

CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023



CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023

Organizadores:

- **Isaac Lozano** (isaac.lozano@urjc.es)
- Raúl Martín (raul.martin@urjc.es)
- **Sergio Salazar** (s.salazarc.2018@alumnos.urjc.es)
- Francisco Tórtola (f.tortola.2018@alumnos.urjc.es)
- Cristian Pérez (c.perezc.2018@alumnos.urjc.es)
- Xuqiang Liu (x.liu1.2020@alumnos.urjc.es)
- Alicia Pina (a.pinaz.2020@alumnos.urjc.es)
- Sara García (s.garciarod.2020@alumnos.urjc.es)
- Raúl Fauste (r.fauste.2020@alumnos.urjc.es)



CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023

¿Quién somos?



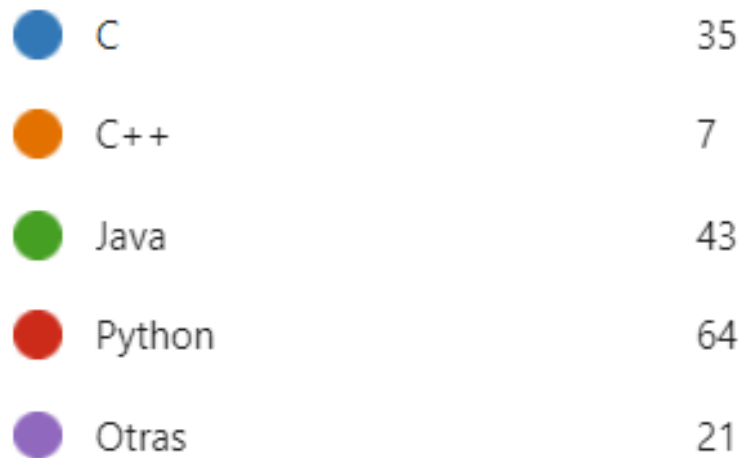
CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023

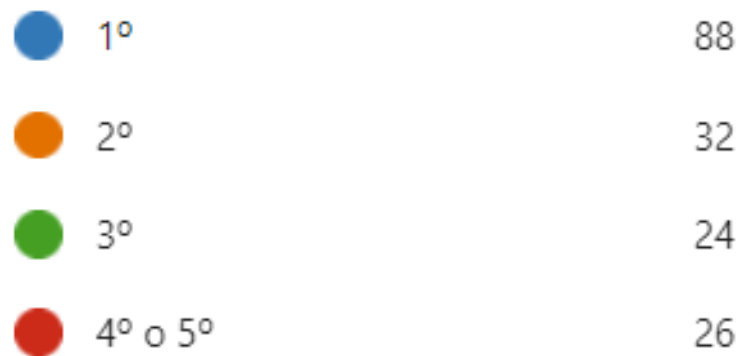
¿En que trabajamos?



¿Lenguajes?

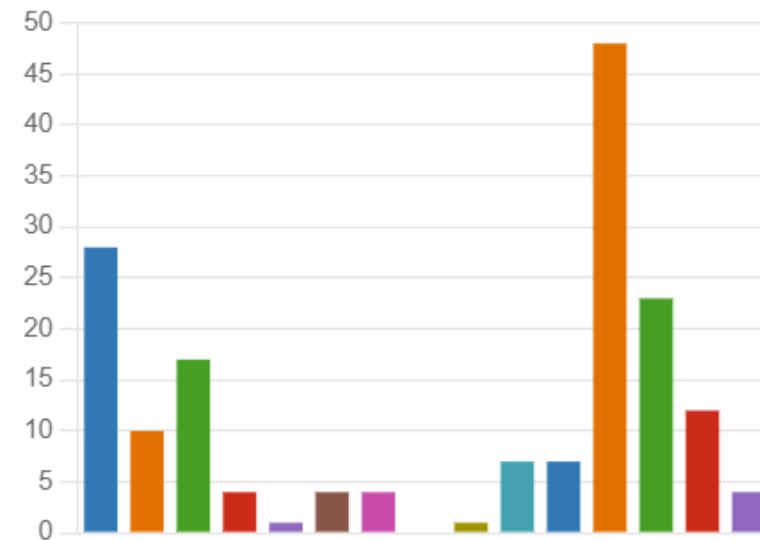


¿Cursos?



¿Carreras?

● Ingeniería Informática	28
● Ingeniería de Computadores	10
● Ingeniería del Software	17
● Diseño y Desarrollo de Videojue...	4
● Matemáticas	1
● Diseño y Desarrollo de Videojue...	4
● Ingeniería del Software + Mate...	4
● Ingeniería Informática + Admini...	0
● Ingeniería Informática + Ingenie...	1
● Ingeniería Informática + Ingenie...	7
● Ingeniería Informática + Matem...	7
● Ingeniería de la Ciberseguridad	48
● Inteligencia Artificial	23
● Ciencia e Ingeniería de Datos	12
● Otras	4



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



¿EMPRESAS QUE TRABAJÓ/A GENTE QUE COMPLETÓ EL CURSO?

amazon



accenture



IBM



Deloitte.

Telefonica



gmw
INNOVATING SOLUTIONS



Motivación

- Participación en concursos
 - SWERC '24 en ????: ? equipos
 - AdaByron '23 -> **10/03** y 07/07
 - Google Hashcode -> 13/04
 - 12 Uvas -> 31/12
 - ... premios y honor!!!





CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023

Curso:

- 11 sesiones
- 3 de febrero - 28 de abril (inclusive)
- Viernes: 17:00 - 19:00
- Aula 001 – Aulario 2 (Teoría)
- Aular 109/111 - Laboratorios 3 (Práctica)



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



CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2023

Fechas importantes:

- El 14/02 no existe debido al viaje a Milán para participar al SWERC

UNIVERSIDAD REY JUAN CARLOS

Historical standings

- Hackbulla
 - Marcelino Siles Rubia
 - Hugo Leal Vara
 - Adán El Yaacoubi
 - Raúl Martín-Santamaría (coach)
- Teamto de Verano
 - Cristian Pérez
 - Sergio Salazar
 - Francisco Tórtola
 - Isaac Lozano Osorio (coach)
- pik2s
 - Raul Fauste Jimenez
 - Sara García
 - Alicia Pina
 - Isaac Lozano Osorio (coach)



CLASIFICATORIO ADABYRON 2023 - URJC

Sesión 24/02 Concurso por Equipos

17:00 a 20:00

Máximo 7 clasificados asegurados 1 por categoría

<https://ada-byron.es/2023/reg/madrid/locales/urjc/>



PLANIFICACIÓN DEL CURSO

- Bloque 1: Introducción (03/02)
- Bloque 2: Estructuras de Datos (10/02, 24/02)
- Bloque 3: Algoritmos de búsqueda y voraces (03/03, 10/03)
- Bloque 4: Grafos (17/04, 24/03)
- Bloque 5: Grafos ponderados (31/03, 14/04)
- Bloque 6: Programación Dinámica (21/04)
- Concurso Final (28/04)

Los concursos tendrán premios relacionados con merchand de la universidad. No sabemos cantidad pero aseguramos mínimo 1 premio no acumulativo (si alguien ganó, pasará al siguiente)



Sesión 10/03 Concurso Individual ONLINE

17:00 a 19:00

Mismos problemas que en el entrenamiento del AdaByron



CANALES Y FORMULARIO DE EQUIPOS



@URJC_CP



@Dijkstraideos



Formulario



NO TENGO EQUIPO!!

Los concursos oficiales FUERZAN que sean 3 personas, en caso de ser menos o más no permiten participar.

Usad la clase para buscar otros compañeros.
Usad los grupos.

Preguntadnos, pero ser 3 para poder participar en cualquier evento.



¿Y MIS CRÉDITOS?

- 11 sesiones - asistir a MÍNIMO 8 de ellas = CRÉDITOS
- Concursos (HashCode, AdaByron, SWERC) = CRÉDITOS
- Voluntarios en concurso = CRÉDITOS



¿CÓMO SE VALIDA ASISTENCIA?

- Sesiones teóricas: Aplicación de la URJC de Asistencia en un momento aleatorio de la clase. En caso de fallar, formulario de asistencia.
- Sesiones prácticas: Mínimo de 1 envío por persona/equipo revisando IP del envío + Aplicación URJC en un momento aleatorio.
- No rellenar el formulario o envío = NO ASISTENCIA



¿Podemos preguntar cualquier duda?

- Sí y no..



PLANIFICACIÓN DEL CURSO

Objetivos:

- Sentirse cómodo en un ambiente competitivo
- Familiarizarse con los jueces (y a ser posible con sistemas UNIX)
- Entrenar de manera dinámica sus habilidades de programación



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...



TIPOS DE COMPETICIONES































[ADA BYRON](#)



- 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	L
1	  Raw Pots Harbour.Space University	10 1303	4 1 try		274 6 tries	59 2 tries	114 1 try	14 1 try		249 1 try	243 3 tries	15 2 tries	84 1 try	67 1 try
2	  TAU++ Tel Aviv University	10 1327	6 1 try	196 2 tries	295 7 tries	23 1 try	70 2 tries	29 1 try	1 try		262 2 tries	34 1 try	103 1 try	129 1 try
3	  TempName Tel Aviv University	9 1043	6 1 try		95 3 tries	42 1 try	76 1 try	9 1 try		2 tries	141 1 try	26 2 tries	274 3 tries	234 3 tries
4	  kempirqosaq Harbour.Space University	9 1082	8 1 try	3 tries	161 3 tries	54 1 try	229 1 try	11 1 try		1 try	208 2 tries	65 1 try	23 1 try	203 4 tries
5	  ENS Ulm 1 École Normale Supérieure de Paris	9 1146	6 1 try		154 3 tries	143 4 tries	37 1 try	19 1 try		122 1 try	1 try	48 1 try	228 5 tries	209 1 try
6	  mETHdical ETH Zürich	9 1306	7 1 try		183 1 try	67 1 try	107 2 tries	19 1 try			262 2 tries	41 2 tries	226 1 try	254 5 tries
7	  EP Oranje Ecole Polytechnique	8 883	14 2 tries		118 1 try	65 1 try	200 2 tries	32 1 try			223 1 try	26 2 tries	3 tries	145 1 try
8	  UPC-2 Universitat Politècnica de Catalunya	8 957	7 1 try		287 4 tries	57 1 try	95 2 tries	43 1 try				44 1 try	206 1 try	138 1 try
9	  ENS Ulm 3 École Normale Supérieure de Paris	8 1007	10 1 try		187 2 tries	70 1 try	96 1 try	33 1 try			296 4 tries	80 3 tries		115 1 try
10	  SGSS University of Padova	8 1018	5 1 try		239 2 tries	67 3 tries	169 1 try	18 2 tries			178 2 tries	49 3 tries		133 2 tries
11	  UPC-1 Universitat Politècnica de Catalunya	8 1349	5 1 try		294 5 tries	79 1 try	214 5 tries	18 2 tries				33 2 tries	270 5 tries	156 1 try
12	  Outlaws Instituto Superior Técnico	7 742	13 1 try		260 1 try	43 1 try	155 1 try	33 1 try				59 1 try		179 1 try
13	  UPC-3 Universitat Politècnica de Catalunya	7 748	6 1 try		241 1 try	89 5 tries	121 2 tries	16 1 try				23 1 try		152 1 try
14	  EP Funk Ecole Polytechnique	7 778	12 1 try		1 try	166 1 try	138 2 tries	18 1 try	287 2 tries	2 tries		39 1 try		78 1 try
15	  Not Last Place Ecole Polytechnique Fédérale de Lausanne	7 1014	7 1 try		228 5 tries	119 3 tries	157 1 try	19 1 try				36 2 tries		288 2 tries

<https://swerc.eu/2021/theme/scoreboard/public/>



MARCADOR DESCONGELADO SESIÓN DE PRUEBA

RANK	TEAM	SCORE	A	B	C	D	E
1	 de.orna.2020 Universidad Rey Juan Carlos	5 18155	992 1 try	1004 1 try	3842 2 tries	5391 4 tries	6826 2 tries
2	ENIGMA	5 28044	6980 1 try	6987 1 try	6994 1 try	7005 1 try	7048 1 try
3	 e.gomezf.2020 Universidad Rey Juan Carlos	5 29444	5190 1 try	5368 3 tries	5208 1 try	6843 3 tries	6755 1 try
4	 a.dekeno.2020 Universidad Rey Juan Carlos	4 10898	100 1 try	5217 14 tries	138 1 try	5043 8 tries	5 tries
5	 s.ramosg.2018 Universidad Rey Juan Carlos	3 7557	751 1 try	2331 5 tries	4255 8 tries	17 tries	
6	 j.camacho.2022 Universidad Rey Juan Carlos	3 10120	85 1 try	5699 7 tries	4196 2 tries	6 tries	
7	 i.penedo.2020 Universidad Rey Juan Carlos	3 11491	1201 1 try	5137 1 try	5153 1 try	7 tries	
8	 i.rodriquezb.2021 Universidad Rey Juan Carlos	3 11805	43 2 tries	5686 11 tries	5836 2 tries		
9	 m.ferrer.2020 Universidad Rey Juan Carlos	3 18403	5136 1 try	6489 12 tries	6538 2 tries		
10	 m.zucchi.2019 Universidad Rey Juan Carlos	3 21875	5229 1 try	8236 6 tries	8270 3 tries		
11	 i.gomeze.2019 Universidad Rey Juan Carlos	3 23838	6821 1 try	8347 7 tries	8530 2 tries		
12	 ja.gordillo.2018 Universidad Rey Juan Carlos	2 15808	7838 1 try	7910 4 tries	7 tries	13 tries	

<https://urjc-cp.numa.host/jury/scoreboard>



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



LENGUAJES DE PROGRAMACIÓN

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



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
}
```

```
n = int(input())
for i in range(n):
    #- 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)

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

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

```
import sys  
  
for line in sys.stdin:  
    #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
}

import sys

for line in sys.stdin:
    if(line=='0\n'):
        sys.exit()
    #codigo
```

PYTHON



CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

AC



Tu solución es al menos tan buena como la esperada

- Imprimes correctamente todos los casos ocultos
- Tu código tarda menos en ejecutarse que el tiempo límite
- Tu código termina de ejecutarse sin problema

CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

WA

WRONG ANSWER

La solución que imprimes en algún caso no coincide con la esperada



CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

RTE



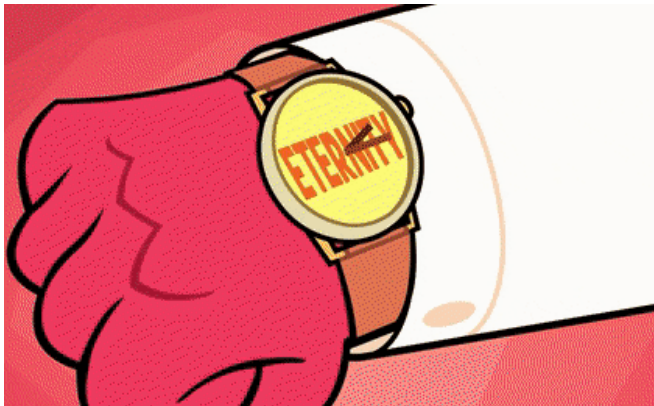
RUN TIME ERROR

Tu código muere por algún fallo en tiempo de ejecución

- Dividir entre cero
- Acceder a zonas de memoria no reservada
- Usar objetos en "null"
- ...

CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

TLE



TIME LIMIT EXCEPTION

Tu código tarda demasiado en ejecutarse

- Hay que reducir la complejidad del algoritmo!!!
- Puede esconder otros veredictos

CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

MLE



MEMORY LIMIT EXCEPTION

Tu código utiliza más memoria de la reservada

- Hay que reducir la memoria del algoritmo!!!
- Puede esconder otros veredictos

Nota: Error muy común en Acepta el Reto, verifican mucho la memoria.

CARACTERÍSTICAS DE UN PROBLEMA: HE ESCRITO MI SOLUCIÓN ¿Y AHORA QUÉ?

PE



PRESENTATION ERROR

Tu código está bien, pero no lo has impreso de la manera correcta

- No siempre se contempla, en algunas ocasiones devolverá un WA

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.



¿Y SI NO LO CONSIGO?

- **SIGUE PROBANDO!!**

Un mal veredicto no es excusa para rendirse, SIGUE!!!

- **INTERNET**

No lo tienes durante los concursos, pero entrenando no te quedes con la duda

- **CP3 y CP4**

La biblia de la programación competitiva (hay copias en la biblioteca)

- **Pide ayuda**

Usad el grupo de telegram para pedir ayuda



- [Ideone](#)
- [Diffchecker](#)



¿Podemos tener todos los casos de prueba?

¿En una empresa te dan casos de prueba para encontrar un fallo?

¿En un examen te los dan?

Si se necesitan tras mucho esfuerzo se os pasarán para aprender, pero aprender a encontrar fallos es algo muy importante.

Busca tus propios casos, pon los casos límites y prueba el algoritmo.



A PIKARSE!!!

Ranking por universidades 2022

<https://aer.lluiscab.net/institutions>

	Nombre	Miembros	Puntuación
1	Universidad Complutense de Madrid	1233	5.164.163
2	I.E.S. Comercio	156	1.338.451
3	Universidad Rey Juan Carlos	136	767.419
4	Universidad Autónoma de Madrid	63	674.278
5	I.E.S. Cuatrovientos (Pamplona)	4	463.083
6	I.E.S. Oretania (Linares)	80	425.001
7	I.E.S. La Senia (Paiporta)	71	328.146
8	Institut Sabadell	85	213.251
9	I.E.S. San Vicente (Alicante)	88	188.804
10	Universidad de Valladolid	42	182.404



A PIKARSE!!!

Ranking por universidades 2023

<https://aer.lluiscab.net/institutions>

	Nombre	Miembros	Puntuación
1	Universidad Complutense de Madrid	1516	6.344.983
2	I.E.S. Comercio	214	1.984.494
3	Universidad Rey Juan Carlos	202	1.148.551
4	Universidad Autónoma de Madrid	68	628.289
5	I.E.S. Oretania (Linares)	101	462.174
6	I.E.S. La Senia (Paiporta)	115	431.214
7	I.E.S. Cuatrovientos (Pamplona)	5	401.030
8	Universidad Politécnica de Madrid	157	383.208
9	I.E.S. Muralla Romana (Lugo)	85	338.353
10	I.E.S. San Vicente (Alicante)	99	255.387



A PIKARSE!!!

Ranking Codeforces

<https://codeforces.com/ratings/organization/2365>

Rating			
	Who	#	=
1 (14397)	 jjluczyn	37	1530
2 (24072)	 Serg_	14	1354
3 (25146)	 tortolavivo23	6	1338
4 (26025)	 alii.pz	22	1325
5 (38761)	 saragarz	14	1176
6 (45814)	 poloWin	15	1103
-	 ddavidm1	15	1581
-	 rmartin	12	1393
-	 Ouyia	2	1345
-	 isaaclo97	26	1079



A PIKARSE!!!

Bot creado por David

AERStalker

Lista de mis comandos

`/help` Muestra este mensaje

`/start` Inicia el bot

`/aer_stalk` <nombre> comienza a seguir a un usuario, cuando tenga actividad en AceptaElReto, lo sabrás

`/aer_unstalk` <nombre> Elimina el seguimiento a un usuario en AceptaElReto

`/cf_stalk` <nombre> comienza a seguir a un usuario, cuando tenga actividad en Codeforces, lo sabrás

`/cf_unstalk` <nombre> Elimina el seguimiento a un usuario en Codeforces

`/list` Muestra todos los usuarios que se están siguiendo actualmente

`/about` Muestra información acerca del bot

Reply

11:43

Si os agobia podéis saliros, pero anima bastante a participar a todos los demás si el BOT no para de saltar!

AERStalker

El usuario ali.pz ha tenido AC 🏆✅ en 672 - Terminando los chalets

Reply

16:16

AERStalker

El usuario cftryharder ha tenido WA 😞❌ en C - Card Game

17:48

El usuario cftryharder ha tenido WA 😞❌ en C - Card Game

17:48

El usuario cftryharder ha tenido AC 🏆✅ en C - Card Game

17:54



Ejemplo Problema: 112A

<https://codeforces.com/problemset/problem/112/A>

A. Petya and Strings

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Little Petya loves presents. His mum bought him two strings of the same size for his birthday. The strings consist of uppercase and lowercase Latin letters. Now Petya wants to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter. Help Petya perform the comparison.

Input

Each of the first two lines contains a bought string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

Output

If the first string is less than the second one, print "-1". If the second string is less than the first one, print "1". If the strings are equal, print "0". Note that the letters' case is not taken into consideration when the strings are compared.



Entrenamiento

- Páginas de entrenamiento.
 - [Acepta el Reto](#)
 - [Codeforces](#)
 - [OnlineJudge](#) y [uHunt](#)
 - [Spoj](#)

Recomendable tener un nick



Enlaces del curso

- El enlace del curso

<https://urjc-cp.github.io/urjc-cp/>

¿QUÉ AULA ERA?



- Enlace de otras ediciones (algunas grabadas)

<https://david8k.github.io/>

Este año el curso NO se grabará.



Tengo mucho interés, ¿cómo avanzar?

1. Practicar lo máximo posible sobre todo en concursos de Codeforces (semanales).
2. Revisar material de otros años de las siguientes clases y practicar (esto el año pasado implicó medalla en primero)
3. Leer, entender y practicar cada tema del libro Competitive Programming 3 o 4 (disponible en biblioteca).

Quien haga estos pasos durante varios años asegura medalla en el Europeo.



¿Preguntas?



HASTA LA SEMANA QUE VIENE!



@URJC_CP

@Dijkstraideos

Formulario

