**Task 14 : Swap Two Numbers**

In C, you can swap the values of two variables by using a third temporary variable.

The following is the logical map of doing so :  
1. Store value of variable1 in temporary variable.  
2. Store value of variable2 in variable1.  
3. Store value of temporary variable in variable2.

Complete the code below so that it swaps the values of var1 and var2;

**NOTE :** *You are expected to fix any other errors that you come across*.

**Sample Output 0**

5 10

#include <stdio.h>

int main()

{

int var1 = 10, var2 = 5;

int temp;

temp=var1;

var1=var2;

var2=temp;

printf("%d %d", var1, var2);

return 0;

## Task 2.1 : Shorthand Assignment Operator

To increase the value of a variable var by 1 or 2, we used var = var + 1 or var = var + 2.

C has a shortcut syntax for carrying out these tasks and they are known as shorthand operators. Their use is : var += 1 or var += 2. Infact any expression var = var + x can be shrinked to var += x;

This works also with other operators :  
- var = var - x; -> var -= x;  
- var = var \* x; -> var \*= x;  
- var = var / x; -> var /= x;  
- var = var % x; ->var %= x

To complete this task, input two numbers A and B, and print the first three terms of the series A + B, A + 2B, A + 3B;

**Sample Input 0**

5 3

**Sample Output 0**

8 11 14

#include <stdio.h>

int main()

{

int A, B;

int l,j,k;

//Increment A by B and print the result. Repeat three times.

scanf("%d%d",&A,&B);

j=A+B;

l=A+2\*B;

k=A+3\*B;

printf("%d %d %d",j, l, k);

return 0;

}

## A 3 : Task 2.13 : Basic Bitwise Operators

Bitwise operators are similar to the operators used in boolean algebra.

Bitwise AND : & : returns 1 if both bits are 1

Bitwise OR : | : returns 1 if atleast one bit is 1

Bitwise XOR : ^ : returns 1 if only one bit is 0 and the other is 1.

These operators operates on the variables bits.

For example : 6 in binary is 110 while 10 in binary is 1010.

Hence

6 & 10 = (0110) & (1010) = (0010) = 2 . Example : var1 & var2

6 | 10 = (0110) | (1010) = (1110) = 14. Example : var1 | var2

6 ^ 10 = (0110) ^ (1010) = (1100) = 12. Example : var1 ^ var2

For this task, write a program to input two numbers and print their bitwise AND, OR and XOR one on each line.

**Sample Input 0**

6 10

**Sample Output 0**

2

14

#include <stdio.h>

# int main()

# {

# int var1, var2, ans\_and, ans\_or, ans\_xor;

# //Take as input var1 and var2 and print the result of their bitwise operations

# scanf("%d%d",&var1,&var2);

# ans\_and=var1 & var2;

# ans\_or=var1 | var2;

# ans\_xor=var1 ^ var2;

# printf("%d\n%d\n%d\n",ans\_and,ans\_or,ans\_xor);

# 

# return 0;

# }

## Task 2.14 : The Bitwise Shift Operators

The left shift operator << shift the bits towards the left while the right shift operator >> shifts the bits towards the right.

Example :

5 = 101 after left shift becomes : (5<<1) = (1010) = 10

5 = 101 after 2 left shifts becomes : (5<<2) = (10100) = 20.

5 = 101 after 1 right shift becomes : (5>>1) = (010) = 2

You can notice that one left shift multiplies the number by 2. Similarly one right shift will divide the number by 2.

Syntax :

(var << no\_of\_shifts) or (var >> no\_of\_shifts)

To complete this task, write a code to input a number, and to print the number divided by 2 and the number multiplied by 2 using bitwise shift operator.

**Sample Input 0**

5

**Sample Output 0**

2 10

# #include <stdio.h>

# int main()

# {

# int var, multiplied\_by\_2, divided\_by\_2;

# scanf("%d", &var);

# //assign values to divided and multiplied using bitwise shift operators

# 

# divided\_by\_2=var>>1;

# multiplied\_by\_2=var<<1;

# 

# printf("%d ", divided\_by\_2);

# printf("%d", multiplied\_by\_2);

# return 0;

# }

## Task 2.15 : The SizeOf Operator

C has an operator named sizeof that returns the number of bytes of memory that its argument holds.

For example, sizeof(int) will return 4 while sizeof(long long int) will return 8. You can also pass a variable as an argument.

So, int var; sizeof(var) will also return 4 as var is an integer that occupies 4 bytes.

For this task print the sizes of int,float,long long int, double and char in increasing order.

**NOTE :** You can check the ***custom input*** option to check your output

#include <stdio.h>

int main()

{

//s

int int\_size = sizeof(int);

int f=sizeof(float);

int k=sizeof(long long int);

int d=sizeof(double);

int c=sizeof(char);

printf("%d %d %d %d %d",c,int\_size,f,k,d);

return 0;

}

## A 5 : Task 2.16 : Precedence

So far we've worked with individual operators, but how do they behave when used together?

Every operator has its own priority or precedence. An operator with higher precedence will be applied before an operator with a lower priority. Precedence is the BODMAS of computing in C. Associativity indicates given two operators of same precedence, in what manner will they be applied i.e. from left to right or from right to left.

Here are the precedence and associativity tables : [http://www.difranco.net/compsci/C\_Operator\_Precedence\_Table.htm](https://www.hackerrank.com/external_redirect?to=http://www.difranco.net/compsci/C_Operator_Precedence_Table.htm)

Below is the code to take as input four numbers : A, B, C and D and print the value of (A + B) \* ( C+ D), however it was written by someone who did not understand operator precedence so it gives the wrong answer. Your task is to fix the code so that it prints the correct answer.

**Sample Input 0**

1 2 3 4

**Sample Output 0**

21

#include <stdio.h>

int main()

{

int A, B, C, D, ans;

scanf("%d%d%d%d", &A, &B, &C, &D);

//change the below expresssion so it does the addition first and then the multiplication

//hint : use parenthesis

ans = (A + B) \* (C + D);

printf("%d", ans);

return 0;

}

## Math or Logic ?

Little Kunya just started going to school. Today at school she was taught how to add two numbers. When she came back she opened the Math for kids magazine to solve the problem of the day only to find out that it asked three numbers to be added instead of two. Little Kunya only knows how to add two numbers. Can you help her solve the problem by adding three numbers?

**Input Format**

One line of input with three integers A, B and C.

**Constraints**

1 <= A, B, C <= 10^9

**Output Format**

One number denoting the sum of three given integers.

**Sample Input 0**

10 20 30

**Sample Output 0**

60

**Explanation 0**

10 + 20 + 30 = 60 so we print 60.

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

long long int num1,num2,num3;

scanf("%lld%lld%lld", &num1,&num2,&num3);

printf("%lld",(num1+num2+num3));

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

return (0);

}

## A D01 - Cut the Fruit

One hot summer day Aarti and his friend Anurag decided to buy a watermelon. They chose the biggest and the ripest one, in their opinion. After that the watermelon was weighed, and the scales showed w kilos. They rushed home, dying of thirst, and decided to divide the berry, however they faced a hard problem.

Aarti and Anurag are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. They are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the watermelon in the way they want. For sure, each of them should get a part of positive weight.

**Input Format**

The first (and the only) input line contains integer number w — the weight of the watermelon bought by the boys.

**Constraints**

1 ≤ w ≤ 100

**Output Format**

Print YES, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and NO in the opposite case.

**Sample Input 0**

8

**Sample Output 0**

YES

**Explanation 0**

The boys can divide the watermelon into two parts of 2 and 6 kilos respectively (another variant — two parts of 4 and 4 kilos)

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a;

cin>>a;

if(a!=2&&a%2==0)

cout<<"YES";

else cout<<"NO";

return 0;

}

## Z 412 The Slopes of Line Segments

In the town of line segments two line segments can only become friends if their slopes are equal. Line segments are not smart enough to calculate their own or some other line segment's slope so they use a machine called the slopeFinder to check their compatibility. Recently someone stole the slopeFinder and now the line segments are upset because they cannot make new friends. The Mayor of the town has hired you to write a code to fix the crisis that their town is facing.

**Input Format**

Input Contains two line segments each on a line of its own. Each line segment is denoted by four integers Xa, Ya, Xb and Yb where (Xa,Ya) and (Xb, Yb) denote the two end points of the line segment.

**Constraints**

0 <= |Xa|,|Xb|,|Ya|,|Yb| <= 100

**Output Format**

Output "yes" if both the line segments have the same slope and "no" otherwise. (without the quotes).

**Sample Input 0**

0 0 1 1

1 0 2 1

**Sample Output 0**

yes

**Sample Input 1**

0 0 1 1

2 1 3 0

**Sample Output 1**

no

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,b,c,d,m1,m2,i=0;

while(i<2)

{

cin>>a>>b>>c>>d;

if(i==0&&(c-a)!=0)

m1=(d-b)/(c-a);

if(i==1&&(c-a)!=0)

m2=(d-b)/(c-a);i++;}

if(m1==m2)

cout<<"yes";

else

cout<<"no";

return 0;

}

## Z 433 Parliament Square

Parliament in the capital city of Delhi has a rectangular shape with the size n × m meters. On the occasion of the city's anniversary, a decision was taken to pave the Square with square granite flagstones. Each flagstone is of the size a × a.

What is the least number of flagstones needed to pave the Square? It's allowed to cover the surface larger than the Parliament Square, but the Square has to be covered. It's not allowed to break the flagstones. The sides of flagstones should be parallel to the sides of the Square.

**Input Format**

The input contains three positive integer numbers in the first line: n,  m and a.

**Constraints**

1 ≤  n, m, a ≤ 10^9

**Output Format**

Write the needed number of flagstones.

**Sample Input 0**

6 6 4

**Sample Output 0**

4

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

long int a,b,c,d,e,m,n;

cin>>n>>m>>a;

while(m%a!=0)

m++;

while(n%a!=0)

n++;

c=n/a;

d=m/a;

e=c\*d;

cout<<e;

return 0;

}

## Domino Piles

You are given a rectangular board of M × N squares. Also you are given an unlimited number of standard domino pieces of 2 × 1 squares. You are allowed to rotate the pieces. You are asked to place as many dominoes as possible on the board so as to meet the following conditions:

1.Each domino completely covers two squares.

2.No two dominoes overlap.

3.Each domino lies entirely inside the board. It is allowed to touch the edges of the board.

Find the maximum number of dominoes, which can be placed under these restrictions.

**Input Format**

In a single line you are given two integers M and N — board sizes in squares

**Constraints**

1 ≤ M ≤ N ≤ 16

**Output Format**

Output one number — the maximal number of dominoes, which can be placed.

**Sample Input 0**

2 4

**Sample Output 0**

4

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,b;

cin>>a>>b;

int k=(a\*b)/2;

cout<<k;

return 0;

}

## Z 434 Into the water

Aditi takes place in a bike race. The race takes place along the roads at the shores of the Hussain Sagar Lake. As everyone knows, the boundaries of the Hussain Sagar Lake consists only of straight lines that are either horizontal with respect to the equator or vertical as seen from the space.

Assume a 2D co-ordinate system with the hussain sagar lake being a polygon in it with all its side either parallel to the X-axis or the Y-axis. The starting point of the race is the southernmost position of the boundary. Since all sections are straight, Aditi is always travelling in one of the four directions : East, West, North or South. Every turn is a 90 degree turn i.e. every two consecutive roads will be perpendicular to each other.

Aditi is still naive so she is not very confident at some turns. Name, Aditi feels insecure at a turn there is a possibility that she can fall into the Hussain Sagar Lake. In formal words, Aditi considers a turn dangerous if she can fall directly into the water upon ignoring the turn.

Help Aditi get ready for the competition -- determine the number of dangerous turns.

**Input Format**

The first line contains integer N, the number of straight sections of the boundary.  
Next N+1 lines contains pairs of integers (xi, yi). The first of these points is the starting position. The ith straight section begins at point (xi, yi) and ends at (xi+1, yi+1).  
It is guaranteed that :

* The first section is directed towards north.
* The southernmost point is the starting point.
* The last point coincides with the first point.
* Any pairs of straight lines share no points except the neighbouring tracks that share one end point.
* No pair of points except the first and the last is same.

**Constraints**

4 <= n <= 100  
-10000 <= xi,yi <= 10000

**Output Format**

Print a single integer — the number of dangerous turns on the track.

**Sample Input 0**

6

0 0

0 1

1 1

1 2

2 2

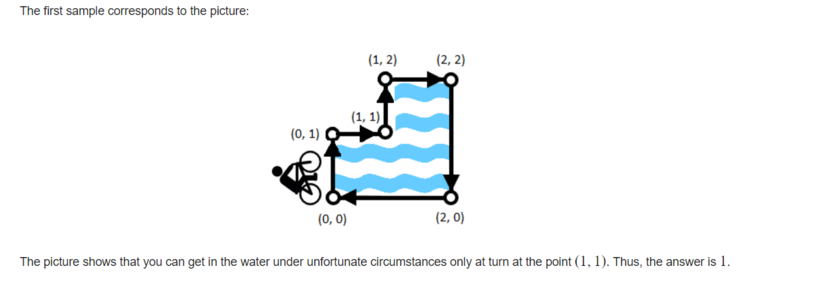
2 0

0 0

**Sample Output 0**

1

**Explanation 0**



#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,b,c;

cin>>a;

while(a<0)

{

cin>>b>>c;

a--;

}

int k=(a-4)/2;

cout<<k;

return 0;

}

## C D01 - Prime Testing - 1

You are given one number N. You must write a program to test whether the given number is prime or not. NOTE : A prime number is such a number that has only two factors : i.e. 1 and itself. 1 is not a prime number.

**Input Format**

First line contains one number, N.

**Constraints**

1 <= N <= 10^5

**Output Format**

Output "yes" if N is a prime number and "no" otherwise. (without the quotes)

**Sample Input 0**

5

**Sample Output 0**

yes

**Sample Input 1**

12

**Sample Output 1**

no

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,i,k=0;;

cin>>a;

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

{

if(a%i==0) k++;

}

if(k==2)

cout<<"yes";

else cout<<"no";

return 0;

}

## Task 3.4 : Odd or Even

Your task is simple, input 5 numbers and for each one of them on a new line print "odd" if they are odd or "even" if they are even.

**Sample Input 0**

5 2 10 7 12

**Sample Output 0**

odd

even

even

odd

even

#include <stdio.h>

int main()

{

int var, input\_count = 0;

//write your code here

while(input\_count<5){

scanf("%d",&var);

if(var%2==0)

printf("even \n");

else

printf("odd\n");

input\_count++;

}

return 0;

}

## Task 3.5 : Prime or Not

Write a program that takes one number as an input and prints "yes" if that number is prime and "no" otherwise.

**Sample Input 0**

37

**Sample Output 0**

yes

**Sample Input 1**

22

**Sample Output 1**

no

#include <stdio.h>

int main()

{

int n,z=0; scanf("%d",&n); for(int i=1;i<=n;++i) {if(n%i==0) z++;} if(z==2) printf("yes"); else printf("no");

return 0;

}

## Z 99 Task 3.8 : Multiplication Table

Execution of the loop body once is known as one iteration. So if the loop body executes 10 times, the loop is said to have run 10 iterations.

Write a program to input a number N and print the first 10 multiples of N.

**Sample Input 0**

6

**Sample Output 0**

6 12 18 24 30 36 42 48 54 60

#include <stdio.h>

int main()

{

int i, N;

scanf("%d", &N);

//write a loop that runs for 10 iterations

for(i=1;i<=10;i++)

printf("%d ",N\*i);

return 0;

}

## Task 3.9 : Goto and Label

The label keyword is used to mark a statement in C. A label can be any plain text except a C keyword. The goto keyword is used to jump to any label. For example :

int n = 1;

while (n <= 10){

if (n % 3 == 0) goto label1;

printf("%d ", n);

label1 : n++;

}

will print :

1 2 4 5 7 8 10

For this task you will write a code that inputs a number N, and keeps dividing it by 2 until it is even printing all the intermediate numbers as well.

**Sample Input 0**

24

**Sample Output 0**

24 12 6 Process Complete

#include <stdio.h>

int main()

{

int num;

scanf("%d", &num);

while (num > 0)

{

if (num % 2 == 1)goto label;//write the goto statement here to skip to the end

printf("%d ", num);

num = num/2;

}

label : printf("Process Complete");

return 0;

}

## Task 3.10 : Sum of Digits

Input a number N, and print the sum of all its digits.

Hint : Use the fact that N % 10 will give the last digit of N while N/10 will eliminate its last digit.

Hint 2: You may need to use a loop

**Sample Input 0**

874

**Sample Output 0**

19

**Sample Input 1**

24567

#include <stdio.h>

int main()

{int i,n,k,sum=0;

scanf("%d",&n);

while(n>0)

{

k=n%10;

sum=sum+k;

n=n/10;

}

printf("%d",sum);

return 0;

}

## Task 3.11 : Nested For Loops

It is possible in C to write one for loop inside another. This is known as nested for loop. For example :

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

{

for (j = 0; j < n; j++)

{

//block of statements.

}

}

To complete this task write a program that takes a number N as input and prints a square block of asterisks '\*' of size NxN.

**Sample Input 0**

2

**Sample Output 0**

\*\*

\*\*

## Task 3.12 : Nested For Loops - II

Write a program that accepts a number N as input and prints the following pattern :

\*

\*\*

\*\*\*

\*\*\*\*

until N lines.

**Sample Input 0**

5

**Sample Output 0**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

#include <stdio.h>

int main()

{

int a;

int i,j;

scanf("%d",&a);

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

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

{

printf("\*");

}printf("\n");}

//printf("\n");

return 0;

}

## Z 432 Contest Participation

Our friend vassily participated in a contest on hackerrank last night, or maybe he didnt? He does not remember. He remembered his rating the day before though. And he knows his rating now.  
If he did not participate, his rating will be the same.  
If he participated and perfomed well, his rating woulve increased and if he did not perform well his rating wouldve decreased. Knowing his previous and current rating can you help him find out what exactly happened?

**Input Format**

Two numbers, X and Y. X denoting his previous rating and Y denoting is current rating.

**Constraints**

0 <= X,Y <= 3000

**Output Format**

Output "Did not participate" if Vassily did not participate. Output "Perfomed well" or "Did not perform well" accordingly if he did participate.

**Sample Input 0**

2400 2450

**Sample Output 0**

Performed well

**Explanation 0**

Since his rating changed from 2400 to 2450, it means he participated in the contest and performed well.

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int a,b;

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

cin>>a>>b;

if(a<b)

cout<<"Performed well";

else if(a>b)

cout<<"Did not perform well";

else

cout<<"Did not participate";

return 0;

}

## Z 311 FACTORS OF X

One day a teacher gave an assignment to the student to find the factors of a number. The student is not intrested to do the given task and he was searching for the shortcuts. So, please help him by writing a program

**Input Format**

First line contains an integer T denoting the no.of testcases.

The next T lines contains integers whose factors we have to determine

**Constraints**

1<=T<=10000

1<=N<=1000000

**Output Format**

Factors of number seperated by a space in the new line

**Sample Input 0**

2

4

5

**Sample Output 0**

1 2 4

1 5

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

int t; scanf("%d",&t);

while(t>0) {

int a,b;

scanf("%d",&a);

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

if(a%b==0) {

printf("%d ",b);

}

}

printf("\n");

t--;

}

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

return 0;

}

## Z 008 MULTIPLE'S OF 4

**by**[**BASICELITE**](https://www.hackerrank.com/profile/BASICELITE)

Ram is a boy who is studying in a school. One his teacher gave him the homework. The task given to him is he is given some numbers he has to find wether the given numbers are multiple's of 4 or not.

**Input Format**

A Single numeric Integer

**Constraints**

1<=N<=10000

**Output Format**

Yes if it is a multiple of 4. Else Print no

**Sample Input 0**

8

**Sample Output 0**

yes

**Explanation 0**

8 is a multiple of 4.

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

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

int n;

scanf("%d",&n);

if(n<100) {

if (n%4==0){

printf("yes");} else

printf("no");} else

{

if ((n%100)%4==0)

{

printf("yes");}

else

printf("no");}

return 0;

}

## All the factors

Your task is simple. Given a number N, print all of its factors.

**Input Format**

Only one number, N.

**Constraints**

1 <= N <= 10^6

**Output Format**

Output all the factors of the number N separated by a space on one line.

**Sample Input 0**

6

**Sample Output 0**

1 2 3 6

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,i;

cin>>a;

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

if(a%i==0)

cout<<i<<" ";

return 0;

}

## Task 2.12 : The Switch-Case Statement

https://hrcdn.net/s3_pub/hr-avatars/05bc174d-13e9-4ea6-8f5e-afa4c5450f5a/150x150.png**by [ccc\_coding\_club](https://www.hackerrank.com/profile/ccc_coding_club)**

* [**Problem**](https://www.hackerrank.com/contests/jan-20-ccc-srm-postfest-ktr-problem-solving-c-1-9/challenges/task-2-12-the-switch-case-statement)
* [**Submissions**](https://www.hackerrank.com/contests/jan-20-ccc-srm-postfest-ktr-problem-solving-c-1-9/challenges/task-2-12-the-switch-case-statement/submissions)

If-Else is the first type of selection statement in C. The second is Switch-Case.

Syntax :

switch(variable)

{

case 1 : statements to be executed if value of variable is 1

break;

case 2 : statements to be executed if value of variable is 2

break;

..

..

Default : statements to be executed if no condition is satisfied

break;

}

See the code below for better understanding, complete the rest of it so that it prints "vowel" or "not vowel".

**Sample Input 0**

a

**Sample Output 0**

vowel

**Sample Input 1**

u

**Sample Output 1**

vowel

#include <stdio.h>

int main()

{

char ch;

scanf("%c", &ch);

switch(ch)

{

case 'a' : printf("vowel"); break;

case 'e' : printf("vowel"); break;

//write the cases for i, o, and u.

case 'i':printf("vowel");

break;

case 'o':printf("vowel");break;

case 'u':printf("vowel"); break;

default : printf("not vowel"); break;

}

return 0;

}

## Task 2.11 Grading System

Write a program that inputs a student's marks and prints his grade according to the following rules :

Marks Grade

76 - 100 A

51 - 75 B

26 - 50 C

0 - 25 D

**Sample Input 0**

37

**Sample Output 0**

C

**Sample Input 1**

85

**Sample Output 1**

A

#include <stdio.h>

int main()

{

int m;

char grade;

scanf("%d", &m);

//write the code to assign the grade

if(m>=0 && m<=25)

{

printf("D");

}

//printf("D");

else if(m>=25 && m<=50)

{

printf("C");

}

else if(m>=50 && m<=75)

{

printf("B");

}

else if(m>=76 && m<=100)

{

printf("A");

}

//printf("%c", grade);

return 0;

}

## Z 302 Weird Challenge

Given an integer,n, perform the following conditional actions:

If n is odd, print Weird If n is even and in the inclusive range of 2 to 5, print Not Weird If n is even and in the inclusive range of 6 to 20, print Weird If n is even and greater than 20, print Not Weird

**Input Format**

A single line containing a positive integer,n.

**Constraints**

1<=n<=100

**Output Format**

Print Weird if the number is weird; otherwise, print Not Weird

**Sample Input 0**

3

**Sample Output 0**

Weird

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int n;

cin>>n;

if(n%2==0)

{

if(n>=2 && n<=5)

{

cout<<"Not Weird";

}

if(n>=6 && n<=20)

{

cout<<"Weird";

}

if(n>20)

{

cout<<"Not Weird";

}

}

else

cout<<"Weird";

}

## Barua Skywatcher

Barua is standing at x = 0 facing the positive x-axis. He loves watching stars but is too lazy to move or even turn around at that matter. For the same reason he can see the stars located in the first quadrant but cannot see the ones located in the second quadrant (because he doesn't have eyes on his back). You are given the co-ordinates of N stars and you need to tell him the fraction of the total stars that he will be able to observe.

**Input Format:** First line contains the integer N.  
Next N lines contain two integers each, the x and the y co-ordinate of the ith star.

**NOTE :  
Barua can observe every star that is in front of him i.e in the first quadrant, the only ones he cannot observe are the ones behind him i.e. lying in the second quadrant. He cannot observe the stars that are vertically above him.  
The third and the fourth quadrants are obviously the ground that Barua stands on.**

**Output Format:**  
Output one decimal number denoting the percentage of the total stars that Barua can observe.

**NOTE : Print the answer upto 6 decimal places.**

**Input Constraints:**

* 1 <= N <= 10^5
* 1 <= y[i] <= 1000
* -1000 <= x[i] <= 1000

**Sample Input 0**

6

2 2

5 6

-4 5

-1 1

7 3

-4 2

**Sample Output 0**

0.500000

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/ int t,x,y,n=0,z; float p; scanf("%d",&t); z=t; while(t--) { scanf("%d %d", &x ,&y); if(x>0) n++; else if(x<0 && y<0 ) n++;

}

p=(float)n/z;

printf("%.6f",p);

return 0;

}

## Reincarnation

In the world of numbers every number dies at some point and reincarnates. The newer generation will always surpass the previous generation and hence every number X upon incarnation will be promoted with a power Y i.e. X will now become X raised to the power Y. The god of the world of numbers needs a small break and hence has given you the task of reincarnating the incoming numbers after raising them to the power Y.

**Input Format**

Two numbers, X and Y.

**Constraints**

1 <= X <= 10  
0 <= Y <= 10

**Output Format**

One number, X raised to the power Y.

**Sample Input 0**

2 4

**Sample Output 0**

16

**Explanation 0**

2 raised to the power 4 = 2 \* 2 \* 2 \* 2 = 16

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

long int a,b,k;

cin>>a>>b;

k=pow(a,b);

cout<<k;

return 0;

}

**Z 321 Add Two Values**

Given two values, they can be integer or floating point numbers, add them.

**Input Format**

Two values separated by a space.

**Constraints**

No constraints.

**Output Format**

One value, the result of sum of two input values.

**Sample Input 0**

5 4

**Sample Output 0**

9

**Explanation 0**

5 + 4 = 9

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

float a,b;

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

long int c,d;

c=a;

d=b;

if(c-a==0&&d-b==0)

printf("%.f",a+b);

else

printf("%.2f",(a+b));

//float a,b; long int c; scanf("%f %f",&a,&b); c=a; if(c-a==0) printf("%.f",a+b); else printf("%.2f",a+b);

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

return 0;

}

## Z 435 NUMBER PATTERN-1 1

PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.

**Input Format**

A SINGLE INTEGER DENOTING N VALUE

**Constraints**

1<=N<=100

**Output Format**

PATTERN AS SHOWN IN SAMPLE TEST CASE

**Sample Input 0**

4

**Sample Output 0**

1

12

123

1234

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int x;

cin>>x;

for(int i=1;i<=x;i++)

{

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

{

cout<<j;

}

cout<<endl;

}

return 0;

}

## Z 436 NUMBER PATTERN-2 1

PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.

**Input Format**

A SINGLE INTEGER DENOTING N VALUE

**Constraints**

1<=N<=100

**Output Format**

PATTERN AS SHOWN IN SAMPLE TEST CASE

**Sample Input 0**

5

**Sample Output 0**

1

01

101

0101

10101

**Sample Input 1**

3

**Sample Output 1**

1

01

101

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int i, j;

int count = 1;

int n;

cin>>n;

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

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

{

printf("%d", count % 2);

count++;

}

printf("\n");

if (i % 2 == 0)

count = 1;

else count = 0; }

return 0; }

## NUMBER PATTERN

PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.

**Input Format**

A SINGLE INTEGER DENOTING N VALUE

**Constraints**

1<=N<=100

**Output Format**

PATTERN AS SHOWN IN SAMPLE TEST CASE

**Sample Input 0**

5

**Sample Output 0**

1

12

123

1234

12345

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

int n;

scanf("%d",&n);

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

{

for(int j=n;j>i;j--)

{

printf(" ");

}

for(int k=1;k<=i;k++)

{

printf("%d",k);

}

printf("\n");

}

return 0;

}

## Z 439 NUMBER PATTERN-4 1

* [**Problem**](https://www.hackerrank.com/contests/jan-20-ccc-srm-postfest-ktr-problem-solving-c-1-9/challenges/z-439-number-pattern-4-1)
* [**Submissions**](https://www.hackerrank.com/contests/jan-20-ccc-srm-postfest-ktr-problem-solving-c-1-9/challenges/z-439-number-pattern-4-1/submissions)

PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.

**Input Format**

A SINGLE INTEGER DENOTING N VALUE

**Constraints**

1<=N<=100

N is only odd

**Output Format**

PATTERN AS SHOWN IN SAMPLE TEST CASE

**Sample Input 0**

5

**Sample Output 0**

1 5

2 4

3

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int i,j,a;

cin>>a;

for(i=1;i<=a/2+1;i++)

{

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

if(i==j||i+j==a+1)

cout<<j;

else cout<<" ";} cout<<"\n";

}

return 0;

}

## Z 438 NUMBER PATTERN-5 1

PRINT THE BELOW MENTIONED PATTERN FOR ANY "N" VALUE. WHERE "N" INDICATES NO.OF ROWS.

**Input Format**

A SINGLE INTEGER DENOTING N VALUE

**Constraints**

1<=N<=100

**Output Format**

PATTERN AS SHOWN IN SAMPLE TEST CASE

**Sample Input 0**

5

**Sample Output 0**

1

2

3

4

5

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

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

int n,i,j;

scanf("%d",&n);

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

{

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

{

if(j==i)

printf("%d",j);

else

printf(" ");

}

printf("\n");

}

return 0;

}

## Z 402 Rava Idli

A little girl living in a village craves some rava idli even though she has had rava idli for the last 346514534 days in a row !! (Strange, you might think. But its normal down here)

At the idli shop there are two types of Rava Idli's available.

One goes for Rs.A per piece and the other goes for Rs.B per piece.  
The girl has a total of K rupees.

What is the maximum number of rava idlis that she can have?

Note that she does not care about the type of idli she gets, she just wants to have as many rava idlis of any type as possible.

Input

The first line contains the number of test cases T  
1 ≤ T ≤ 1000  
Each test case contains three integers, A, B and K.  
1 ≤ A,B,K ≤ 10^9

Output

Print the maximum number of idlis she can buy for each test case on a new line

**Sample Input 0**

5

5 5 21

4 5 21

4 3 20

3 2 21

1 2 20

**Sample Output 0**

4

5

6

10

20

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int a,b,k,i;

cin>>i;

while(i--)

{

cin>>a>>b>>k;

if(a>b)

cout<<k/b<<"\n";

else cout<<k/a<<"\n";

}

return 0;

## }

## T 114 - Equal Bases

Given a decimal number N, you need to find the number of bases 1 < b <= n such that when the number N is represented in base 'b', it ends in a zero.

**Input Format**

Input contains only one number, N.

**Constraints**

1 <= N <= 10000

**Output Format**

Output one number, the number of bases b such that when number N is represented in base b, the number contains a trailing zero.

**Sample Input 0**

12

**Sample Output 0**

5

**Explanation 0**

Base : Number :  
2 1100  
3 110  
4 30  
5 22  
6 20  
7 15  
8 14  
9 13  
10 12  
11 11  
12 10

12 ends with a zero in 5 different number systems - 2,3,4,6 and 12. Hence the answer is 12.

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {int n,c=0;

cin>>n;

for(int i=2;i<=n; i++ )

{

if(n%i==0)

c++;

}

cout<<c;

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

return 0;

}

## Number of digits in N!

As simple as the title, given a number N, print the number of digits in N!

N! is defined as : N! = 1\*2\*3...(N-1)\*N

0! = 0 and 1! = 1.

No number ever contains any leading zeros.

**Input Format**

Input contains only one number, N.

**Constraints**

1 <= N <= 1000

**Output Format**

Output one number that is equal to the number of digits in N!

**Sample Input 0**

6

**Sample Output 0**

3

**Explanation 0**

6! = 1x2x3x4x5x6 = 720 which has 3 digits. So the answer is 3.

import math

n=int(input());

x=math.factorial(n)

s=str(x)

count=0

for i in s:

count+=1;

print(count)

#print(x)

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() { int n; double sum=1;

scanf("%ld",&n);

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

{

sum+=log10(i);

}

long x=sum;

printf("%d",x);

return 0;

}

## C D09 - To and Fro

A futuristic company is building an autonomous car. The scientists at the company are training the car to perform Reverse parking. To park, the car needs to be able to move in backward as well as forward direction. The car is programmed to move backwards B meters and forwards again, say F meters, in a straight line. The car does this repeatedly until it is able to park or collides with other objects. The car covers 1 meter in T units of time. There is a wall after distance D from car's initial position in the backward direction.

The car is currently not without defects and hence often hits the wall. The scientists are devising a strategy to prevent this from happening. Your task is to help the scientists by providing them with exact information on amount of time available before the car hits the wall.

**Input Format:**

First line contains total number of test cases, denoted by N Next N lines, contain a tuple containing 4 values delimited by space F B T D, where 1. F denotes forward displacement in meters 2. B denotes backward displacement in meters 3. T denotes time taken to cover 1 meter 4. D denotes distance from Car's starting position and the wall in backward direction

**Output Format:**

For each test case print time taken by the Car to hit the wall

**Constraints:**  
First move will always be in backward direction  
1 <= N <= 100  
backward displacement > forward displacement i.e. (B > F)  
forward displacement (F) > 0  
backward displacement (B) > 0  
time (T) > 0  
distance (D) > 0  
All input values must be positive integers only

**Sample Input 0**

2

6 9 3 18

3 7 5 20

**Sample Output 0**

162

220

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

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

int n;

cin>>n;

while(n--)

{

int f,b,t,d;

cin>>f>>b>>t>>d;

int tot=0;

int r=d;

while(r>b)

{

tot=tot+f+b;

r=r-(b-f);

}

tot+=r;

cout<<tot\*t<<"\n";

}

return 0;

}