



# Programación Competitiva

**B.S. Rodolfo Mercado Gonzales**  
**Universidad Nacional de Ingeniería**

# Programación Competitiva UNI

Capacitamos a los estudiantes en algoritmos y estructuras de datos a nivel avanzado, con el objetivo de ser líderes en programación competitiva.



Facebook: ProgramacionCompetitivaUNI

# Programación Competitiva

Consiste en resolver “lo antes posible” problemas de índole algorítmico, lógico y matemático, a través de programas computacionales, teniendo en cuenta los siguientes recursos :

- ❑ Tiempo de ejecución del programa
- ❑ Memoria usada por el programa



# Tiempo de ejecución

- ❑ Es el tiempo que le toma a un programa procesar una determinada entrada.
- ❑ En las competencias hay un límite para el tiempo de ejecución, generalmente es **1 segundo**.

|                                |         |          |
|--------------------------------|---------|----------|
| Time limit exceeded on test 12 | 2000 ms | 16200 KB |
|--------------------------------|---------|----------|

# Memoria

- ❑ Las variables usadas en un programa son las que principalmente ocupan espacio en memoria.
- ❑ En las competencias hay un límite para el uso de memoria, generalmente es **256 MB**.

|                                 |       |           |
|---------------------------------|-------|-----------|
| Memory limit exceeded on test 1 | 93 ms | 262100 KB |
|---------------------------------|-------|-----------|

# Programación Competitiva

El objetivo es formar profesionales que estén preparados para producir mejor software y enfrentarse en el futuro a problemas de investigación.

## Fundadores ACM-ICPC



# ¿Cómo son los problemas?

- ❑ Nos enfrentamos a problemas que involucran algoritmos, lógica y matemática, que previamente han sido resueltos al menos por el autor.
- ❑ “No son imposibles de resolver”.

# ¿Cómo son los problemas?

## A. Watermelon

time limit per test: 1 second

memory limit per test: 64 megabytes

input: standard input

output: standard output

One hot summer day Pete and his friend Billy 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.

Pete and Billy 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. The boys 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

The first (and the only) input line contains integer number  $w$  ( $1 \leq w \leq 100$ ) — the weight of the watermelon bought by the boys.

### Output

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.

### Examples

|               |      |
|---------------|------|
| <b>input</b>  | Copy |
| 8             |      |
| <b>output</b> | Copy |
| YES           |      |



# ¿Cómo afrontar un problema?

## Método Pólya

- ❑ **Entender el problema :** identificar los datos de entrada, salida y las restricciones.
- ❑ **Diseñar un plan:** recordar problemas similares, partir de la solución “más ingenua”, intentar con los algoritmos que conocemos, resolver casos particulares/generales/parte del problema.
- ❑ **Ejecutar el plan:** comprobar que nuestra estrategia es correcta (es posible cambiar de estrategia).
- ❑ **Examinar la solución:** verificar qué funcionó y qué no en nuestro plan, para mejorar ante nuevos problemas.

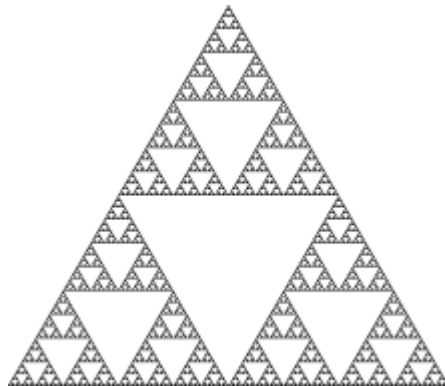
# ¿Qué conocimientos debo tener/adquirir?

- ☐ Razonamiento lógico-matemático.
- ☐ **Algoritmos y estructuras de datos.**
- ☐ Un lenguaje de programación a nivel básico.



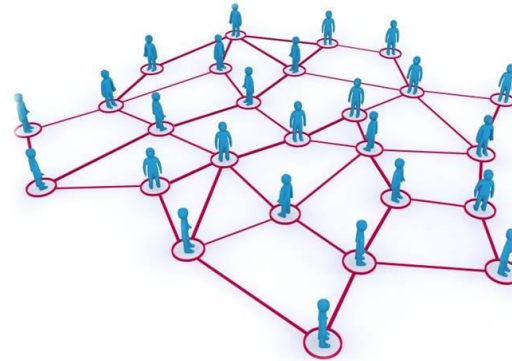
# Algoritmos y Estructuras de Datos

- ☐ Análisis de Algoritmos
- ☐ Standard Template Library
- ☐ Fuerza Bruta
- ☐ Recursividad
- ☐ Backtracking
- ☐ Divide y Vencerás
- ☐ Búsqueda Binaria
- ☐ Programación Dinámica



# Algoritmos y Estructuras de Datos

- ❑ Teoría de números
- ❑ Geometría Computacional
- ❑ Grafos
- ❑ Estructuras de Datos Avanzadas
- ❑ Procesamiento de Cadenas
- ❑ Teoría de Juegos



# Competencias de programación

- ❑ Las competencias consisten en resolver un conjunto de problemas algorítmicos en un determinado tiempo.
- ❑ Gana quien resuelva la mayor cantidad de problemas. En caso de empate, quien tuvo menor penalidad será el vencedor.



# Evaluación de una solución

1. Se envía el código fuente a un juez online.



# Evaluación de una solución

2. Este juez se encargará de testear tu “solución” con un input secreto.

| Who             | Problem                                   | Lang      | Verdict           |
|-----------------|---|-----------|-------------------|
| system_1        | <a href="#">911D - Inversion Counting</a> | GNU C++   | Running on test 1 |
| army_of_one     | <a href="#">459A - Pashmak and Garden</a> | GNU C++11 | Running on test 6 |
| manojkannekanti | <a href="#">911A - Nearest Minimums</a>   | GNU C++14 | Running on test 4 |

# Evaluación de una solución

3. En unos segundos el juez dará un veredicto para tu solución.

AC (accepted ) – Tu solución es correcta.

WA (wrong answer) – Tu programa da una respuesta incorrecta.

TLE (time limit exceeded ) – Tu programa tarda mucho tiempo.

MLE (memory limit exceeded ) – Tu programa usa demasiada memoria.

RE (runtime error) – Tu programa se cae durante su ejecución.

CE (compilation error) – Tu programa no compila.



# Concursos en equipo



**IEEE**XTREME  
**PROGRAMMING**  
COMPETITION

# ACM-ICPC



- ❑ Competencia organizada por la ACM desde 1970.
- ❑ Compiten equipos de 3 estudiantes.
- ❑ Consta de 2 fases de manera presencial, cada una de 5 horas :
  - **Regional Sudamérica/Sur:** Argentina, Perú, Chile, Bolivia, Uruguay y Paraguay (3 cupos para el mundial)
  - **Final Mundial :** todos los equipos clasificados de las distintas regionales.

# Regionales



# Mundiales



Varsovia 2012



San Petersburgo 2013



Phuket 2016

# IEEExtreme



- ❑ Competencia organizada por la IEEE desde 2006.
- ❑ Compiten equipos de 3 estudiantes.
- ❑ Consta de una sola fase online.
- ❑ El concurso dura 24 horas.



# IEEExtreme

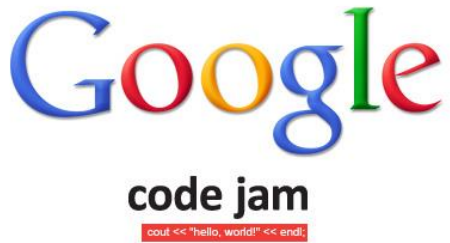


**2013 - 1° puesto**

**2012 - 2° puesto**

**2011 - 4° puesto**

# Concursos individuales



# Beneficios de competir

1. Dominarás algoritmos y estructuras de datos.
2. Aprenderás C++ y Python.
3. Mejorarás tus habilidades para la resolución de problemas.
4. Trabajarás en equipo y bajo presión.

```
#include <bits/stdc++.h>
using namespace std;

struct Tuple{
    int gcd, x, y;
    Tuple( int a, int b, int c ){
        gcd = a, x = b, y = c;
    }
};

Tuple extGcd( int a, int b ){
    if ( b == 0 ) return Tuple( a, 1, 0 );
    Tuple ret = extGcd( b, a % b );
    return Tuple( ret.gcd, ret.y, ret.x - a/b * ret.y );
}

int main( ){
    Tuple ans = extGcd( 3, 5 );
    cout << ans.x << " " << ans.y << endl;
}
```



# Beneficios de competir

5. Podrás participar en congresos nacionales.

Iquitos



# Beneficios de competir

6. Podrás participar en campamentos de programación.

Argentina



# Beneficios de competir

7. Tendrás muchas ofertas laborales a nivel nacional.



# Beneficios de competir

8. Incrementarás tus posibilidades de trabajar en las mejores empresas a nivel mundial.



# Beneficios de competir

9. Incrementarás tus posibilidades de hacer estudios en el extranjero.



# Beneficios de competir

10. Podrás ganar muchos premios.

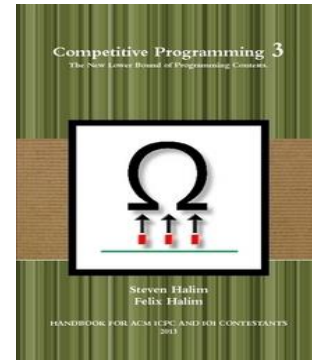
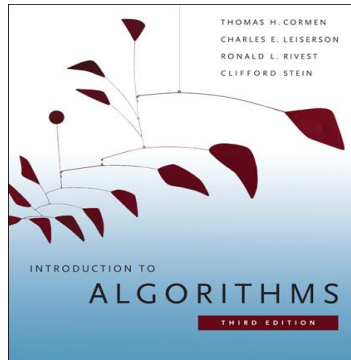


# Tips para ser competitivo

- ❑ Domina tu lenguaje de programación preferido.
- ❑ Tipea rápido ([typingtest](#)).
- ❑ Testea tu solución.
- ❑ Practica a full.
- ❑ Trabaja en equipo

# Referencias

- ❑ Thomas Cormen et al. - **Introduction to Algorithms**
- ❑ E-Maxx - **E-Maxx Algorithms**
- ❑ Steven Halim & Felix Halim - **Competitive Programming 3**





¡ Good luck and have fun !