Swapping data members Given two integers u are requested to swap the contents of two variables. Constraints number accepted n >MIN\_INT and n<MAX\_INT Sample Input 10 20 Sample Output 20 10

#include <math.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <assert.h>

#include <limits.h>

#include <stdbool.h>

int main() {

int a,b;

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

printf("%d %d",b,a);

return 0;

}

A Pythagorean triplet is a set of three integers a, b and c such that a2 + b2 = c2.

You have been given the task to accept three integers and check whether they are Pythagorean triplets or not.

Sample Input

1 2 3

Sample Output

no

Sample Input

3 4 5

Sample Output

yes

#include <math.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <assert.h>

#include <limits.h>

#include <stdbool.h>

int main() {

int a,b,c;

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

if(c\*c==a\*a+b\*b) printf(“yes");

else printf("no");

return 0;

}

Count Divisors

You have been given 3 integers l, r and k

Find how many numbers between l and r (both inclusive) are divisible by k.

You do not need to print these numbers, you just have to find their count.

Input Format

The first and only line of input contains 3 space separated integers

l, r and k

Output Format

Print the required answer on a single line.

1<=l<=r<=1000

1<=k<=1000

Sample Input

3 300 3

Sample Output

100

int main() {

int l,r,k;

scanf("%d %d %d",&l,&r,&k);

printf("%d",r/k);

return 0;

}

Multiples of 3 and 5

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below N.

Input Format

First line contains number of test cases

This is followed by T lines ,each containing an integer N

Constraints

1<=T<=105

1<=N<=109

Output Format

For each test case, print an integer that denotes the sum of all the multiples of 3 or 5 below N.

Sample Input

2

10

100

Sample output

23

2318

int main() {

int n,i,a,c=0;

scanf("%d",&n);

while(n>0){

scanf("%d",&a);

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

if(i%3==0||i%5==0)c=c+i;

}

printf("%d\n",c);

n--;

c=0;

}

return 0;

}

Number reverse

U have been given the task to accept a number and reverse it

Sample Input

123

Sample Output

321

Sample Input

-123

Sample Output

-321

Constraints

INT\_MIN<number<INT\_MAX

int main() {

long long int lock=0,n,sum=0;

scanf("%lld",&n);

if(n<0){

n=-n;

lock=1;

}

while(n>0){

sum=sum\*10+n%10;

n=n/10;

}

if(lock==1){sum=-sum;

lock=0;}

printf("%lld",sum);

return 0;

}

Sum Of Numbers

U he been given the task to accept the numbers which are in the range 0 to 100, till it encounters -1. Print the sum of all the integers that you have read before you encountered -1

Input

A sequence of integers separated by whitespace. There may be other integers following -1.

Output

Sum of all integers in the sequence before you encounter -1. Any integer that is input after -1 in the sequence should be ignored.

Sample Input

-1

Sample Output

0

Sample Input

1 2 3 -1

Sample output

6

int main() {

int a,sum=0;

scanf("%d",&a);

while(a!=-1){

sum=sum+a;

scanf("%d",&a);

}

printf("%d",sum);

return 0;

}

Digital Root

The digital root (also called repeated digital sum) of a number is a single digit value obtained by an iterative process of

summing digits.

Digital sum of 65536 is 7, because 6+5+5+3+6=25 and 2+5 = 7.

Write a program that takes an integer as input and prints its digital root.

INPUT:

A single integer N

OUTPUT:

Digital root of the number N.

Constraints

1<=n<107

int main() {

long long int a,b,sum=0,c=0;

scanf("%lld",&a);

while(a>0){

b=a%10;

a=a/10;

sum=sum+b;

}

b=0;

while(sum>0){

b=sum%10;

sum=sum/10;

c=c+b;

}

printf("%lld",c);

return 0;

}

Points on a Straight Line

Given three points (x1, y1), (x2, y2) and (x3, y3), write a program to check if all the three points fall on one straight line.

INPUT:

Six integers x1, y1, x2, y2, x3, y3 separated by whitespace.

OUTPUT:

Print “Yes” if all the points fall on straight line, “No” otherwise.

CONSTRAINTS:

-1000 <= x1, y1, x2, y2, x3, y3 <= 1000

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

float x1, y1, x2, y2, x3, y3,m1,m2,m3;

scanf("%f %f %f %f %f %f",&x1,&y1,&x2,&y2,&x3,&y3);

if((x2-x1)!=0)m1=(y2-y1)/(x2-x1);

if((x3-x2)!=0)m2=(y3-y2)/(x3-x2);

if(m1==m2)printf("Yes");

else

printf("No");

return 0;

}

Decimal To Binary

You have been given an integer decimal number.Convert that number to binary number.

Sample input

7

Sample output

111

Sample input

16

Sample output

10000

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int decimal,lock=0,a=1;

scanf("%d",&decimal);

long long int binary=0;

if(decimal<0){

decimal=-decimal;

lock=1;

}

while(decimal>0){

if(lock==1){

binary=1;

lock=0;

}

binary =binary+((decimal%2)\*a);

decimal=decimal/2;

a=a\*10;

}

printf("%lld",binary);

return 0;

}

Power Number

Given two positive numbers x and y, check if y is a power of x or not.

Sample Input

10 1001

Sample output

False

Explanation:In this sample input x=10 and y=1001. There is no power of x which can make it 1001.So output is False.

Sample input

10 1000

Sample output

True

Explanation: In sample input x=10 and y=1000.10^3 is equivalent to 1000.So output is True.

int main() {

long long int a,c,b,d,e,lock=0,i=1,j,f;

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

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

for(j=0;j<=10;j++){

f=pow(a,j);

if(f==b){lock=1;break;}

}

/\* d=pow(a,2);

c=pow(a,3);

e=pow(a,4);

\*/

/\*if(a==b||b==c||e==b||a==b)lock=1;\*/

if(lock==1)printf("True");

else printf("False");

return 0;

}

Fibonacci Variation

A series is defined in the following manner:

Given the nth and (n+1)th terms,

the (n+2)th can be computed by the following relation

Tn+2 = (Tn+1)2 + Tn

So, if the first two terms of the series are 0 and 1:

the third term = 12 + 0 = 1

fourth term = 12 + 1 = 2

fifth term = 22 + 1 = 5

Sample Input:

2

5

6

Sample Output:

5

27

Explanation:

In sample input first line of input stands for number of testcases.

The first two terms of the series are 0 and 1.

The fifth term is 5.

The sixth term is 27.

int main() {

int x,ans=0,f,s,t,n,i;

scanf("%d",&x);

while(x>0){

scanf("%d",&n);

if(n==1)t=0;

if(n==2)t=1;

if(n>2){

f=0;s=1;

for(i=3;i<=n;i++){

t=s\*s+f;

f=s;

s=t;

}

}

printf("%d\n",t);

x--;t=0;n=0;

}

return 0;

}

PerFectSquare

Write a program to find the number of perfect squares between given two numbers A and B (both inclusive).

A number is called a perfect square if it can be written as x\*x for some integer x.

Sample Input

Two numbers A and B separated by a space

Output

Count of the number of perfect squares

Sample Input

3 10

Sample Output

2

(Explanation 2\*2, 3\*3)

int main() {

int a,b,c=0,i,j;

float x;

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

for(i=0;i<=b/2;i++){

x=i\*i;

for(j=a;j<=b;j++)if(j==x)c++;

}

printf("%d",c);

return 0;

}

ProductOfDigits

Write a C program that takes a positive number N and produces an output that is the product of its digits.

sample Input

657

Sample output

210

(6\*5\*7)

Constraints

1<=N<=999999

int main() {

int x,z=1;

scanf("%d",&x);

while(x>0){

z=z\*(x%10);

x=x/10;

}

printf("%d",z);

return 0;

}

Counting Digits

Given an integer,traverse its digits(d1,d2,d3,...dn) and determine how many digits evenly divide N(count the number of times

N is divided by each digit has a remainder of 0).Print the number of evenly divisible digits

Sample Input

24

Sample Output

2

Explanation If N = 24, there are 2 digits - 2 & 4.

Both these digits exactly divide 24.

So our answer is 2.

Sample Input

122

Sample Output

3

Explanation:Duplicate numbers should also be counted, For N=122, 2 divides 122 exactly and occurs at ones' and tens'

position.

So it should be counted twice.

So, answer is 3.

int main() {

int x,y,a,b,c=0;

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

scanf("%d",&x);

a=x;

while(x>0){

y=x%10;

b=a%y;

if(b==0)c++;

x=x/10;

}

printf("%d",c);

return 0;

}

Proportion Check

U have given the task to accept four integers a b c and d.

Formally, four numbers x, y, z, w are said to make a proportion if ratio of a : b is same as that of c : d.

U r task is to check whether you can shuffle these numbers so as to make some proportion?

four numbers a, b, c, d are said to make a proportion if ratio of a : b is same as that of c : d.

Constraints

1 <= a, b, c, d <=1000

Input:

1 2 4 2

Output:

Possible

Explanation

By swapping 4 and the second 2, we get 1 2 2 4.

Note that 1 2 2 4 make proportion as 1 : 2 = 2 : 4.

Hence answer is "Possible"

Input

3 6 9 12

Output

Impossible

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

float a,b,c,d,lock=0;

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

if(a/b==c/d)lock=1;

else if(b/a==c/d)lock=1;

else if(a/b==d/c)lock=1;

else if(b/a==c/d)lock=1;

if(lock==1)printf("Possible");

else printf("Impossible");

return 0;

}

Product of Divisors of a number

Q1) Being in love with number theory, Johnny decided to take a number theory course. On the first day, he was challenged by his teacher with the following problem:

given a number N, compute the result X which is the product of its positive divisors. Johnny is desperate to impress his new teacher so he asks you for help.

In this problem, the divisors of N do not include the number N itself.

Constraint:

N>=1 and N<=1000

Sample Test Case

Input:

12

Output:

144

In the above test case as N=12, the divisors of N

(excluding N) are 1, 2, 3, 4, and 6. Thus, the product of divisors is 1x2x3x4x6=144

int main() {

long long int x=1,a,i;

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

scanf("%lld",&a);

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

if(a%i==0)x=x\*i;

}

printf("%lld",x);

return 0;

}

Palindrome Sum

A number is called palindromic if its decimal representation is a palindrome. You are given a range, described by a pair of integers L and R.

Find the sum of all palindromic numbers lying in the range [L, R], inclusive of both the extrema.

Input

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.

The first line of each test case contains a pair of space separated integers L and R denoting the range for which you are required to find the sum of the palindromic numbers.

Output

For each test case, output a single line containing the sum of all the palindromic numbers in the given range.

Constraints

1 ≤ T ≤ 100

Subtask 1 : 1 ≤ L ≤ R ≤ 103

Subtask 2 : 1 ≤ L ≤ R ≤ 105

Sample Input

2

1 10

123 150

Sample Output

45

272

int main() {

long long int rev=0,a,i,b,sum,x,n;

scanf("%lld",&n);

while(n>0){

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

for(i=a;i<=b;i++){

x=i;rev=0;

while(x>0){

rev=rev\*10+(x%10);

x=x/10;

}

if(rev==i)sum=sum+i;

}

n--;

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

sum=0;

}

return 0;

}

Factorial Digit Counting

Given an integer n (can be very large), find the number of digits that appear in its factorial.

Input Format

Input will be an integer n

constraints

n<=15

Output Format

Number of digits in factorial of n

Sample Input

1

Sample output

1

Sample Input

5

Sample output

3

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

long long int fact=1,i,n,c=0;

fact = 1;

scanf("%lld" , &n);

for(i = 1; i <= n; i++)

{

fact = fact\*i;

}

while(fact>0){

c++;

fact=fact/10;

}

printf("%lld",c);

return 0;

}

Factorial Of A Number

Write a program that will read an integer N from STDIN and compute the factorial of N. The input to your program will contain many lines. On each line there will be exactly one integer. Your program should print the factorial of each integer in a separate line. Your program should terminate when it reads a number that is not positive.

1 >= N <= 15

Sample Input

1

2

3

-1

Sample Output

1

2

6

int main() {

long long int n=0,fact=1;

scanf("%lld",&n);

while(n>=0){

while(n>0){

fact=fact\*n;

n--;

}

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

fact=1;

scanf("%lld",&n);

}

return 0;

}

Sherlock And Squares

Watson gives two integers A & B to Sherlock and asks if he can count the number of square integers between A and B (both inclusive).

A square integer is an integer which is the square of any integer. For example, 1, 4, 9, 16 are some of the square integers as they are squares of 1, 2, 3, 4 respectively.

Input Format

First line contains T, the number of testcases. T test cases follow, each in a newline. Each testcase contains two space separated integers denoting A and B.

Output Format

For each testcase, print the required answer in a new line.

Constraints

1 ≤ T ≤ 100

1 ≤ A ≤ B ≤ 109

Sample Input

2

3 9

17 24

Sample output

2

0

Explanation

In the first testcase, 4 and 9 are the square numbers.

In second testcase no square numbers exist between 17 and 24 (both inclusive).

int main() {

int a,b,c=0,i,j,n;

float x;

scanf("%d",&n);

while(n>0){

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

for(i=0;i<=b/2;i++){

x=i\*i;

for(j=a;j<=b;j++)if(j==x)c++;

}

printf("%d\n",c);

n--;

c=0;

}

return 0;

}

Smallest Currency Numbers of Notes

Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100.

If the sum of Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.

Input

The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.

Output

Display the smallest number of notes that will combine to give N.

Constraints

1 ≤ T ≤ 1000

1 ≤ N ≤ 1000000

Example

Input

3

1200

500

242

Output

12

5

7

int main() {

int a,note100,note50,note10,note5,note2,note1,sum=0,i,r;

scanf("%d",&a);

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

scanf("%d",&r);

note100=r/100;

r=r%100;

note50=r/50;

r=r%50;

note10=r/10;

r=r%10;

note5=r/5;

r=r%5;

note2=r/2;

r=r%2;

note1=r;

sum=note50+note100+note10+note5+note2+note1;

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

sum=0;

}

return 0;

}