LAB14-05 – Guess the Code

The program will grab a number within 1~1000 randomly (here we call it **the code**) and start the game. One will have 10 chances to guess the code. Once a guess match the code, the game ends. Otherwise, one fails with 10 miss guesses. You will have to judge whether the game ends for each guess. If not, update the range for the next guess.

Try to do this by using functions that define by yourself. You will have to use the function **srand()** and **rand()** to implement this program.

Input

The inputs will be the seeds of the random function, which will be used in "srand()". With the first input of each round, your program will run the function rand() 10 times to get the mystery code within 1~1000. After the code is determined, you will get new inputs as seeds and take the 10-th random numbers as the guesses.

Output

For each round, you will have to output the numbers you guess. Once the code is matched, output "**win(the code, how many times you guess)**", otherwise, output "**fail(the code)**".

Sample inputs and Outputs

Round 1											
Seeds	4	10	20	3	7	16	9	2	100	1	4
Guesses	(21)	900	562	388	221	188	7	144	61	39	14
Ranges	[1,1000]	[1,900)	[1,562)	[1,388)	[1,221)	[1,188)	(7,188)	(7,144)	(7,61)	(7,39)	(14,39)
Inputs	4 10 20 3 7 16 9 2 100 1 4										
Outputs	900 562 388 221 188 7 144 61 39 14 fail(21)										
Round 2											
Seeds	4	9	2	78	0	20	1	2	2		
Guesses	(21)	48	34	15	28	19	20	25	21		
Ranges	[1,1000]	[1,48)	[1,34)	(15,34)	(15,28)	(19,28)	(20,28)	(10,25)	match		
Inputs	4 9 2 78 0 20 1 2 2										
Outputs	48 34 15 28 19 20 25 21 win(21,8)										

More Explanations and Hints

The code is 21 for both sample rounds above. 21 is the 10-th number under seed=4 while executing the function **rand()** recursively. In this game, it works the same for the guesses. For example, the first guess in Round 1, 900, is the 10-th random number under seed=10.

Try to understand **how to set the range of generating a random number**, then this work will be a piece of cake.