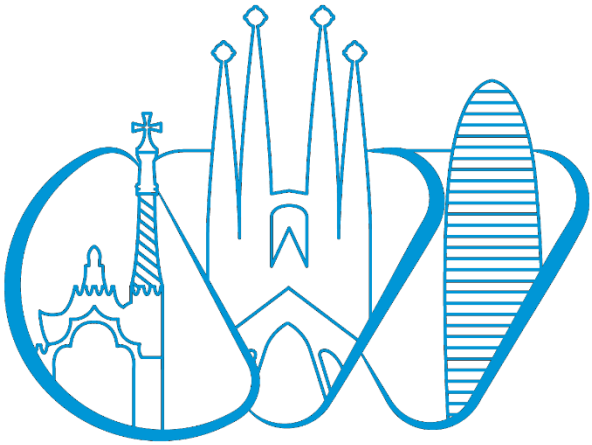


CodeWars

BCN 2023

Contestant's Guide

Important contest information

Please, read the following instructions carefully. They contain meaningful information for the contest. If you have any questions regarding these points, please ask for support before the contest starts.

Before starting...

All teams will be given the same set of problems to solve.

Problems? Each problem will have a specified point value. The more difficult the problem, the more points a correct solution will receive.

Points? All or nothing, there are no intermediate scores. You will get the maximum points only if the problem is correctly solved using a suitable algorithm.

A suitable algorithm? It has no sense solving a problem by directly printing out the correct outputs based on the inputs that appear in the problem tests. Don't fall to the Dark Side!

Note that the jury may have prepared public and private multiple test files for each problem. The judgement system will report back the first incorrect result as verdict.

If any team decides to break the rules and use inappropriate programming methods (hardcoding, cheating, ...) to solve a problem, the team will be disqualified.

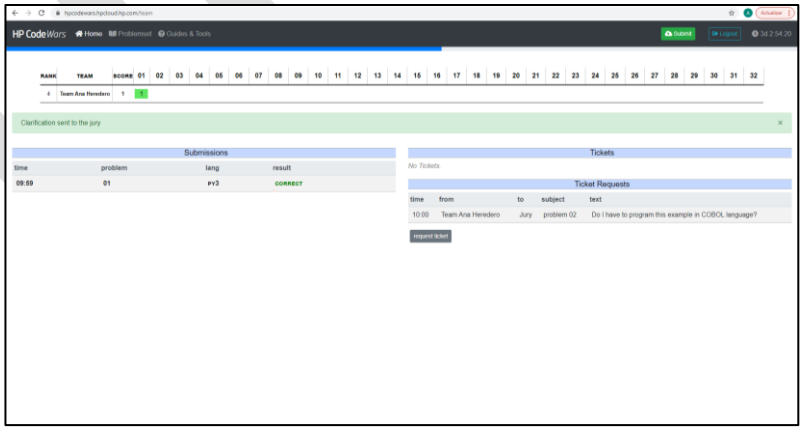
If a question is unclear, participants may request clarification from the jury. If a clarification is warranted, it will be posted on the contest site.

How to connect to the judgement system

Login into the competition website http://cw-bcn-comp-1.auth.hpicorp.net/

using the username and password the organization has provided to your team.

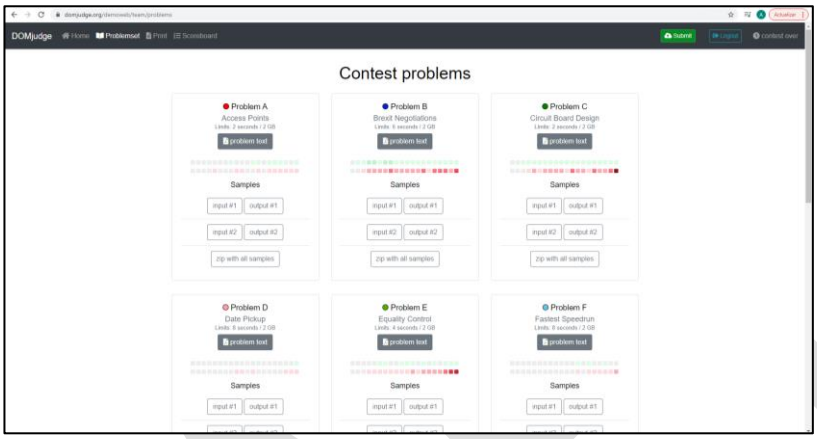
This is how will look the team web interface overview page.



The HP CodeWars contest will start on Saturday March 4th at 10:30h.

Problemset

In the team web interface overview page, you can quickly access to the any single problem statement as a PDF document opened in a new tab by clicking over the numbers under the menu bar.
For a detailed view please find on the upper left at the menu bar and next to Home the Problemset icon. Hit there and you will get the whole contest problems. For each problem there is a problem statement in the form of a PDF document and a set of public input and output tests. There is also an option to get the zip file with all samples.



Submitting solutions

Solutions can be submitted from the web interface. The solutions must be developed in a single source code file in one of the supported programming languages: C, C++, Java or Python 3. Just click the green Submit button at the menu bar on every page. Then click the file selection button and select one or multiple files for submission. The server will try to determine the problem and language. Otherwise, select the appropriate values.

After you hit the submit button and confirm the submission, you will be redirected back to your submission list page. On this page, a message will be displayed that your submission was successful, and the submission will be present in the list. An error message will be displayed if something went wrong.

Checking the results of submissions

The left columns of your team web page show an overview of your submissions. It contains all relevant information: submission time, programming language, problem and status.

Clarifications

All communication with the judges is to be done through clarification messages. These can be found in the right column on your team page. Both, clarification replies from the judges and requests sent by you are displayed there.

There is also a button to submit a new clarification request to the judges; you can associate a specific problem or one of the general categories to a request. This clarification request is only readable for the judges. The judges can answer specifically to your team or send a reply to everyone if it is relevant for all.

General considerations

It is recommended to start out by finding a suitable problem to solve. Then proceed to code the solution to the problem. After this, you submit the code for review. At this stage your code will be compiled and executed with some private test inputs. Finally, our judgement system will inform you whether your code behaved as expected or not. The possible verdict outputs are:

CORRECT

The submission passed all tests: you solved this problem!

COMPILER-ERROR

There was an error when compiling your program. On the submission details page, you can inspect the exact error. Note that when compilation takes more than 30 seconds, it is aborted and this counts as a compilation error.

TIMELIMIT

Your program took longer than the maximum allowed time for this problem. Therefore, it has been aborted. This might indicate that your program hangs in a loop or that your solution is not efficient enough.

RUN-ERROR

There was an error during the execution of your program. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g. by indexing arrays out of bounds), trying to use more memory than the limit, etc. Also check that your program exits with exit code 0!

NO-OUTPUT

Your program did not generate any output. Check that you write to standard out.

OUTPUT-LIMIT

Your program generated more output than the allowed limit. The output was truncated and considered incorrect.

WRONG-ANSWER

The output of your program was incorrect. This can happen simply because your solution is not correct but remember that your output must comply exactly with the specifications of the jury.

TOO-LATE

The submission was sent after the contest ended! Your submission is stored but will not be processed anymore.

None of the submitted files match any of the allowed extensions for *language*
(allowed: *list of file extensions*)

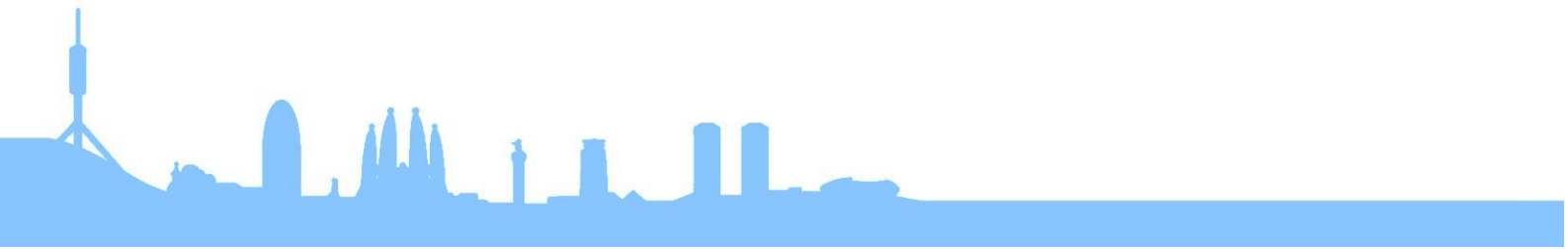
The file you have submitted does not have a valid extension filename and will not be judged. Check that the extension you sent is one of the valid extensions for each supported language:

- C – Allowed file extensions: c
- C++ – Allowed file extensions: cpp, cc, cxx, c++
- Java – Allowed file extensions: java
- Python 3 – Allowed file extensions: py, py3

Note that the jury may have prepared public and private multiple test files for each problem. The judgement system will report back the first incorrect result as verdict.

Contestants may write their programs in whichever language they prefer from the following programming languages. In parenthesis figures the current compiler version used by our judgement system. You are allowed to use all standard libraries.

- C (gcc 8.3.0)
- C++ (g++ 8.3.0)
- Java SE 11 (11.0.14)
- Python3 (3.7.3)



Input/output

All programs are text based. Your program should read its input from standard input and produce output on standard output. Input will always follow the input specification described in the problem (so you do not need to validate the input). Your output must follow the output specification.

The sample data provided in the problem statement is just there to help you make sure you understood what the problem asks for, and the input/output format. When you submit your solution, we will run it on an extensive set of additional test data to verify that it solves the problem correctly and efficiently.

Useful definitions

The following definitions can be useful for your problems.

English number notation

In English, comma (,) is used for thousand grouping, while dot (.) is used for decimals. For instance: 1,678.354 is read "one thousand six hundred and seventy-eight and 354 thousandths."

Leap year calculation

A leap year is that one with 366 days, when February has a 29th day.
This happens when the year number is multiple of 4 (divisible by 4) except for the multiples of 100 that are not multiples of 400. For example, 2000 is leap but 1700, 1800 and 1900 are not.

Roman Numerals

Roman Numerals were used by the ancient Romans as their numbering system. This system is base decimal, like the numbers we use today, but there is no number zero. It only uses the letters I, V, X, L, C, D, M and these letters are combined to make different whole numbers.

Roman Numeral	I	V	X	L	C	D	M
Arabic numeral	1	5	10	50	100	500	1000

There are a few rules for writing numbers with Roman Numerals.

1. Repeating a numeral up to three times represents addition of the number. For example, III represents 1 + 1 + 1 = 3. Only I, X, C, and M can be repeated; V, L, and D cannot be, and there is no need to do so.

2. Writing numerals that decrease from left to right represents addition of the numbers. For example, LX represents 50 + 10 = 60 and XVIII represents 10 + 5 + 3 = 18.

3. To write a number that otherwise would take repeating of a numeral four or more times, there is a subtraction rule. Writing a smaller numeral to the left of a larger numeral represents subtraction. For example, IV represents 5 - 1 = 4 and IX represents 10 - 1 = 9.

Units of information

In computing and telecommunications, a unit of information is the capacity of some standard data storage system or communication channel, used to measure the capacities of other systems and channels.

The most common units are the bit, the capacity of a system which can exist in only two states, and the byte, which is equivalent to eight bits. Multiples of these units are kilobyte (kB), megabyte (MB), gigabyte (GB), terabyte (TB) and petabyte (PB).

1 kB	1024 bytes
1 MB	1024 kB
1 GB	1024 MB
1 TB	1024 GB
1 PB	1024 TB

Divisible/Remainder/Modulo operation

Divisible numbers

Two numbers are divisible when the reminder of their natural division equals 0. For example, 8 is divisible by 4, 15 is divisible by 3, 100 is divisible by 10...

Remainder

The remainder of a division is the integer "left over" after dividing an integer by another.

Examples:

- If you divide 9 by 2, the result is 4 and the remainder is 1.
- If you divide 17 by 3, the result is 5 and the remainder is 2.

Modulo operator

It is the operator that outputs the remainder of the division. It is expressed by % in all the supported languages. For example:

17 % 3 would be evaluated as 2

Working with decimal numbers

Floating-point numbers have a limited number of digits, so they cannot represent all real numbers accurately. Consider the case of fraction 1/3 which can be represented as the decimal number 0.33333333333... which has an infinite length. But computers do not have infinite memory to store them, instead computers have a few bytes.

Therefore, floating point numbers store only a certain number of significant digits, and the rest are lost. The precision of a floating-point number defines how many significant digits it can represent without information loss.

Each programming language has a default precision when a floating point is printed out. For example, in C++ the default precision is 6, which means up to 6 significant digits are used to represent a number. So the good news are that there are functions in the programming languages that allow you to set specifically the precision to be output.

In the same way there are also several functions in the programming languages that make your life easier when handling floating point numbers:

Round – Rounds off the given floating-point value to the closest integer, with halfway cases rounded away from zero.

Ceil – Rounds up the given floating-point value to the closest integer which is more than the given value.

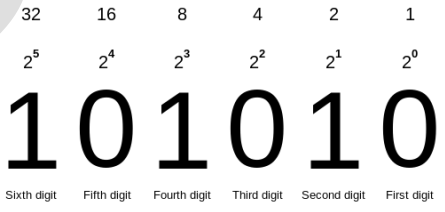
Floor – Rounds down the given floating-point value to the closest integer which is less than the given value.

While solving problems keep this in mind and avoid performing any explicit type conversion neither rounding. Once you get the result and before printing it, then you can apply the proper conversion or rounding to get the requested number of decimals in the problem statement.

Binary numbers

The decimal number system also known as arabic numerals is a positional number system. It only has symbols for the first ten values, including a symbol for zero. The only symbols in decimal are: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 and form a base-10 numeral system. The reason that only ten symbols are needed (no matter how huge the number may be) is that the value contributed by a symbol depends on its position. The symbol is multiplied by its positional value.

In Computer Science instead of using decimal numbers, we prefer to deal with binary number. The only symbols in binary are: 0 and 1 forming a base-2 numeral system where each position has also a positional value. Consequently, a binary number is represented in a row of bits that could have a value of 0 or 1. It is assigned to each bit a position number, ranging from zero to N-1, where N is the number of bits in the binary representation used. Usually, this is simply the exponent for the corresponding bit weight in base-2 (such as in $2^{31}..2^{0}$).



Consider the decimal number 42, as an example, its representation in binary is 101010. Meaning that the position of 2^5 is 1, 2^4 is 0,2^3 is 1, 2^2 is 0, 2^1 is 1 and 2^0 is 0. So, the exponent of the highest power of 2 is 5 (2^5).

The same approach is followed by any base-n numeral system.

ASCII table

ASCII stands for American Standard Code for Information Interchange. This is a character encoding standard for electronic communication. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'A' or '+' or an action of some sort. The ASCII table defines the codes that represent text in computers.

Ascii	Char	Ascii	Char	Ascii	Char	Ascii	Char
0	Null	32	Space	64	@	96	`
1	Start of heading	33	!	65	A	97	a
2	Start of text	34	"	66	B	98	b
3	End of text	35	#	67	C	99	c
4	End of transmit	36	$	68	D	100	d
5	Enquiry	37	%	69	E	101	e
6	Acknowledge	38	&	70	F	102	f
7	Audible bell	39	'	71	G	103	g
8	Backspace	40	(	72	H	104	h
9	Horizontal tab	41	)	73	I	105	i
10	Line feed	42	*	74	J	106	j
11	Vertical tab	43	+	75	K	107	k
12	Form feed	44	,	76	L	108	l
13	Carriage return	45	-	77	M	109	m
14	Shift in	46	.	78	N	110	n
15	Shift out	47	/	79	O	111	o
16	Data link escape	48	0	80	P	112	p
17	Device control 1	49	1	81	Q	113	q
18	Device control 2	50	2	82	R	114	r
19	Device control 3	51	3	83	S	115	s
20	Device control 4	52	4	84	T	116	t
21	Neg. acknowledge	53	5	85	U	117	u
22	Synchronous idle	54	6	86	V	118	v
23	End trans. block	55	7	87	W	119	w
24	Cancel	56	8	88	X	120	x
25	End of medium	57	9	89	Y	121	y
26	Substitution	58	:	90	Z	122	z
27	Escape	59	;	91	[	123	{
28	File separator	60	<	92	\	124	|
29	Group separator	61	=	93	]	125	}
30	Record separator	62	>	94	^	126	~
31	Unit separator	63	?	95	_	127	Forward del.

Do your best to solve the problems, don't give up but if you get stuck in a problem, go to the following! We wish you the best of luck!

This is an example problem with a proposed solution for the different programming languages supported in the contest:

Let's begin introducing yourselves

Introduction

You'll have no chance to win at HP CodeWars (or life) if you don't know how to do Input and Output properly. You also won't do well at CodeWars if you are rude to your judges. Write a program to greet your esteemed judges appropriately. Read in the name of a judge and output your greeting in the appropriate format. If you're confused at this point, go back and re-read your contest instructions.

Input

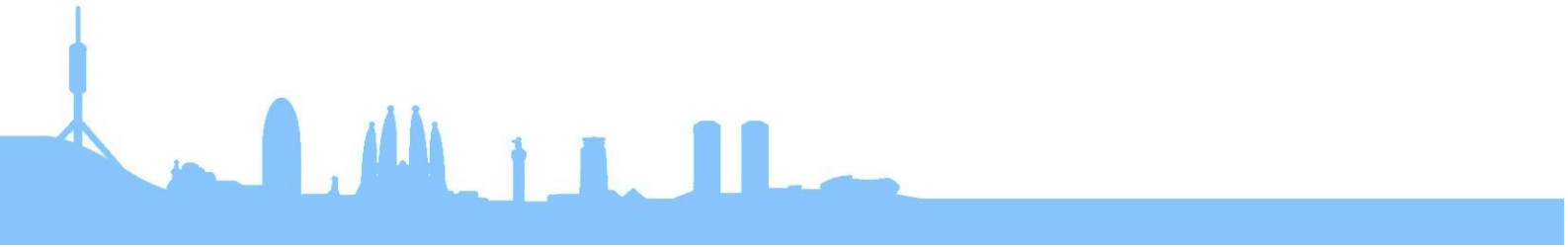
The input will be your judge's first name, a single word with no spaces.

```
Simon
```

Output

Welcome your judge with a friendly greeting.

```
Greetings, Simon the Great! I genuflect in your general direction!
```

Python 3 language implementation

```
# input data
judge_name = input()

# output data
print ("Greetings," + judge_name + "the Great! I genuflect in your general direction!")
```

Java language implementation

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        String judge_name;

        // input data
        Scanner read=new Scanner(System.in);
        judge_name=read.next();

        //output data
        System.out.println("Greetings, " + judge_name + " the Great! I genuflect in
your general direction!");
    }

}
```

C language implementation

```
#include <stdio.h>

void main()
{
    char judge_name[32];

    /* input data */
    scanf("%s", judge_name );

    /* output data */
    printf("Greetings, %s the Great! I genuflect in your general direction!",
           judge_name);
}
```

C++ language implementation

```
#include <iostream>

using namespace std;

int main()
{
    string judge_name;

    // input data
    cin >> judge_name;

    //output data
    cout << "Greetings, " << judge_name << " the Great! I genuflect in your general direction!" << endl;
}
```

A Where no man has gone before

1 point

NOTE: This is the first of the two problems that can be solved and submitted before the start of the Code *Wars* competition. Teams are **strongly** encouraged to submit these problems **prior** to the start of the competition – hey, it's basically a free point!

Introduction

After another complete orbit around the Sun, here we are ready again for a new edition HP CodeWars. Feeling like if we were in a mission of the starship Enterprise, repeat with us the phrase made popular through the title sequence of the original Star Trek science fiction television series:

"To boldly go where no man has gone before!"

Input

There is no input for this program.

Output

The program must output the phrase:

```
To boldly go where no man has gone before!
```

B Welcome message

1 point

NOTE: This is the second of the two problems that can be solved and submitted before the start of the Code *Wars* competition. Teams are **strongly** encouraged to submit these problems **prior** to the start of the competition – hey, it's basically a free point!

Introduction

When setting up the HP CodeWars contest, year after year, the organization updates a lot of documentation with the current year. To make things easier you can help to automate this task.

Write a simple program that reads the number corresponding to current year and prints the HP CodeWars welcome message adding the year.

Input

The input consists of a single number representing the current year.

Output

The program must output the message Welcome to HP CodeWars adding the year read.

Example

Input

```
2022
```

Output

```
Welcome to HP CodeWars 2022
```