LAB-PRAC-05_CONDLOOPS

Forgetful Mr C (p1v1d1)

Forgetful Mr C [10 marks]

Problem Statement
Mr. C is teaching the course ESC101 next semester. Given that the number of students and sections in the course is very large, he is unable to remember the lab day and the tutorial rooms for every section. He turns to you for help! Using the table given below, write a program which take input a section number in the format given below and print the lab day and the tutorial room in the format given below.

Note?: If the section provided as input is not present in the table given below then your program should print "Incorrect section!" (without quotes).

Section	Day of Lab	Tutorial Room
B1	**Monday**	TB 105
B2	**Monday**	TB 106
B3	**Monday**	TB 111
B4	**Tuesday**	TB 112
B5	**Tuesday**	TB 203
B6	**Tuesday**	TB 204
B7	**Thursday**	TB 205
B8	**Thursday**	TB 206
B9	**Thursday**	TB 207
B10	**Wednesday**	TB 208
B11	**Wednesday**	TB 209
B12	**Wednesday**	TB 210
B13	**Tuesday**	TB 211
B14	**Wednesday**	TB 212

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful about spelling mistakes while printing various text messages.

EXAMPLE:
INPUT
B1

OUTPUT:
Monday TB 105

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
B1	Monday TB 105
B100	Incorrect section!
B0	Incorrect section!
B10	Wednesday TB 208
B13	Tuesday TB 211
B7	Thursday TB 205

Rich Mr C (p1v2d1)

Rich Mr C [20 marks]

Problem Statement
The family of Mr. C is super rich. Each sibling of Mr. C has a single but vast piece of land in the outskirts of Kanpur city. The shape of the lands owned by siblings of Mr C can be either a square or a triangle. They want to sell their lands but they donâ€™t have the information about its area. All they know is the shape of their land and length of sides. They ask Mr. C to help them with the calculation of the area.

You are given as input, the shape of the land (represented as an integer as shown below). If the shape is a square, you are given its side length. If the shape is a triangle, then all three side lengths are given. Side lengths may be given as **non-integers**. Print the name of the shape and its area **rounded to three decimal places**. These two outputs must be printed on **different lines**.

Caution

1. Use **double variables** and double typecasting while doing area calculations to avoid loss of precision. Do not use float variables and float typecasting.
2. Be careful about extra/missing lines and extra/missing spaces.
3. A square is represented by the integer 0
4. A triangle is represented by the integer 1
5. To calculate the area of triangle, Mr. C recalls the Heronâ€™s formula from his high school. If a, b, c are the side lengths of a triangle, then if we let s = (a + b + c)/2 then
   $$area = \sqrt{s(s-a)(s-b)(s-c)}$$

EXAMPLE:
INPUT
0 2
OUTPUT:
Square
4.000

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
0 2.0	Square 4.000
1 2 2 2	Triangle 1.732
0 2.5	Square 6.250
1 20 20 20	Triangle 173.205
1 10000 10000 10000	Triangle 43301270.189
0 125.5	Square 15750.250

Perfect Numbers (p1v3d1)

Perfect Numbers [20 marks]

Problem Statement
A positive integer is called a perfect number if the sum of proper divisors of that number is equal to the number itself. When you go back home, you can read more about this here: https://en.wikipedia.org/wiki/Perfect_number (however, DO NOT BROWSE these websites during the lab).

The proper divisors of a number are all those numbers, except the number itself, which divide that number. Thus, 1 is a proper divisor of every number but 1 itself. For example, the proper divisors of 6 are: 1, 2 and 3. Clearly, 6 = 1 + 2 + 3. Thus, 6 is a perfect number.

You will be given **a positive number as an int** as input and you will have to give two outputs in **two different lines**. In the first line, give the sum of the proper divisors of that number and in the second line, print "YES" (without quotes) if the number is a perfect number, otherwise print "NO".

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. This question will require you to use loops. We are giving below, the general structure of a loop, to help you.

HINTS:

1. A loop is written as follows
   int counter;
   for(counter = 1; counter <= 5; counter++){
   // Do something with the counter
   }
2. You can change the initial value 1, final value 5, and the update step as you wish.
3. Inside a for loop, you can put in any number of statements.
4. Be careful that the sum of the divisors of a number may not fit inside the int datatype even if the number itself is int datatype. To avoid such errors, use the long datatype to perform computations.

EXAMPLE:
INPUT
28

OUTPUT:
28
YES

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
28	28 YES
15	9 NO
1	0 NO
8128	8128 YES
100000000	149511591 NO
150000	334344 NO

Mr C builds a Calculator (p2v1d1)

Mr C builds a Calculator [10 marks]

Problem Statement
Mr. C is fed up of making calculation mistakes in his CHM101 lab assignments and decides to build a scientific calculator to help him do calculations accurately. He needs some of the basic arithmetic and trigonometric functions for the lab. Help Mr C build this calculator.

You will be given an operation (e.g. addition, sin etc) represented using a code given below, followed by the inputs for that operation. Write a program to read the operation as well as the inputs and then perform the operation on those inputs. Your answers should always be printed rounded off to 4 decimal places.

Operation Code	Operation Symbol	Input1	Input2	Output
1	+	a	b	a+b
2	-	a	b	a-b
3	*	a	b	a*b
4	/	a	b	a/b

5	pow	a	b	pow(a,b)
6	sin	a		sin(a)
7	cos	a		cos(a)
8	fabs	a		fabs(a)

Caution

1. Note that the input numbers **may be non-integer**
2. It is advised that you use double datatype and double typecasts in your code
3. Note that some of the above operations are binary e.g. * and pow, whereas others are unary e.g. sin, fabs. If the operation is a unary operation, we will supply you only one number, if the operation is a binary operation, we will supply you two numbers.
4. Be careful about extra/missing lines and extra/missing spaces.

EXAMPLE:
INPUT
1 2 3

OUTPUT:
5.0000

Note 1 stands for addition, so we added the two numbers 2 and 3 and output it rounded to four decimal places as 5.000

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
1 2 3	5.0000
8 -2.5	2.5000
2 3 4	-1.0000
5 2 10	1024.0000
6 3.14	0.0016
7 6.28	1.0000

Love for Primes (p2v2d1)

Love for Primes [20 marks]

Problem Statement
Given a positive **integer** N >= 2, write a program to find the smallest divisor D of N such that D > 1. Further if N is prime, then **on a second line** print "PRIME" (without quotes) otherwise print "COMPOSITE" (without quotes).

Caution

1. The smallest divisor D of N such that D > 1 may be N itself (if this does happen, what does this say about the number N)?
2. We will always give you a value of N that can fit inside int variable
3. Be careful about the primality of small single digit numbers
4. Be careful about extra/missing lines and extra/missing spaces.

EXAMPLE:
INPUT
3

OUTPUT:
3
PRIME

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
2	2 PRIME
323	17 COMPOSITE
3	3 PRIME
10000005	3 COMPOSITE
1022117	1009 COMPOSITE

15478777	15478777 PRIME

Tryst with Taylor (p2v3d1)

Tryst with Taylor [20 marks]

Problem Statement
Given a **real number** x such that |x| < 1, and a **positive integer** k, we can calculate the value of log(1+x) using the first k terms of the Taylor series for the log function as follows

$$
\log(1+x) = x - \frac{x^2}2 + \frac{x^3}3 - \frac{x^4}4 + \ldots + (-1)^{k+1}\frac{x^k}k
$$

We will give you the real number x such that |x| < 1 and a positive integer k. You have to output **on two different lines**

1. the value of log(1+x) calculated using the Taylor series upto 3 terms
2. the value of log(1+x) calculated using the Taylor series upto k terms

Give all outputs **rounded to four decimal places**

Caution

1. The number x may be negative too
2. Use double variable to store x and use double typecasts and computations for this problem.
3. Do not attempt to cheat and use the math.h function log() to compute this. The math.h function uses a really large number of terms while calculating the log function whereas we will ask you to use smaller number of terms. You will fail the test cases if you try to use the math.h log function.
4. Be careful about extra/missing lines and extra/missing spaces.

INPUT:
x k

OUTPUT:
log(1+x) computed using the first 3 terms of the Taylor series
log(1+x) computed using the first k terms of the Taylor series

EXAMPLE:
INPUT

OUTPUT:

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
0.1 5	0.0953 0.0953
0.1 1	0.0953 0.1000
-0.1 1	-0.1053 -0.1000
0.9 2	0.7380 0.4950
-0.9 6	-1.5480 -1.9187
0 6	0.0000 0.0000

Mr C is very busy (p3v1d1)

Mr C is very busy [10 marks]

Problem Statement
Mr. Câ€™s son Cc is in primary school. He knows the textual word representation for numbers from 1 to 20 (e.g. 11 is eleven, 19 is nineteen, and so on). Next week Cc has to submit a homework in which he has to write the word representation for numbers from 21 to 99.

Now Mr. C is a very busy man as he has to run all the loops you are making him execute this week. He doesnâ€™t have the time to teach his son Cc the word representation for all the numbers between 21 and 99. Write a program which will take an input integer from 21 to 99 and print itâ€™s word representation.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. In particular, for the number 40, we want the text "Forty" (without the quotes) and not the string "Forty " (note the extra space at the end).
3. Please note that the evaluation may show you that you have passed the test case even if you write "Forty ". However, the final autograder will give you zero marks if you do this.
4. Also note that for words like 22, the expected output is "Twenty two" (without the quotes). Note the spelling of the word "two" is in small case.
5. Look at the visible test cases carefully to see where you are going wrong.
6. You may use copy-paste to avoid typing code again and again. However, be careful not to make typing errors while doing so.

EXAMPLE 1:
INPUT
40

OUTPUT:
Forty

Note: no extra space after the word Forty

EXAMPLE 2:
INPUT
22

OUTPUT:
Twenty two

Note: only one space between the words Twenty and two no extra space after the word two
----

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
40	Forty
22	Twenty two
35	Thirty five
99	Ninety nine
67	Sixty seven
80	Eighty

Fabulous Fibonacci (p3v2d1)

Fabulous Fibonacci [20 marks]

Problem Statement
The Fibonacci sequence forms one of the most important number series in the nature. Many natural phenomenon follows this sequence. However, we will look at an even more general class of sequences known as Lucas sequences in this question. You will be given three **integers**, P, Q, n. n will always be non-negative.

Given P, Q, the Lucas numbers of the first kind are defined as follows
$$\begin{align}
L_0 &= 0\\
L_1 &= 1\\
L_n &= P \cdot L_{n-1} - Q \cdot L_{n-2},\qquad\text{for } n \geq 2
\end{align}$$
You will first have to output L_5 for the given value of P and Q. In **the next line** you will have to output L_n for the given value of P and Q.

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Remember that n can be 0 or 1 too
3. Remember that P, Q can be negative or positive integers too
4. Lucas numbers can grow very very quickly. Use a long datatype to perform all your computations.
5. All Lucas numbers are integers. Using float or double variables will result in errors.
6. You may use copy-paste to avoid typing code to the print the first and second lines twice. However, be careful not to make typing errors while doing so.
7. Warning: if you use try to use recursion to solve this problem, you may land up with a TLE error

----

INPUT:
P Q n

OUTPUT:
L_5
L_n

EXAMPLE:
INPUT

OUTPUT:

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
1 -1 20	5 6765
3 -5 23	241 38537005750589

3 -5 0	241 0
2 4 35	-16 -17179869184
5 4 25	341 375299968947541
4 -3 1	409 1

Digit Debacle (p3v3d1)

Digit Debacle [20 marks]

Problem Statement
You will be given two **strictly positive integer** (i.e. positive and non-zero) N and another **non-negative integer** K. We promise that the integers will not contain any leading zeros i.e. we will never give you N as 0123 or 00123 -- instead we will always give N as 123 i.e. the first digit of N in the way we give it to you will always be non-zero. You will have output two things on **two different lines**

1. On the first line, print the number of digits in the number N
2. On the second line, print "YES" (without quotes) if the sum of the first K digits of N is greater than K*K. If not, print "NO" (without quotes).

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful about the capitalization of the words YES and NO.
3. The number N we give you might not fit inside an int variable so use a long variable to store N.
4. Be careful that K may be zero as well

HINTS:

1. The function log10(n) available from math.h gives you back the base 10 logarithm of a number n. This function can help you in finding the number of digits in an integer n.
2. We can extract the first i.e. leading digit of any integer n by using the formula
   $$(\text{int})(n/\text{pow}(10,len - 1))$$
   where len is the number of digits in n.

EXAMPLE:
INPUT
125 2

OUTPUT:
3
NO

Explanation: The number of digits in 125 is 3. The sum of the first 2 digits is 3 which is not greater than 2*2 = 4 so the second answer is NO.

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
5 1	1 YES
105 3	3 NO
18999 4	5 YES
133 0	3 NO
987654321987 9	12 NO
111111111111111111 1	18 NO

May the fourth be with you (p4v1d1)

May the fourth be with you [10 marks]

Problem Statement
On the coming 4th of May, IIT Kanpur astronomy club is planning to celebrate Star-Wars day. For 10 days starting from 4th of May and lasting till 13th of May, movie screenings are planned for original as well as prequel movies of Star-Wars series.

The schedule is given below:

Dates	Movie
4 May, 7 May, 10 May	The Phantom Menace
5 May, 11 May	Attack of the Clones
6 May, 8 May, 9 May, 12 May	Revenge of the Sith
13 May	A New Hope

Write a program which will take input a date **in the form of an integer** and print the name of the movie that will be screened on that day, if the input date is not on the schedule, print "Outside schedule!" (without quotes).

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Note that the same movie is shown on multiple days
3. Be very careful about spelling mistakes and capitalization errors. Autograder will mark all spelling mistakes and capitalization errors with a zero.

EXAMPLE:
INPUT
14

OUTPUT:
Outside schedule!

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

All Test Cases (Visible + Hidden)

Input	Output
9	Revenge of the Sith
15	Outside schedule!
7	The Phantom Menace
1	Outside schedule!
5	Attack of the Clones
13	A New Hope

Phone a friend (p4v2d1)

Phone a friend [20 marks]

Problem Statement
In a planet called Vormir, there is a weird system for determining friendship between two persons.

- If the **number of even digits** in their telephone numbers are the same, then the two persons are friends
- If two persons are not friends but the **sum of the number of even digits** in their telephone numbers is even, then they are acquaintances.
- If two persons are neither friends nor acquaintances, they are enemies.

It is known that the planet has an **eight digit phone number** system. Write a program which takes as input, two phone numbers given as **positive integers** separated by a space and first prints the total number of even digits in each phone number, **separated by a space**, then in a **new line**, prints the relationship of the two people (Friends, Acquaintances, Enemies).

Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Be careful not to make spelling mistakes in the output. The label should be either of "Friends", "Acquaintances", or "Enemies" (without quotes)
3. 0 is considered and even number

EXAMPLE:
INPUT
11111111 22222222

OUTPUT:
0 8
Acquaintances

Explanation 0 is not equal to 8 but their sum is 8 which is even
----

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
11111111 22222222	0 8 Acquaintances
12121212 90909009	4 4 Friends
33445567 11100000	3 5 Acquaintances
11805520 11107305	4 2 Acquaintances
12345678 12345678	4 4 Friends
00000000 00000000	8 8 Friends

The legend of Chess (p4v3d1)

The legend of Chess [20 marks]

Problem Statement
There is a famous legend related to the origin of the game of chess. There was an emperor who was very impressed by this game and wished to reward the inventor. The conversation between them is given below. In this world however, the chessboard only has 7 squares on each side, for a total of 49 squares on the board.

Emperor: Name your reward, O wise one!
Inventor: O generous emperor, my wish is simple. I only wish for this -- give me one grain of rice for the first square of the chessboard, two grains for the next square, four for the next, eight for the next and so on for all 49 squares, with each square having double the number of grains as the square before.

We will provide you with the number of rice grains present with the emperor as a **long integer**

1. If the king is able to fulfill his promise with those many rice grains, print **YES** in the first line and then in the **next line**, print how many grains of rice he will be left with after fulfilling his promise.
2. If the king is not able to fulfill his promise with those many grains of rice, print **NO** in the first line and then in the **next line**, print the last square he was able to **completely fill**. This will be a number between 0 and 49.

Caution

1. The number of rice grains can be very large. Use a long type variable to do calculations.
2. Be careful about extra/missing lines and extra/missing spaces.

EXAMPLE:
INPUT
2

OUTPUT:
NO
1

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries 50% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

All Test Cases (Visible + Hidden)

Input	Output
2	NO 1
1125899906842624	YES 562949953421313
3	NO 2
0	NO 0
562949953421311	YES 0
1023	NO 10