CURSO DE PROGRAMACIÓN COMPETITIVA
URJC - 2019

CURSO DE PROGRAMACIÓN COMPETITIVA
URJC - 2019

Curso:

- 14 sesiones
- 18 de enero - 26 abril (inclusive)
- Viernes: 17:00 - 19:00

Organizadores:

- David Morán (ddavidmorang@gmail.com)
- Sergio Pérez (sergio.perez.pelo@urjc.es)
- Juan Quintana (juandavid.quintana@urjc.es)
- Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)

CURSO DE PROGRAMACIÓN COMPETITIVA URJC - 2019

Fechas importantes:

- Festivo: viernes 19 de abril (sin sesión)
- Google Hashcode: 28 de febrero
- Ada Byron: viernes 8 y sábado 9 de marzo
- SWERC 2019 (octubre-noviembre?)
  - Clasificación equipos URJC el 26/04/19 (última clase)

Motivación

- Programming skills
  - diseño de algoritmos
  - estructuras de datos
  - nociones de complejidad
  - …aprobar asignaturas!!! (ED, EDA, DAA, …)

Motivación

- Empresas patrocinadoras
  - cazatalentos
  - concursos internos
  - entrevistas de trabajo

Motivación

- Participación en concursos
  - SWERC ‘19 en París: 3 equipos
  - AdaByron
  - Google Hashcode
  - 12 Uvas
  - … premios y honor!!!


Les Couleurs
de la vie
UNIVERSIDAD
URJC
ETSII

OBJETIVOS EN PROGRAMACIÓN COMPETITIVA

- Resolver los problemas en el menor tiempo posible
- Tener nociones intuitivas:
  - Tipos de problemas, algoritmos...
  - Complejidad vs límite de tiempo (eficiencia)
  - Estructuras de datos necesarias
- Trabajo en equipo (nombres creativos)
- Representar tu institución, país...

PLANIFICACIÓN DEL CURSO

- Cada bloque se dividirá en dos sesiones; una teórica y una práctica
- En la teórica, los docentes explicarán los algoritmos y los diferentes usos, así como resolver algún problema de ejemplo tipo
- En la práctica, se levantará un juez y se hará un concurso de prueba con enunciados cortos para fortalecer lo dado la semana anterior

PLANIFICACIÓN DEL CURSO

Objetivos:

- Que los alumnos se sientan cómodos en un ambiente competitivo
- Que se familiaricen con los jueces (y a ser posible con sistemas UNIX)
- Que entrenen de manera dinámica sus habilidades de programación

PLANIFICACIÓN DEL CURSO

- Bloque 1: Introducción (18/01, 25/01)
- Bloque 2: Estructuras de Datos (01/02, 08/02)
- Bloque 3: Algoritmos de búsqueda y voraces (15/02, 22/02)
- Bloque 4: Grafos (01/03, 08/03)
- Bloque 5: Grafos ponderados (15/03, 22/03)
- Bloque 6: Programación Dinámica(introducción) (29/03)
- Bloque 7: Programación Dinámica (05/04, 12/04)
- Eliminatorias SWERC 2019 (26/04)

TIPOS DE COMPETICIONES

- ACM-ICPC:
  - 5 horas de duración
  - Equipos: 3 personas (1 ordenador)
  - Puntuación: problemas resueltos (0/1)
    - Empates: tiempo + penalizaciones

TIPOS DE COMPETICIONES

RANK	TEAM	SCORE		A	B	C	D	E	F	G	H	I	J	K
1	**ENS Ulm 1** ENS Paris	**10**	1526	1/33	2/139	0	1/39	1/59	1/287	4/235	1/299	1/100	1/154	1/101
2	**Team RaclETH** ETH Zürich	**9**	888	1/14	1/56	0	1/38	1/67	1/83	5/278	2/118	1/117	4	1/17
3	**UPC-1** Universitat Politècnica de Catalunya	**9**	1421	2/9	1/29	0	2/61	1/82	3/147	3/274	1/196	1/241	0	3/222
4	**SNS 1** Scuola Normale Superiore	**8**	803	1/13	2/70	1	4/113	1/23	1/107	0	2/147	1/152	0	1/78
5	**illUSIon** Università della Svizzera italiana	**8**	986	1/9	2/103	0	2/66	1/80	2/157	0	1/249	2/206	1	1/36
6	**UPC-2** Universitat Politècnica de Catalunya	**8**	990	2/14	4/89	0	1/112	1/25	1/128	0	3/196	1/259	4	1/47
7	**EP red** École Polytechnique	**7**	1092	1/17	2/210	0	5/123	1/79	1/240	0	0	2/166	0	1/137
8	**UPC-3** Universitat Politècnica de Catalunya	**7**	1168	1/12	1/231	0	1/97	1/59	2/170	0	0	5/282	0	2/197
9	**Moradonellani** Politecnico di Milano	**7**	1181	1/16	1/102	0	2/50	3/78	5/285	0	0	2/223	0	1/267
10	**SNS 2** Scuola Normale Superiore	**7**	1319	3/69	1/175	0	2/107	2/140	1/236	0	2	1/295	0	2/197

https://swerc.eu/2018/theme/scoreboard/public/

TIPOS DE COMPETICIONES

- ACM-ICPC (Proceso de selección)
  - Eliminatorias en la universidad si hay más de tres equipos
  - Eliminatorias en el conjunto de países que forman una región (South-Western Europe)
  - Eliminatorias entre los potenciales candidatos en todo el continente (Super regional europeo (Beta)
  - Final Mundial

TIPOS DE COMPETICIONES

- Codeforces y Topcoder
  - Concursos muy rápidos y frecuentes
  - Libre para cualquiera
  - Dos o tres divisiones para novatos y expertos
  - De 95 a 120 minutos de duración
  - Puedes ver y ‘romper’ el código de otros
  - Sistema de puntuación (mientras más tardes en resolver problemas, más te penalizan en puntos)

TIPOS DE COMPETICIONES

- Facebook Hacker Cup y Google Code Jam
  - Evento de gente masiva online
  - Al menos 4 rondas
  - Suele haber ronda de clasificación, 2 rondas de filtro y luego la fase final
  - Dos tipos de evaluación (small y large)
  - El caso small se corrige automáticamente
  - El caso large se corrige al terminar la competición
  - Se permite cualquier tipo de solución (incluso manual ó *hardcodeada*) que permita llegar al output

TIPOS DE COMPETICIONES

- USACO/COCI/IOI
  - Concursos dirigidos a alumnos de bachiller/secundaria
  - ¡NO SON TAN FÁCILES!
  - Son evaluados con sistemas de puntuación (no binario ni penalizando tiempo de solución)
  - Resultados después de la competición
  - Funcionan por temporadas (de noviembre a abril) por ser eliminatorias para el IOI (International Olympiads in Informatics)

CARACTERÍSTICAS DE UN PROBLEMA

Enunciado: Se explica el problema con una narración que lo justifica
Análisis del Problema: Se requiere una solución determinista para el problema (siempre encontraremos una solución óptima y válida)
Entrada: Se especifica lo que nuestro programa debe leer
Salida: Se especifica lo que nuestro programa debe mostrar
Ejemplos I/O: Muestras de entrada/salida con el comportamiento esperado para el programa
Límites [Opcionales]: Lo máximo ó mínimo en cuanto a variables que nuestro programa debe tomar en cuenta

CARACTERÍSTICAS DE UN PROBLEMA

- Tipos de Lectura:
  - Un caso: Se lee un caso de prueba y a partir de la entrada se genera una salida y termina la ejecución
  - Múltiples casos: Se leen varios casos de pruebas y, dadas múltiples entradas, se generan múltiples salidas
- ¡Cuidado con reutilizar estructuras de datos!
- No hace falta guardar todos los resultados y mostrarlos al final

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA DE T CASOS

Se recibe un entero T y luego vendrán T casos de prueba

```
Scanner sc = new Scanner(System.in);
int t = sc.nextInt();
for(int i=0; i<t; i++) {
    //- leer datos de cada caso
    //- codigo + generar salida
}
```

```
int t;
scanf("%d", &t);
for(int i=0; i<t; i++){
    // - leer datos de cada caso
    // - codigo + generar salida
}
```

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA HASTA EOF

Se leen los casos hasta leer la marca EOF (End-Of-File)

```java
Scanner sc = new Scanner(System.in);
while(sc.hasNextInt()) {
    int n = sc.nextInt();
    // codigo
}
```

```c
int n;
while(scanf("%d", &n) != EOF){
    //codigo
}
```

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA HASTA CASO EN 0

Se lee el número de casos hasta que se consiga una condición de parada (generalmente cuando la entrada sea 0)

```
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
while(n!=0) {
    //codigo
    n = sc.nextInt();
}
```

```
int n;
while(scanf("%d", &n)!= EOF && n!=0){
    //codigo
}
```

LENGUAJES DE PROGRAMACIÓN

- C
- C++
- Java
- Python (muy nuevo)

Ejemplo Problema: ¡Hola mundo!

https://www.aceptaelreto.com/problem/statement.php?id=116

¡Hola mundo!

Tiempo máximo: 1,000 s Memoria máxima: 2048 KiB

Dicen los viejos que en este país el latín era una asignatura obligatoria por la que todos los estudiantes de bachillerato tenían que pasar.

Dicen los viejos que el primer día de clase de latín cualquiera esperaba que los alumnos salieran sabiendo el "rosa rosae".

Dicen los viejos que esa era la primera declinación.

Quizá, dentro de muchos años, nosotros seamos los viejos que contemos batallitas de cómo se enseñaba informática. Y quizá entonces, digamos que en la primera clase de cualquier curso en el que se explicara un lenguaje de programación, se tenía que salir habiendo escrito "un hola mundo".

Y eso es lo que vamos a hacer. Escribir un programa que escriba tantos "hola mundo" como nos pidan.

Entrada

La entrada consta de una única línea que contiene un número n, $0 \leq n \leq 5$, que indica cuántos mensajes hay que emitir.

Salida

Cada mensaje a escribir aparecerá en una única línea y será la cadena "`Hola mundo.`".

Entrada de ejemplo

```
3
```

Salida de ejemplo

```
Hola mundo.
Hola mundo.
Hola mundo.
```

Ejemplo Problema: Test

https://www.spoj.com/problems/TEST/

TEST - Life, the Universe, and Everything

#basic #tutorial #ad-hoc-1

Your program is to use the brute-force approach in order to *find the Answer to Life, the Universe, and Everything*. More precisely... rewrite small numbers from input to output. Stop processing input after reading in the number 42. All numbers at input are integers of one or two digits.

Example

```
Input:
1
2
88
42
99

Output:
1
2
88
```

Registrarse en el entrenamiento

- Enlace: http://programming-urjc.herokuapp.com
- Ejercicios de SPOJ y AceptaElReto divididos por categorías
- Registro mediante bot de Telegram
- Validados solo por correo de la URJC (@alumnos.urjc.es)

Semana que viene

- Primeros problemas de programación competitiva
- Nos vemos en el laboratorio (Aula por decidir)
- Portátil Opcional

Hasta la vista

Baby