The 43rd Annual ACM

International Collegiate Programming Contest

Asia Regional – Seoul

Nationwide Internet Competition

Practice Problem B

Happy Number

Time Limit: 0.2 Seconds

Consider the following function f defined for any natural number n:

$f(n)$ is the number obtained by summing up the squares of the digits of n in decimal (or base-ten).

If $n = 19$, for example, then $f(19) = 82$ because $1^2 + 9^2 = 82$.

Repeatedly applying this function f, some natural numbers eventually become 1. Such numbers are called *happy numbers*. For example, 19 is a happy number, because repeatedly applying function f to 19 results in:

$$f(19) = 1^2 + 9^2 = 82$$
$$f(82) = 8^2 + 2^2 = 68$$
$$f(68) = 6^2 + 8^2 = 100$$
$$f(100) = 1^2 + 0^2 + 0^2 = 1$$

However, not all natural numbers are happy. You could try 5 and you will see that 5 is not a happy number. If n is not a happy number, it has been proved by mathematicians that repeatedly applying function f to n reaches the following cycle:

$$4 \to 16 \to 37 \to 58 \to 89 \to 145 \to 42 \to 20 \to 4.$$

Write a program that decides if a given natural number n is a happy number or not.

Input

Your program is to read from standard input. The input consists of a single line that contains an integer, n $(1 \le n \le 1{,}000{,}000{,}000)$

Output

Your program is to write to standard output. Print exactly one line. If the given number n is a happy number, print out `HAPPY`; otherwise, print out `UNHAPPY`.

The following shows sample input and output for two test cases.

Sample Input 1	Output for the Sample Input 1
19	HAPPY

Sample Input 2	Output for the Sample Input 2
5	UNHAPPY