 1<=N<=10000 )**denoting the number of students. Next **N** lines will have a single integer **X ( 0 <= X <= 100 )**which is the marks obtained by that particular student.

আউটপুট এর বর্ণনা

প্রত্যেক ছাত্রের জন্য, তার ক্রমিক নম্বর এবং প্রাপ্ত গ্রেড প্রিন্ট করো।

Output Specification

For each student, print the student id and obtained grade.

Sample

| **Input** | **Output** |
| --- | --- |
| 5  90  45  80  35  34 | Student 1: A+  Student 2: C  Student 3: A  Student 4: D  Student 5: F |

#include <stdio.h>

#define MAX\_SIZE 8

int LOW[MAX\_SIZE] = {90, 80, 70, 60, 50, 40, 35, 0};

int HIGH[MAX\_SIZE] = {100, 89, 79, 69, 59, 49, 39, 34};

char GRADE[MAX\_SIZE][3]= {"A+", "A", "A-", "B+", "B-", "C", "D", "F"};

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc;

scanf("%d", &tc);

while (current\_tc < tc) {

int i;

int marks;

scanf("%d", &marks);

for (i = 0; i < MAX\_SIZE; i++) {

if (marks >= LOW[i] && marks <= HIGH[i]) {

printf("Student %d: %s\n", current\_tc + 1, GRADE[i]);

break;

}

}

current\_tc++;

}

return 0;

} // end main

ফিজবাজ এক্সট্রিম ( FizzBuzz Extreme )

CPU: 1s  
Memory: 1024MB

তুমি কি কখনো ফিজবাজের কথা শুনেছো? এটি একটি নাম করা ইন্টারভিউ সমস্যা। প্রবলেমটা এরকম:

"একটি প্রোগ্রাম লিখ যেটি ১ থেকে ১০০ পর্যন্ত সংখ্যা প্রিন্ট করবে। তবে যে সংখ্যাগুলো তিনের গুণিতক তাদের বেলায় সংখ্যাটির বদলে “Fizz” এবং পাঁচের গুণিতকের জন্য সংখ্যাটির বদলে "Buzz" প্রিন্ট করা লাগবে। যেই সংখ্যাগুলো ৩ ও ৫ উভয়ের গুণিতক, তাদের বেলায় প্রিন্ট করতে হবে, "“FizzBuzz”।

তুমি কি জানো কেন এই সমস্যাটি এত বিখ্যাত? কারণ অধিকাংশ প্রোগ্রামাররাই এই সহজ প্রোগ্রামটি লিখতে গিয়েই গুবলেট বসে। অদ্ভুত ব্যাপর, তাই না? এত সহজ প্রবলেমে ভুল?

আমরা অবশ্য তোমাকে এত সহজ প্রবলেম সলভ করতে দিব না। আমরা সমস্যাটাকে আরো মজার করব।

আমরা এই প্রবলেমটার খুব "কঠিন" একটা ভার্সন করব আজকে! আসলে এত কঠিনও না, তোমাকে যদি একটি সংখ্যা A দেওয়া থাকলে নিচের কাজগুলো করতে হবে:

১. আউটপুটে লেখ "Fizz", যদি A তিন দ্বারা বিভাজ্য, তবে পাঁচ দ্বারা বিভাজ্য না।

২. আউটপুটে লেখ "Buzz", যদি A পাঁচ দ্বারা বিভাজ্য, তবে তিন দ্বারা বিভাজ্য না।

৩. আউটপুটে লেখ "FizzBuzz", যদি A তিন ও পাঁচ উভয় দ্বারা বিভাজ্য।

৪. অন্যথায় আউটপুট "Null".

তুমি কি খেলাটি ঠিকমত খেলতে পারবে?

Do you know about the FizzBuzz game? It is a famous “Interview Problem”. The problem goes something like this:

"Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”."

Do you know why it is famous? Cause majority of the programmers seem to fail to write a program that solves the problem! Weird, right? It is so easy!

There is no way we will ask you to solve such an easy problem. We are gonna make it more interesting.

We will play an extreme version of the game. Given an integer A, you have to perform one of the following action:

Output “Fizz” if A is divisible by 3 but not 5.

Output “Buzz” if A is divisible by 5 but not 3.

Output “FizzBuzz” if A is divisible by both 3 and 5.

Output “Null” if A is not divisible by 3 or 5.

Can you play the game properly?

ইনপুট এর বর্ণনা

প্রথম লাইনে একটি ধনাত্মক পূর্ণসংখ্যা **T ( T <= 100 )** নির্দেশ করে, যেটি টেস্টকেসের সংখ্যা নির্দেশ করে। পরবর্তী **T** লাইনে একটি করে ধনাত্মক সংখ্যা **A** থাকবে। **A** সংখ্যাটিতে **105** এর থেকে কম ডিটিজ থাকবে এবং সংখ্যাগুলোর সামনে কখনো শূণ্য থাকবে না।

**৩৫ শতাংশ কেসের জন্য, A এর মান ৬৪-বিট সংখ্যার ভিতরে এঁটে যাবে।**

Input Specification

The first line contains a positive integer **T ( T <= 100 )**, number of test case. In the following **T** lines, there will be a single positive integer, **A**. **A** will have less than **105** digits and no leading zeros.

**For 35% of the cases, A will fit into 64-bit signed integer variable.**

আউটপুট এর বর্ণনা

প্রতিটি কেসের জন্য কেস নাম্বার প্রিন্ট করো, তারপর খেলাটির সঠিক উত্তর প্রদান করো। সঠিক ইনপুট/আউটপুট ফরম্যাটের জন্য উদাহরণের কেসগুলো একবার করে দেখে নাও।

Output Specification

For each case print the case number, then the proper output for the game. See sample input/output for more details..

Sample

| **Input** | **Output** |
| --- | --- |
| 4  12  25  30  7 | Case 1: Fizz  Case 2: Buzz  Case 3: FizzBuzz  Case 4: Null |

#include <stdio.h>

#include <string.h>

int charToInteger(char ch);

int sumOfDigit(char num[], int length);

int isDividedBy3(char num[], int length);

int isDividedBy5(char num[], int length);

#define TOTAL\_DIGIT 10

int num\_char[2][TOTAL\_DIGIT] = {{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, {48, 49, 50, 51, 52, 53, 54, 55, 56, 57}};

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

char num[100001];

int length;

scanf("%s", num);

length = strlen(num);

printf("Case %d: ", current\_tc + 1);

if (isDividedBy3(num, length) && isDividedBy5(num, length)) printf("FizzBuzz\n");

else if (isDividedBy3(num, length) && !isDividedBy5(num, length)) printf("Fizz\n");

else if (isDividedBy5(num, length) && !isDividedBy3(num, length)) printf("Buzz\n");

else printf("Null\n");

current\_tc++;

}

return 0;

} // end main

int charToInteger(char ch) {

int i;

int n = ch;

for (i = 0; i < TOTAL\_DIGIT; i++) {

if (num\_char[1][i] == n) return num\_char[0][i];

}

} // end charToInteger

int sumOfDigit(char num[], int length) {

int i;

int sum = 0;

for (i = 0; i < length; i++) sum += charToInteger(num[i]);

return sum;

} // end sumOfDigit

int isDividedBy3(char num[], int length) {

int sum = sumOfDigit(num, length);

if (sum % 3 == 0) return 1;

return 0;

} // end isDividedBy3

int isDividedBy5(char num[], int length) {

char last\_digit = num[length - 1];

if (last\_digit == '0' || last\_digit == '5') return 1;

return 0;

} // end isDividedBy5

মধ্যম সংখ্যা (Middle Number)

CPU: 0.3s  
Memory: 512MB

তিনটি পূর্ণসংখ্যা দেওয়া আছে, তোমাকে তাদের মধ্যে এমন সংখ্যাটি বের করতে হবে যেটি তাদের মধ্মা। অন্য ভাষায় আমরা যদি সংখ্যা তিনটিকে ছোটো থেকে বড় আকারে সাজাই, তাহলে যে সংখ্যাটি মাঝে থাকবে সেটি বের করতে হবে।

Given three integers A, B and C, you have to output the median. In other words, if we sort the numbers in ascending order, the middle number is our required output.

ইনপুটের বর্ণনা

ইনপুট এ থাকবে তিনটি করে পূর্ণসংখ্যা। A, B এবং C-এর পরমমান ২০০ থেকে বেশি হবে না। ( |A|, |B|, |C| <= 200 )।

Input Specification

You will be given three integers. The absolute value of A and B and C will not exceed 200.

আউটপুটের বর্ণনা

মধ্যম সংখ্যার মান প্রিন্ট দিতে হবে।

Output Specification

Print the median.

Sample

| **Input** | **Output** |
| --- | --- |
| 4  1 3 10  3 2 4  1 -3 -2  2 2 1 | Case 1: 3  Case 2: 3  Case 3: -2  Case 4: 2 |

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i, j;

int num[3];

scanf("%d %d %d", &num[0], &num[1], &num[2]);

int index = 3;

int temp, index\_2, minimum;

for (i = 0; i < index; i++) {

minimum = 201;

for (j = i; j < index; j++) {

if (num[j] < minimum) {

minimum = num[j];

index\_2 = j;

}

}

temp = num[i];

num[i] = minimum;

num[index\_2] = temp;

}

printf("Case %d: %d\n", current\_tc + 1, num[1]);

current\_tc++;

}

return 0;

} // end main

সংখ্যার পিরামিড (Pyramid of Numbers)

CPU: 0.5s  
Memory: 512MB

মারকাস সম্প্রতি মিশর থেকে ঘুরে আসলো। এখন সে পিরামিডের প্রেমে পরেছে। এইজন্যে সে পিরামিড আঁকবে চিন্তা করেছে। যেহেতু সে এখনো নতুন, তার আঁকা পিরামিডগুলো বাম্ পাশে ঘেঁষে থাকে। নিম্নে তার দ্বারা আঁকা একটি ৫ উচ্চতার পিরামিডের নমুনা দেয়া হল।

|  |
| --- |
| 1  12  123  1234  12345  Pyramid with  height = 5 |

এখন সে N উচ্চতার পিরামিড আঁকতে চায়। সে তোমার সাহায্য খুজছে কাজটির জন্যে। তোমাকে যদি N দেয়া হয়, অনুগ্রহ করে তার মত করে N উচ্চতার একটি পিরামিড আঁক।

Marcus recently visited Egypt. Now he has fallen in love with the pyramids. He wants to draw pyramids now. But as he is still amateur, his pyramids will be left aligned. You can see a example of pyramid with height 5 in the figure above. Now he wants to draw a pyramid with height N. He seeks your help for this. Please help him in drawing a pyramid of height N.

ইনপুটের বর্ণনা

প্রথম লাইনে থাকবে একটি ধনাত্মক পূর্ণসংখ্যা T (T <= 9 ), যেটি হল টেস্ট কেসের সংখ্যা। তার পরের T লাইনে থাকবে একটি পূর্ণ সংখ্যা N ( ১ <= N <= ৯ ), যেটি হল পিরামিডের উচ্চতা।

Input Specification

The first line contains a positive integer T (T <= 9 ), number of test case. In the following T lines, you will be given a positive integers N (1<=N <= 9) which is the height of the pyramid.

আউটপুটের বর্ণনা

প্রতিটি কেসের জন্য কেস নম্বর প্রিন্ট দিয়ে, তার পরের লাইন থেকে N উচ্চতার একটি পিরামিড আঁকতে হবে।

Output Specification

For each case print the case number, and then the pyramid with height = N.

Sample

|  |
| --- |
|  |

| **Input** | **Output** |
| --- | --- |
| 3  1  2  3 | Case 1:  1  Case 2:  1  12  Case 3:  1  12  123 |

#include <stdio.h>

//#include <math.h>

//#include <string.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i, j;

int height;

scanf("%d", &height);

printf("Case %d:\n", current\_tc + 1);

for (i = 1; i <= height; i++) {

for (j = 1; j <= i; j++) {

printf("%d", j);

}

printf("\n");

}

current\_tc++;

}

return 0;

} // end main

ব্যাটম্যানের ভয় ( Batman - Scared of Metric System )

CPU: 1s  
Memory: 1024MB

ব্যাটমান ভাবছে সে বাংলাদেশে একবার ঢুঁ মেরে যাবে। কিন্ত এখানে কিছু সমস্যা আছে। ব্যাটম্যান দুরত্ব মাপার সময় মাইল ব্যবহার করে অভ্যস্ত। কিন্ত মাইলকে কিলোমিটারে নিতে গেলেই সে হিসাবে গরমিল পাকিয়ে ফেলে।

ব্যাটম্যান ভাবল মাইলকে কিলোমিটারে পাল্টে ফেলার কাজটি সে তার "স্মার্টফোন" দিয়েই করে ফেলবে। কিন্ত ব্যস্ততার কারণে তার নিজের পক্ষে এই প্রোগ্রামটি লেখা হয়ে উঠছে না। তুমি কি একটু কষ্ট করে তাকে সাহায্যের হাত বাড়িয়ে দিতে পারো?

তোমাকে একটি বাস্তব সংখ্যা **D** দেওয়া আছে, যেটি হলো কিলোমিটারের দুরত্ব, তোমাকে এটি মাইলে রূপান্তরিত করতে হবে।

**হিন্ট:** 1 কিলোমিটার = 0.621371 মাইল।

Batman is thinking about visiting Bangladesh. But he is worried about the metric system. Since he is used to measuring distance in miles, he will have trouble adjusting to measuring in kilometers.

He decided he will use a gadget that will convert distance in kilometer to miles. But he is extremely busy and cannot find time to write the program. Can you please help him?

Given a real number **D**, which is distance in kilometer, convert it into miles.

**Hint:** 1 Kilometer = 0.621371 Mile

ইনপুটের বর্ণনা

প্রথম লাইনে একটি ধনাত্মক পূর্ণ সংখ্যা **T ( T <= 50 )** আছে, সেটি টেস্টকেসের সংখ্যা নির্দেশ করে। এরপরের **T** লাইনে, তোমাকে একটি ধনাত্মক বাস্তব সংখ্যা **D ( D <= 100,000 )** দেওয়া থাকবে, যেটির মান দশমিকের পর **দুই ঘর** পর্যন্ত হতে পারে।

Input Specification

The first line contains a positive integer **T ( T <= 50 )**, number of test case. In the following **T** lines, you will be given a positive real number **D ( D <= 100,000 )**, accurate to two decimal places.

আউটপুট এর বর্ণনা

প্রতিটা কেসের জন্য তোমার উত্তরটি প্রিন্ট করতে হবে। দশমিকের পরে **১০ ঘর** পর্যন্ত প্রিন্ট করতে হবে।

Output Specification

For each case print the distance when converted to miles. Print **10 decimal places** after your answer.

Sample

| **Input** | **Output** |
| --- | --- |
| 1  6830.16 | 4244.0633493600 |

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

double kilo, mile;

scanf("%lf", &kilo);

mile = kilo \* 0.621371;

printf("%0.10lf\n", mile);

current\_tc++;

}

return 0;

} // end main

সংখ্যার ধারা ( Sequence of Numbers )

CPU: 1s  
Memory: 1024MB

একটি **সমান্তর** ধারা হল এমন একটি ধারা যার পরপর দুইটি পদের মধ্যে পার্থক্য সমান। উদাহরণস্বরূপ, **5, 7, 9, 11, 13, 15 …** এটি একটি সমান্তর ধারা যার সাধারণ অন্তর **2**. এই ধারার পরবর্তী পদ **15 + 2 = 17**. একইভাবে, **10, 7, 4 …** এটিও একটি সমান্তর ধারা এবং এর সাধারণ অন্তর **-3**.

একটি **গুণোত্তর** ধারা হল এমন একটি ধারা যার পরপর দুইটি পদের মধ্যে অনুপাত সমান। উদাহরণস্বরূপ, **2, 6, 18, 54, 162** … এটি একটি গুণোত্তর ধারা যার সাধারণ অনুপাত **3**. এই ধারার পরবর্তী পদ **162 \* 3 = 486**.

**N** সংখ্যক পদের একটি ধারা দেওয়া থাকবে, তোমার কাজ হল ধারাটির **(N+1)** তম পদ নির্ণয় করা। প্রদত্ত ধারাটি হয় সমান্তর অথবা গুণোত্তর ধারা হবে এবং ধারাটি গুণোত্তর ধারা হলে এর সাধারণ অনুপাত একটি পূর্ণসংখ্যা হবে। **(N+1)** তম পদটি একটি integer দ্বারা প্রকাশ করা যাবে।

An **arithmetic** sequence is a sequence of numbers such that the difference between the consecutive terms is constant. For instance, the sequence **5, 7, 9, 11, 13, 15 …** is an arithmetic sequence with common difference of **2**. The next term of this sequence is **15 + 2 = 17**. Similarly, **10, 7, 4 …** is also an arithmetic sequence with common difference of **-3**.

A **geometric** sequence, on the other hand, is a sequence of numbers such that the ratio between the consecutive terms is constant. For instance, the sequence **2, 6, 18, 54, 162 …** is a geometric sequence with common ratio of **3**. The next term of this sequence is **162 \* 3 = 486**.

Given a sequence of **N** integer numbers, your task is to predict the **(N+1)th** term in the sequence. The given sequence is guaranteed to be either an arithmetic sequence or a geometric sequence. In case it is a geometric sequence, the ratio is guaranteed to be an integer number. The **(N+1)th** term is guaranteed to fit in an integer variable.

ইনপুটের বর্ণনা

ইনপুটের প্রথম লাইনে টেস্টকেসের সংখ্যা **T (1 <= T <= 100)** দেওয়া থাকবে। এর পরে **T** টি টেস্টকেসের বর্ণনা থাকবে। প্রতিটি টেস্টকেসের প্রথম লাইনে একটি পূর্ণসংখ্যা **N (3 <= N <= 1000)** দেওয়া থাকবে এবং পরবর্তী লাইনে স্পেস দিয়ে আলাদা করা **N** টি পূর্ণসংখ্যা থাকবে, যেগুলি হল ধারাটির প্রথম **N** টি পদ।

Input

The first line of the input contains the number of test cases, **T (1 <= T <= 100)**. Then T test cases follow. Each case will consist of one line containing a single integer **N (3 <= N <= 1000)**, followed by a line containing **N** space-separated integers, which are the first **N** terms of the sequence.

অাউটপুটের বর্ণনা

প্রতিটি টেস্টকেসের জন্য একটি লাইনে একটি পূর্ণসংখ্যা **X** আউটপুট দিতে হবে, যেখানে **X** হল প্রদত্ত ধারাটির **(N+1)** তম পদ।

Output

For each test case, output one line containing a single integer **X**, where **X** is the **(N+1)th** term of the given sequence.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  6  5 7 9 11 13 15  3  10 7 4  5  2 6 18 54 162 | 17  1  486 |

#include <stdio.h>

#include <math.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i;

int n;

scanf("%d", &n);

int num[n];

int nth\_elm;

for (i = 0; i < n; i++) {

scanf("%d", &num[i]);

}

if ((num[1] - num[0]) == (num[2] - num[1])) nth\_elm = num[0] + (n \* (num[1] - num[0]));

else if ((num[1] / num[0]) == (num[2] / num[1])) nth\_elm = num[0] \* (int) pow((num[1] / num[0]), n);

printf("%d\n", nth\_elm);

current\_tc++;

}

return 0;

} // end main

নামতা ( Multiplication Table )

CPU: 1s  
Memory: 1024MB

মীনা রাজুকে নামতা শিখানোর ব্যাপারে সাহায্য করছে। যখনই রাজু মীনাকে একটা সংখ্যা বলে, মীনা ধপাধপ সেই সংখ্যার ঘরের নামতা বলে দেয়।

কিন্ত কাজটা অনেক পরিশ্রমের এবং মীনার নিজেরও আগামীকাল একটা পরীক্ষা আছে। তাই মীনার তোমার সাহায্য দরকার। তুমি মীনা আর রাজুকে এমন একটি প্রোগ্রাম লিখে দাও, যেটিকে একটি সংখ্যা বললে সে সেটির নামতা বলে দিতে পারবে।

Meena is helping Raju with his multiplication table lessons. Every time Raju tells Meena a number, she makes a multiplication table of that number.

But it is a tedious work and Meena herself has an exam tomorrow. So she has asked for your help. Write a program to help Meena and Raju with this task.

ইনপুটের বর্ণনা

ইনপুটের প্রথম লাইনে কেবল একটি সংখ্যা **T ( 1 <= T <= 15 )** দেওয়া থাকবে যেটি টেস্টকেসের সংখ্যা নির্দেশ করে। পরবর্তী T লাইনের প্রতিটিতে একটি করে ধনাত্মক পূর্ণসংখ্যা **Pi** **( 1 <= Pi <= 105)**, which is the **ith** থাকে যেটি রাজুর **i** তম প্রশ্ন।

Input Specification

On the first line of input, there will be only one integer **T ( 1 <= T <= 15 )** denoting the number of test cases. Each of the next **T** lines will contain a positive integer **Pi** **( 1 <= Pi <= 105)**, which is the **ith** query of Raju.

আউটপুট এর বর্ণনা

i তম কেসের জন্য প্রথম লাইনে কেস সংখ্যা লিখো, এরপরের দশ লাইনে **Pi** সংখ্যাটির নামতা লিখে ফেলো। সঠিক ফরম্যাটের জন্য উদাহরণটা দেখে নিতে পারো।

Output Specification

For the **ith** case, output the case number on the first line. In the next ten lines, make a multiplication table for **Pi**. (See sample output for the exact output format)

Sample

| **Input** | **Output** |
| --- | --- |
| 2  22  13 | Case 1:  22 \* 1 = 22  22 \* 2 = 44  22 \* 3 = 66  22 \* 4 = 88  22 \* 5 = 110  22 \* 6 = 132  22 \* 7 = 154  22 \* 8 = 176  22 \* 9 = 198  22 \* 10 = 220  Case 2:  13 \* 1 = 13  13 \* 2 = 26  13 \* 3 = 39  13 \* 4 = 52  13 \* 5 = 65  13 \* 6 = 78  13 \* 7 = 91  13 \* 8 = 104  13 \* 9 = 117  13 \* 10 = 130 |

include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i;

int num;

scanf("%d", &num);

printf("Case %d:\n", current\_tc + 1);

for (i = 1; i < 11; i++) {

printf("%d \* %d = %d\n", num, i, num \* i);

}

current\_tc++;

}

return 0;

} // end main

তারার ছাপ - ১ ( Print the Star - 1 )

CPU: 3s  
Memory: 1024MB

এই প্রবলেম এ তোমাকে ত্রিভূজ আকতে হবে। ত্রিভূজ গুলোর আকৃতির ছবি দেয়া হল, ত্রিভূজ গুলোর নিচে সেপগুলোর সংগা দেয়া আছে। বুঝতে চেস্টা কর কোনটা কিভাবে হল,

In this problem you will have to print Triangle। The shapes are defined below, Try to find out how they are printed,

http://s27.postimg.org/frhmq4z77/abc.png

ইনপুটের বর্ননা (Input Specification)

প্রথমেই একটা নাম্বার থাকবে T যেটা বলবে তোমাকে কতগুলো টেস্ট কেস আছে। এর পর প্রত্তেক লাইন এ একটি নাম্বার এবং দুটি ক্যারেক্টার থাকবে যার ব্যাক্ষা উপরের ছবিতে দেয়া আছে।

Input will start with an integer denoting the value of number of test case T(<=100). Each of the test case will have a number n denoting the size of the triangle. Next there is two character denoting the shape of the triangle.

আউটপুটের বর্ননা (Output Specification)

কেস নাম্বার সহ তোমাকে ত্রিভূজটি প্রিন্ট করতে হবে।

Output the case number and print the triangle.

Sample

| **Input** | **Output** |
| --- | --- |
| 4  4 U R  4 U L  4 D R  4 D L | Case 1:  \*  \*\*  \*\*\*  \*\*\*\*  Case 2:  \*  \*\*  \*\*\*  \*\*\*\*  Case 3:  \*\*\*\*  \*\*\*  \*\*  \*  Case 4:  \*\*\*\*  \*\*\*  \*\*  \* |

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i, j;

int height;

char down\_or\_up, left\_or\_right;

scanf("%d %c %c", &height, &down\_or\_up, &left\_or\_right);

printf("Case %d:\n", current\_tc + 1);

if (down\_or\_up == 'U' && left\_or\_right == 'R') {

for (i = 0; i < height; i++) {

for (j = 0; j <= i; j++) {

printf("\*");

}

printf("\n");

}

}

if (down\_or\_up == 'D' && left\_or\_right == 'R') {

for (i = 0; i < height; i++) {

for (j = height; j > i; j--) {

printf("\*");

}

printf("\n");

}

}

if (down\_or\_up == 'U' && left\_or\_right == 'L') {

for (i = 0; i < height; i++) {

for (j = height - 1; j > i; j--) {

printf(" ");

}

for (j = i; j >= 0; j--) printf("\*");

printf("\n");

}

}

if (down\_or\_up == 'D' && left\_or\_right == 'L') {

for (i = 0; i < height; i++) {

for (j = i; j > 0; j--) printf(" ");

for (j = height; j > i; j--) {

printf("\*");

}

printf("\n");

}

}

current\_tc++;

}

return 0;

} // end main

প্রিফিক্স প্রিন্ট করো ( Print the Prefix! )

CPU: 1s  
Memory: 1024MB

তোমাকে কিছু স্ট্রিং দেওয়া আছে। প্রতিটা স্ট্রিংয়ের জন্য তোমাকে সেই স্ট্রিংটার সবগুলো প্রিফিক্স প্রিন্ট করতে হবে। যদি তোমাকে একটা স্ট্রিং দেওয়া হয় "YOUAREAWESOME", তাহলে এর প্রিফিক্সগুলো নিচে দেওয়া আছে।

You are given a few strings. For every string you have to print every prefix of the string. Suppose there is a string "YOUAREAWESOME", The prefixes of this string are:

Y

YO

YOU

YOUA

YOUAR

YOUARE

YOUAREA

YOUAREAW

YOUAREAWE

YOUAREAWES

YOUAREAWESO

YOUAREAWESOM

YOUAREAWESOME

ইনপুটের বর্ণনা

ইনপুট শুরু হবে একটি পূর্ণসংখ্যা T**( 1<=T<=1000 )** দিয়ে যেটি টেস্টকেসের সংখ্যা নির্দেশ করে। প্রতিটা কেসে একটি করে স্ট্রিং **Si** থাকবে। স্ট্রিং এ ২০০ টার বেশি অক্ষর থাকবে নাহ।

Input Specification

Input starts with an integer **T( 1<=T<=1000 )**, denoting the number of test cases. Each case will contain a string **Si**. Length of string will not exceed 200.

আউটপুটের বর্ণনা

প্রতিটা টেস্টকেসের জন্য, প্রথম লাইনে কেস সংখ্যা প্রিন্ট করো। এরপরের লাইন থেকে ঐ টেস্টকেসের প্রিফিক্সগুলো প্রিন্ট করা শুরু করো, প্রিফিক্স এমনভাবে প্রিন্ট করো যেন প্রিফিক্সের দৈর্ঘ্য ছোট থেকে বড় হয়। সঠিক ফরম্যাট বোঝার জন্য উদাহরণের ইনপুট আউটপুট দেখে নাও।

Output Specification

For each case, print the case number on the first line. On the next line print the prefix which length is 1, on the next line print the prefix which length is 2 and so on. See the sample input/output for more specification.

Sample

| **Input** | **Output** |
| --- | --- |
| 2  WLRBBMQBHCDARZO  KKYHIDDQSCDXRJMOWFRXSJYB | Case 1:  W  WL  WLR  WLRB  WLRBB  WLRBBM  WLRBBMQ  WLRBBMQB  WLRBBMQBH  WLRBBMQBHC  WLRBBMQBHCD  WLRBBMQBHCDA  WLRBBMQBHCDAR  WLRBBMQBHCDARZ  WLRBBMQBHCDARZO  Case 2:  K  KK  KKY  KKYH  KKYHI  KKYHID  KKYHIDD  KKYHIDDQ  KKYHIDDQS  KKYHIDDQSC  KKYHIDDQSCD  KKYHIDDQSCDX  KKYHIDDQSCDXR  KKYHIDDQSCDXRJ  KKYHIDDQSCDXRJM  KKYHIDDQSCDXRJMO  KKYHIDDQSCDXRJMOW  KKYHIDDQSCDXRJMOWF  KKYHIDDQSCDXRJMOWFR  KKYHIDDQSCDXRJMOWFRX  KKYHIDDQSCDXRJMOWFRXS  KKYHIDDQSCDXRJMOWFRXSJ  KKYHIDDQSCDXRJMOWFRXSJY  KKYHIDDQSCDXRJMOWFRXSJYB |

#include <stdio.h>

#include <string.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

int i, j;

char s[201];

int length;

scanf("%s", s);

length = strlen(s);

printf("Case %d:\n", current\_tc + 1);

for (i = 0; i < length; i++) {

for (j = 0; j <= i; j++) printf("%c", s[j]);

printf("\n");

}

current\_tc++;

}

return 0;

} // end main

পয়সা বিতরণ ( Coin Distribution )

CPU: 1s  
Memory: 1024MB

রবিন হূড একটি সফল অভিযান সম্পন্ন করে ফিরে এসেছে। এখন সে শেরউড বনের N জন অধিবাসীদের ভিতরে স্বর্ণমুদ্রাগুলো বিলি করে দিতে চাইছে। এই কাজের সুবিধার জন্য সে তার K জন বিশ্বস্ত সহযোগীকে ডেকে পাঠালো।

এই K জন বিশ্বস্ত সহযোগীর আবার খেলাধুলার শখ। তারা N জন বাসিন্দাদেরকে একটি লাইনে দাঁড়া করালো এবং তাদের প্রত্যককে একটি পূর্ণসংখ্যা দিয়ে চিহ্নিত করল। তারপর তারা তাদেরকে স্বর্ণমুদ্রাগুলো দেয়া শুরু করল। একজন সহযোগী এসে তার প্রিয় পূর্ণসংখ্যা F চয়ন করে, যেখানে [1,100] এর মধ্যে থাকবে F এর মান। এরপর সে, F এর গুণিতকতম অবস্থানের লোকদেরকে একটি করে স্বর্ণমুদ্রা দেয়।

উদাহরণস্বরূপ, যদি দশজন বাসিন্দা এবং তিনজন সাহায্যকারী থাকে, যেখানে সাহায্যকারীদের প্রিয় সংখ্যা হচ্ছে {2,3,5}। তাহলে, প্রথম সাহায্যকারিী 2, 4, 6, 8, 10 নম্বর বাসিন্দাদের একটা করে স্বর্ণমুদ্রা প্রদান করবে। দ্বিতীয় সাহায্যকারী 3,6,9 তম বাসিন্দাকে স্বর্ণমুদ্রা দেবে। এরপর তৃতীয় সাহায্যকারী এসে ৫ এবং ১০ নম্বর লোককে দেবে।

এভাবে করে মূদ্রা বিতরণের পর তারা বাসিন্দাদের আনন্দের পরিমাণ বের করে। একজন ব্যক্তি তখনি আনন্দিত হবে, যখন সে অন্তত একটি স্বর্ণমুদ্রা পাবে। আর কেউ যদি কোন মুদ্রা না পায়, তাহলে সে সুখী হবে না।

রবিনহুড জানতে চায়, শেরউডের কতজন মানুষ আসলে সুখী। যেহেতু, সাহায্যকারীরা এখন পার্টি করায় ব্যস্ত, তোমারই রবিন হুডকে সাহায্য করতে হবে।

যদি বাসিন্দাদের সংখ্যা, সাহায্যকারীদের সংখ্যা এবং তাদের প্রত্যেকের প্রিয় সংখ্যা দেওয়া থাকে, তাহলে তোমাকে বলতে হবে যে খেলা শেষে কতজন মানুষ অসুখী থাকবে।

Robin Hood has just returned from a successful heist. He is now going to distribute gold coins among the N residents of Sherwood Forest. He asked K people from his band of “Merry Men” to help him with the task.

Since the K people are “Merry”, they decided to play a game. They first made the N residents stand in a line and assigned them integer id starting from 1. Then they started to distribute the coins in turn. A helper comes, and chooses his favorite integer number F, which is between [1,100]. Then the helper gives a single coin to each person standing at a position that is multiple of F.

For example, if there are 10 residents and 3 helpers, where each of the helper has favorite numbers {2,3,5}. Then in first turn, first helper gives person number 2, 4, 6, 8, 10 a single coin. The second helper gives person number 3,6,9 one coin. Then comes third helper and he gives person number 5 and 10 one coin.

So after distributing coins in such way, they determine happiness of the residents. A resident is said to be happy, if he has received at least one coin. Meaning, a person who did not receive any coin is unhappy.

Robin Hood wants to know how many people in Sherwood is unhappy. Since the “Merry Men” are partying after the game, he wants you to help him.

Given number of residents, number of helpers and favorite number of each of the helper, how many people are unhappy once the game ends?

ইনপুটের বর্ণনা

প্রথম লাইনে একটি ধনাত্মক পূর্ণসংখ্যা T ( T <= 100 ) দেওয়া থাকবে, যেটি টেস্টকেসের সংখ্যা নির্দেশ করে। এরপরের T কেসের শুরুতে দুইটি করে ধনাত্মক পূর্ণসংখ্যা N (N <= 10^16) এবং K (K <= 10) থাকবে। পরের লাইনে K সংখ্যক পূর্ণসংখ্যা থাকবে, যেগুলো হচ্ছে সাহায্যকারীদের প্রিয় সংখ্যা। সংখ্যাগুলোর মান ১ থেকে ১০০ এর ভিতরে থাকবে।

পয়েন্টের ব্যাখ্যা

১৫ পয়েন্টের জন্য N <= 100 এবং K = 1

২৫ পয়েন্টের জন্য N <= 10^5 এবং K <= 10

২৫ পয়েন্টের জন্য N <= 10^16 এবং K = 2

৩৫ পয়েন্টের জন্য N <= 10^16 এবং K <= 10

Input Specification

The first line contains a positive integer T ( T <= 100 ), number of test case. In the following T cases, there will be two positive integers N (N <= 10^16) and K (K <= 10). On the next line, there will be K integer, denoting the favorite numbers of K helpers between [1,100].

**Point Distribution**   
Subtask 1: 15 Points for N <= 100 and K = 1.   
Subtask 2: 25 Points for N <= 10^5 and K <= 10.   
Subtask 3: 25 Points for N <= 10^16 and K = 2.   
Subtask 4: 35 Points for N <= 10^16 and K <= 10.

আউটপুটের বর্ণনা

প্রতিটা টেস্টকেসের জন্য প্রথমে কেস সংখ্যা প্রিন্ট করার পর, শেরউডের অসুখী লোকের সংখ্যা বলতে হবে।

Output Specification

For each case print the case number and then number unhappy people in Sherwood.

Sample

| **Input** | **Output** |
| --- | --- |
| 3  10 1  3  10 2  2 3  10 3  2 3 5 | Case 1: 7  Case 2: 3  Case 3: 2 |

**Explanation**

In first case, there are 10 people in Sherwood. Person number {3,6,9} gets coin. So 7 people are unhappy.

In second case, out of 10 people {2,3,4,6,8,9,10} gets coin. {1,5,7} has 0 coins. So they are sad.

In third case, {2,3,4,5,6,8,9,10} gets coin. Only {1,7} are unhappy.

#include <stdio.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc = 0;

scanf("%d", &tc);

while (current\_tc < tc) {

current\_tc++;

int i, j;

long long int total\_people; // N

int total\_fav; // K

scanf("%lld %d", &total\_people, &total\_fav);

int fav\_num[total\_fav];

long long int coins[total\_people];

long long int sum;

long long int count\_unhappy = 0;

for (i = 0; i < total\_people; i++) {

coins[i] = 0;

}

for (i = 0; i < total\_fav; i++) {

scanf("%d", &fav\_num[i]);

}

for (i = 0; i < total\_fav; i++) {

sum = fav\_num[i];

for (j = 0; sum <= total\_people; j++) {

coins[sum - 1] = 1;

sum += fav\_num[i];

}

}

for (i = 0; i < total\_people; i++) {

if (coins[i] == 0) count\_unhappy++;

}

printf("Case %d: %lld\n", current\_tc, count\_unhappy);

}

return 0;

} // end main

বাঘা যোদ্ধা হাসমত ( Hashmat The Brave Warrior )

CPU: 1s  
Memory: 1024MB

বাঘা যোদ্ধা হাসমত, এখানে ওখানে ঘুরে বেরায় আর শত্রুর সাথে যুদ্ধ করে। যুদ্ধ শুরুর আগে সে শুধু নিজের এবং প্রতিপক্ষের যোদ্ধার সংখ্যার পার্থক্যই বের করে এই পার্থক্য দেখে সে যুদ্ধ করা বা না করার সিদ্ধান্ত গ্রহণ করে। হাসমত এর যোদ্ধার সংখ্যা কখনোই প্রতিপক্ষ থেকে বেশি হয় না।

Hashmat is a brave warrior who with his group of young soldiers moves from one place to another to fight against his opponents. Before fighting he just calculates one thing, the difference between his soldier number and the opponent's soldier number. From this difference he decides whether to fight or not. Hashmat's soldier number is never greater than his opponent.

ইনপুটের বর্ণনা

ইনপুট ফাইল এর প্রতি লাইন এ **দুটি** সংখ্যা থাকে। এই দুটি সংখ্যা যথাক্রমে হাসমত ও তার প্রতিপক্ষের যোদ্ধার সংখ্যা অথবা তদ্বিপরীত। ইনপুটের কোনো সংখ্যাই **২৬৪ থেকে বড় নয়**। **EOF** দিয়ে ইনপুটের শেষ হয়।

Input Specification

The input file contains **two** integer numbers in every line. These two numbers in each line denotes the number of soldiers in Hashmat's army and his opponent's army or vice versa. The input numbers are not greater than **2^64**. Input is terminated by **End of File**.

আউটপুট এর বর্ণনা

প্রতি লাইন ইনপুটের জন্য হাসমত এবং তার প্রতিপক্ষের যোদ্ধার সংখ্যার পার্থক্য মুদ্রণ করতে হবে। প্রতিটি লাইনের জন্য আউটপুট আলাদা আলাদা লাইনেই দিতে হবে।

Output Specification

For each line of input, print the difference of number of soldiers between Hashmat's army and his opponent's army. Each output should be in separate line.

Sample

| **Input** | **Output** |
| --- | --- |
| 10 12  10 14  100 200 | 2  4  100 |

জাজ ডাটা অনেক বড়। তাই দ্রুত ইনপুট/আউটপুট ব্যবহার করিও।  
Judge is huge. Use faster input/output.

include <stdio.h>

long long int absolute(long long int num) {

if (num < 0) return -num;

return num;

}

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

long long int a, b;

while (scanf("%lld %lld", &a, &b) == 2) {

printf("%lld\n", absolute(a - b));

}

return 0;

} // end main

সংখ্যা সাজাও (Sort the Numbers)

CPU: 0.2s  
Memory: 512MB

তিনটি পূর্ণ সংখ্যা দেওয়া আছে, তোমাকে তাদেরকে বড় থেকে ছোটো আকারে সাজাতে হবে।

Given three integers, you need to sort them in descending order.

ইনপুটের বর্ণনা

ইনপুটে থাকবে তিনটি করে পূর্ণসংখ্যা A, B এবং C। A, B এবং C –এর পরমমান ২০০ থেকে বেশি হবে না । ( |A|,|B|,|C| <= 200 )

Input Specification

You will be given three integers A, B and C. The absolute value of A, B and C will not exceed 200.

আউটপুটের বর্ণনা

আউটপুটে তিনটি নম্বরকে বড় থেকে ছোটো আকারে সাজিয়ে লিখতে হবে।

Output Specification

You need to write the three integers in descending order in the output.

Sample

| **Input** | **Output** |
| --- | --- |
| 1 2 3  2 1 3 | 3 2 1  3 2 1 |

#include <stdio.h>

//#include <math.h>

//#include <string.h>

int main()

{

//freopen("/sdcard/coding/anwar/oj/in.txt", "rt", stdin);

//freopen("/sdcard/coding/anwar/oj/out.txt", "wt", stdout);

int tc;

int current\_tc;

//scanf("%d", &tc);

tc = 1;

for (current\_tc = 1; current\_tc <= tc; current\_tc++) {

int i, j, k;

int index;

int temp;

index = 3;

//scanf("%d", &index);

int num[index];

for (i = 0; i < index; i++) scanf("%d", &num[i]);

// now sort DESC

for (i = 0; i < index; i++) {

for (j = i; j < index; j++) {

if (num[j] > num[i]) {

temp = num[i];

num[i] = num[j];

num[j] = temp;

}

}

}

// now print them

printf("%d", num[0]);

for (i = 1; i < index; i++) {

printf(" %d", num[i]);

}

printf("\n");

}

return 0;

} // end main

……………………………………………………………………………………………………………………………………………………………………………