

PROGRAMACIÓN COMPETITIVA

URJC - 2019



PROGRAMACIÓN COMPETITIVA

URJC - 2019

Organizadores:

- Raul Martín (raul.martin@urjc.es)
- Isaac Lozano (isaac.lozano@urjc.es)
- Jesús Sánchez-Oro (jesus.sanchezoro@urjc.es)
- Juan David Quintana (juandavid.quintana@urjc.es)
- Sergio Pérez (sergio.perez.pelo@urjc.es)

Motivación

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

Motivación

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



Motivación

- Participación en concursos
 - SWERC (Europeo del SurOeste)
 - AdaByron
 - Google Hashcode
 - 12 Uvas
 - ... premios y honor!!!



OBJETIVOS EN PROGRAMACIÓN COMPETITIVA

- Resolver los problemas en el menor tiempo posible
- Trabajo en equipo (nombres creativos. Ej. Calce Team, Raspu Team, etc)
- Representar tu institución, país...

PLANIFICACIÓN DE LA CHARLA
































Objetivos:

- Familiarizarse con jueces
- Que aprendan nociones básicas de la programación
- Familiarizarse con los entornos de desarrollo
- Realizar un primer concurso de programación

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 RaciETH 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	  iIUSlon 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

ABRIR MYAPPS URJC

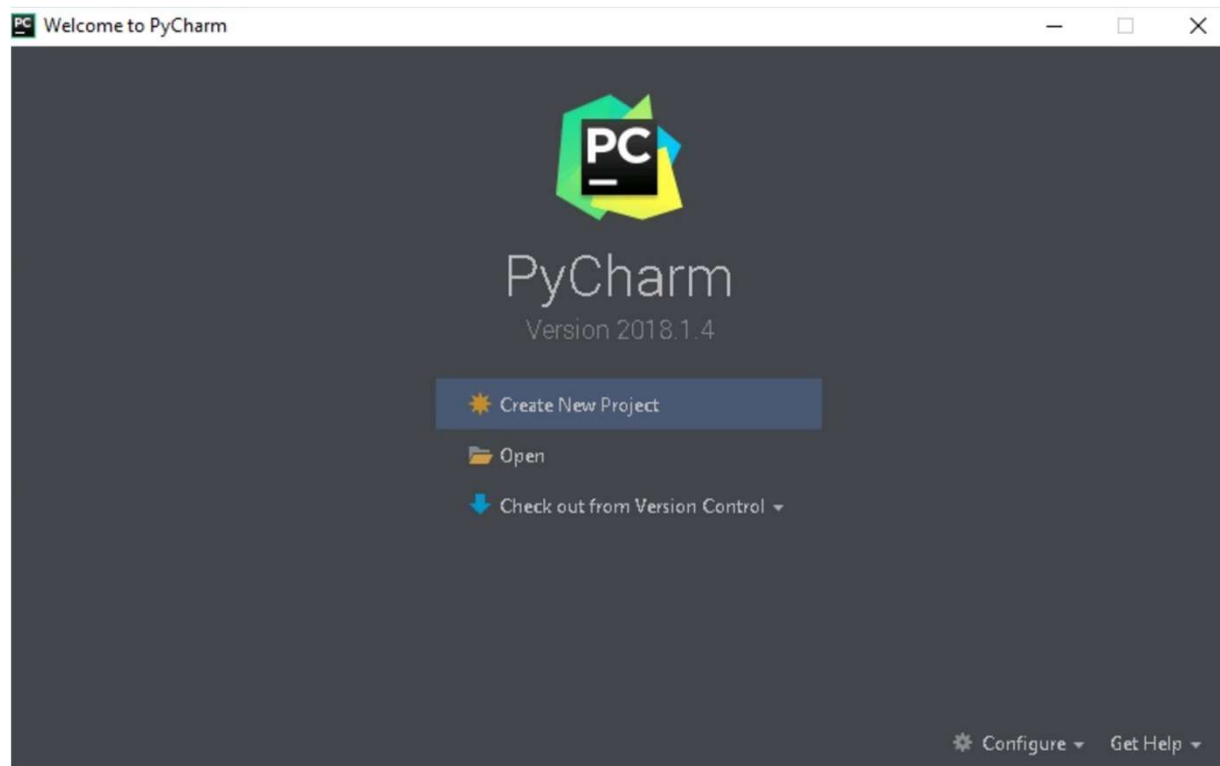
The screenshot displays the myApps URJC web interface. At the top, the Universidad Rey Juan Carlos logo and 'myApps' branding are visible. Navigation tabs include 'Aplicaciones', 'Descargas', 'my R:', and 'Ayuda'. A search bar and a 'Lista' dropdown menu are present. Below the navigation bar, a grid of application icons is shown, including: _ Cerrar todas las Sesiones_, 7- zip 18.01, Adobe Acrobat DC, Escritorio Desarrollo, Explorador de Archivos myApps, Firefox, Google Chrome, Internet Explorer, Microsoft Access 2016, Microsoft Excel 2016, Microsoft OneNote 2016, Microsoft PowerPoint 2016, Microsoft Publisher 2016, Microsoft Word 2016, Ubuntu, and VLC Media Player. Two red arrows point to the 'HTML5' button in the top right corner of the application grid. At the bottom, a disclaimer states: 'Las aplicaciones aquí contenidas son exclusivamente para fines educativos, quedando expresamente prohibidos los usos comerciales, ilícitos o lesivos de bienes, servicios, derechos o intereses de terceros.'

<https://myapps.urjc.es/myapps>

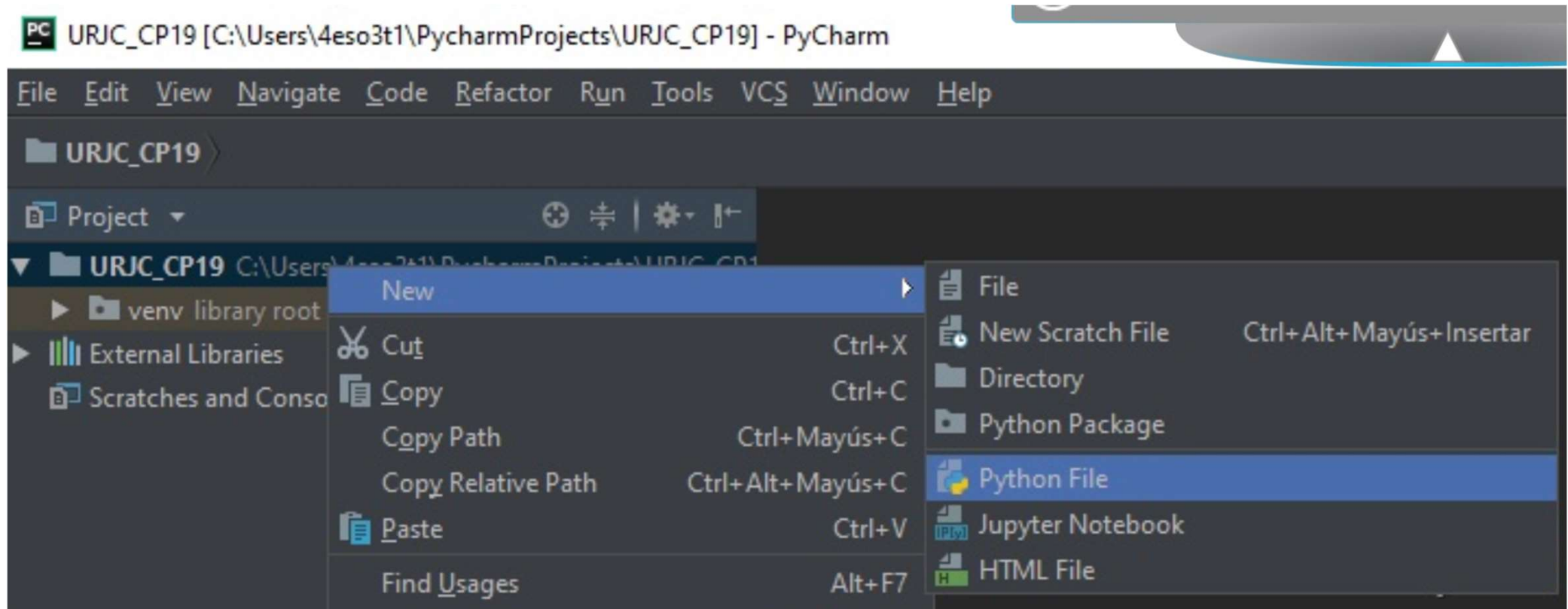
Escritorio de Desarrollo



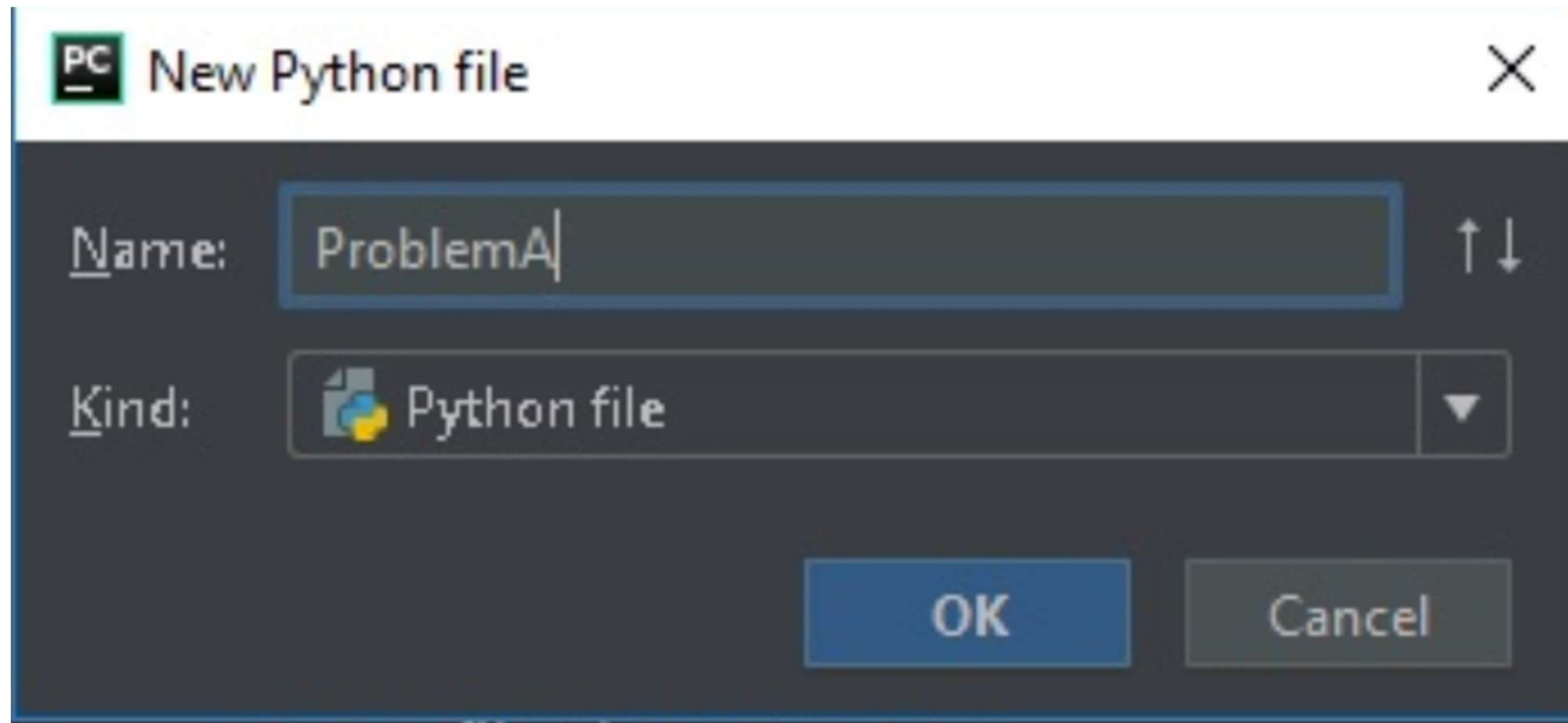
PyCharm - Crear Proyecto



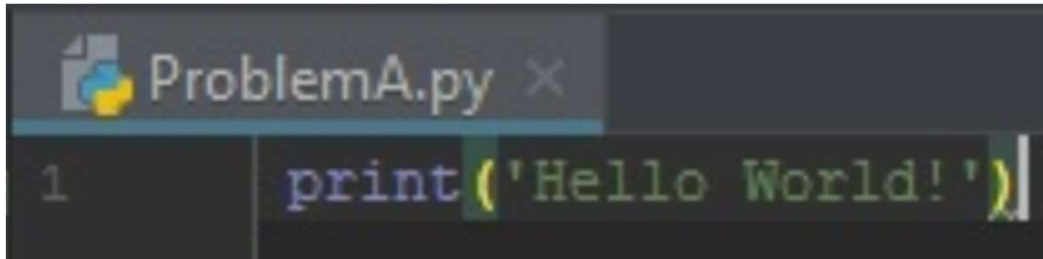
PyCharm - Crear Archivo .py



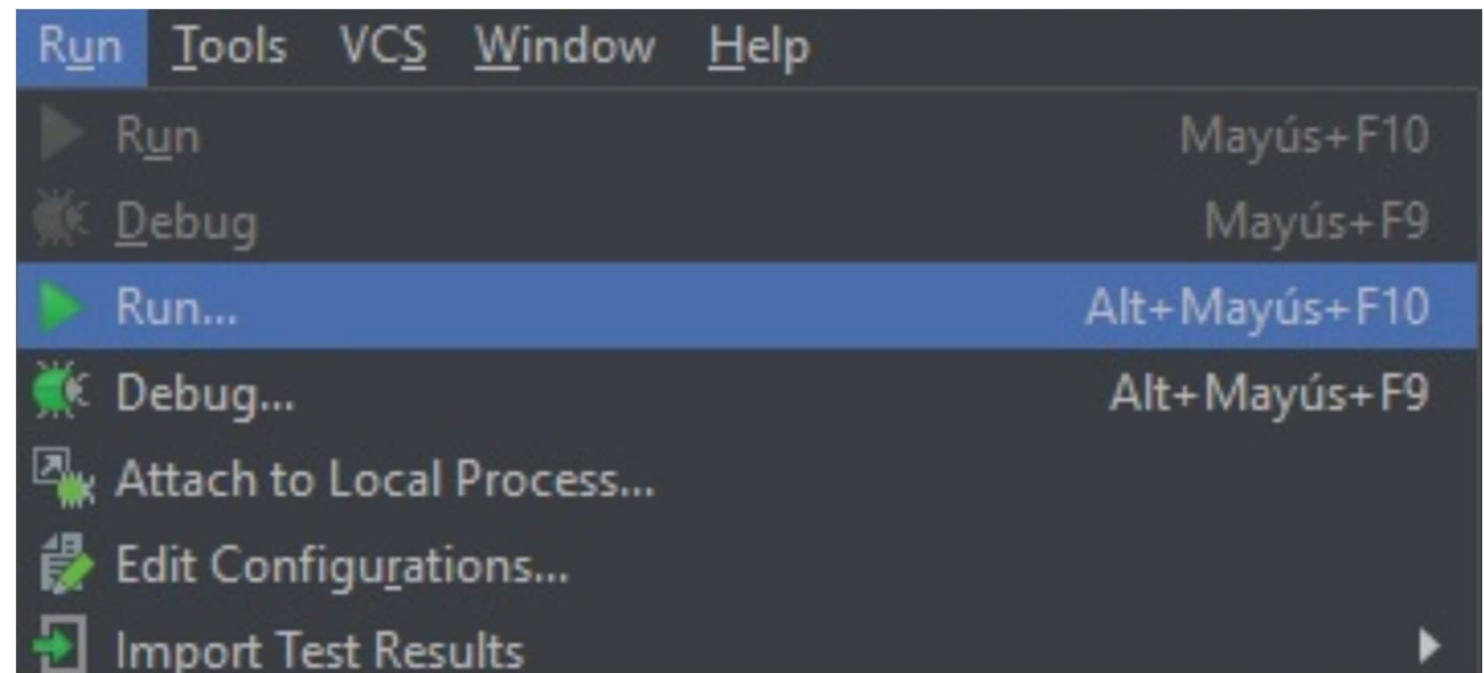
PyCharm - Crear Archivo .py



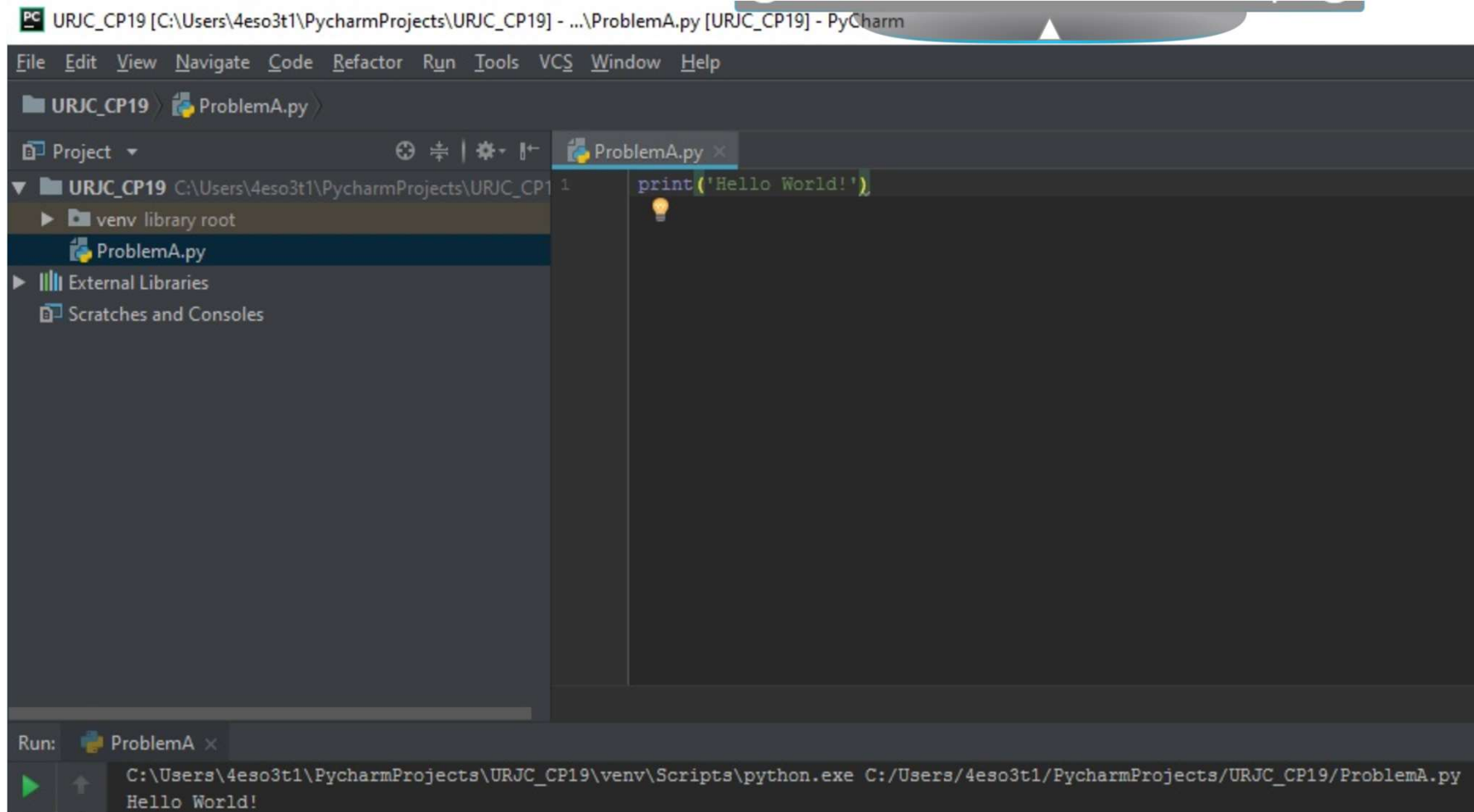
PyCharm - Primer Ejemplo!



```
ProblemA.py x
1 print('Hello World!')
```



PyCharm - Primer Ejemplo!



LENGUAJES DE PROGRAMACIÓN

- Python

Sintaxis Básica Python

- Python

- ¿Que es una variable?

En programación una variable es un valor que puede cambiar dependiendo de las condiciones o la información que se le pasa al programa.

Por ejemplo si queremos tener en nuestro programa un número y utilizarlo lo podemos asociar a la variable y posteriormente operar sobre dicha variable.

Sintaxis Básica Python

- Python

- ¿Que es una variable?

```
num15 = 15
```

```
num15 = num15 + 5 # num15 valdrá 20
```

```
num15 = 2 # num15 valdrá 2
```

```
num10 = num15 # ambos valdrán 2
```

```
nombre = "Hola" # nombre es "Hola"
```

```
nombre = "Pepe" # nombre es "Pepe"
```

Sintaxis Básica Python

- Python

- Pedir un valor: `var = input()`
- Pedir un valor entero: `var = int(input())`
- Pedir dos números, cada número estará en la variable correspondiente.
`n1, n2 = map(int, input().split())`

Sintaxis Básica Python

- Python

- Imprimir un salto de línea: `print()`
- Imprimir texto sin salto de línea: `print(".", end="")`

- Imprimir un texto:

```
imprimir = "Texto"  
print(imprimir)
```

- Imprimir una variable

```
imprimir= "Texto" + str(var)  
print(imprimir)
```

Sintaxis Básica Python

- Python

- Hacer un bucle para recorrer N veces:

N = 3

```
for i in range(N):  
    imprimir = "Iter: " + str(i)  
    print(imprimir)
```

Tabulación!

```
Iter: 0  
Iter: 1  
Iter: 2
```

Sintaxis Básica Python

- Python

- Leer 4 palabras, por cantidad de palabras crear una variable:

```
p, p1, p2, p3 = input().split()
```

- Condiciones: Si numero es igual a 1 leer palabra, si no imprime la palabra

```
palabra = ""  
if(numero==1):    //Si numero es igual a 1  
    palabra = input()  
else:  
    print(palabra)
```

Sintaxis Básica Python

- Python
- Condicionales comparar palabras.

```
palabra = "NO"
if (palabra == "SI") :
    print("NO")
else:
    print("SI")
```

```
palabra = "SI"
if (palabra == "SI") :
    print("NO")
else:
    print("SI")
```

Sintaxis Básica Python

- Python

- Operaciones matemáticas:

```
a = 1+1
```

```
a = a*1
```

```
b = 1+a
```

```
c = a-b
```

```
d = a//b #Division de números enteros
```

```
e = 2*3
```


Sintaxis Básica Python

- Python
- Comentarios en Python

Para comentar se utiliza el #, por ejemplo

```
print("Hola") #Imprime hola, soy un comentario
```

Para comentar múltiples líneas se utiliza ''' '''

```
#Todo comentado
```

```
''' print("Hola")  
    #Imprime hola, soy un comentario  
    print("Fin")
```

```
'''
```

Ejemplo Problema: ¡Hola mundo!

Problem description

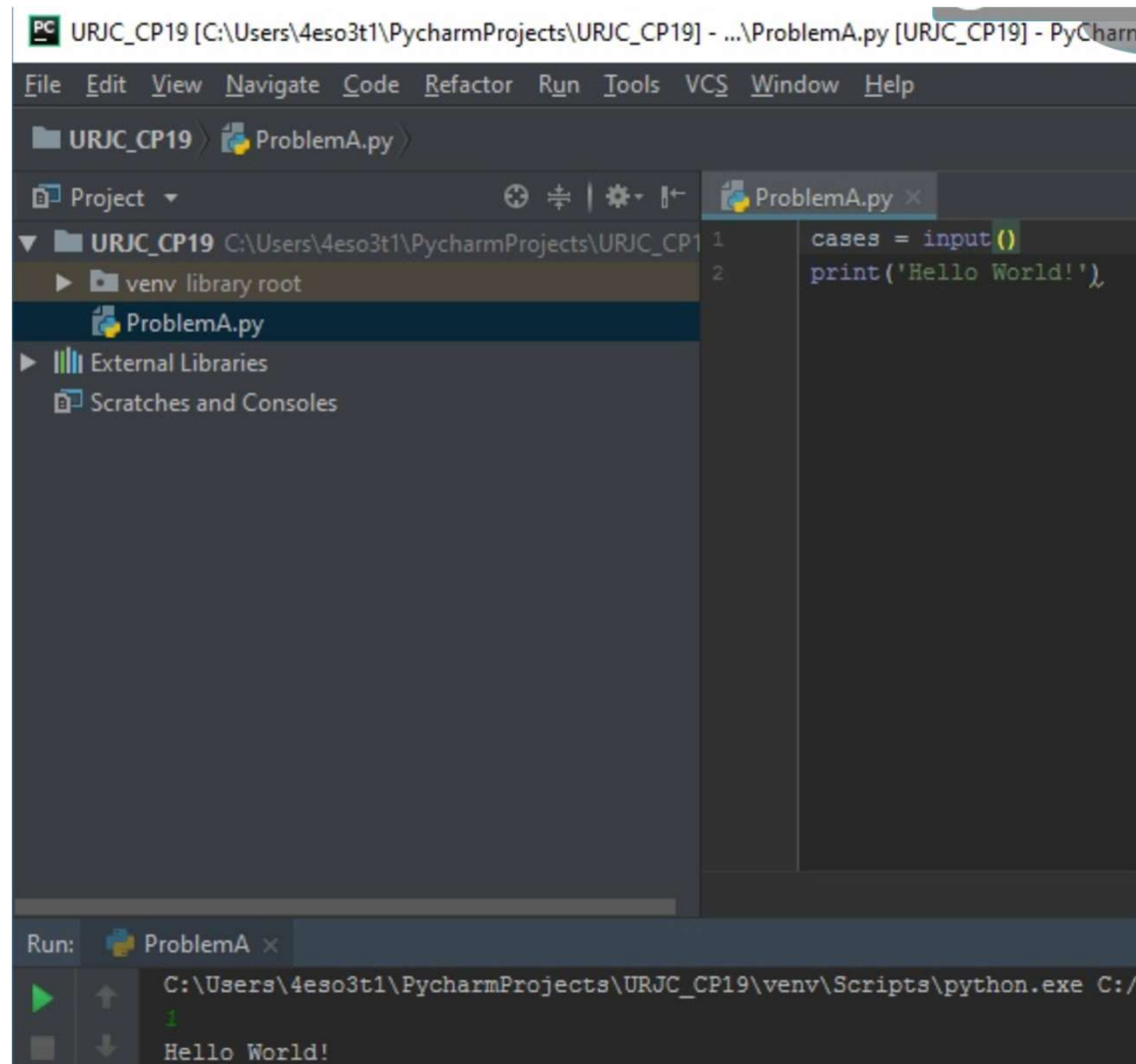
You are to write the most basic program; it should just output “Hello world!” on a single line, no matter what the input.

Sample input/output

Sample input and output for this problem:

Input	Output
1	Hello world!

Ejemplo Problema: ¡Hola mundo!

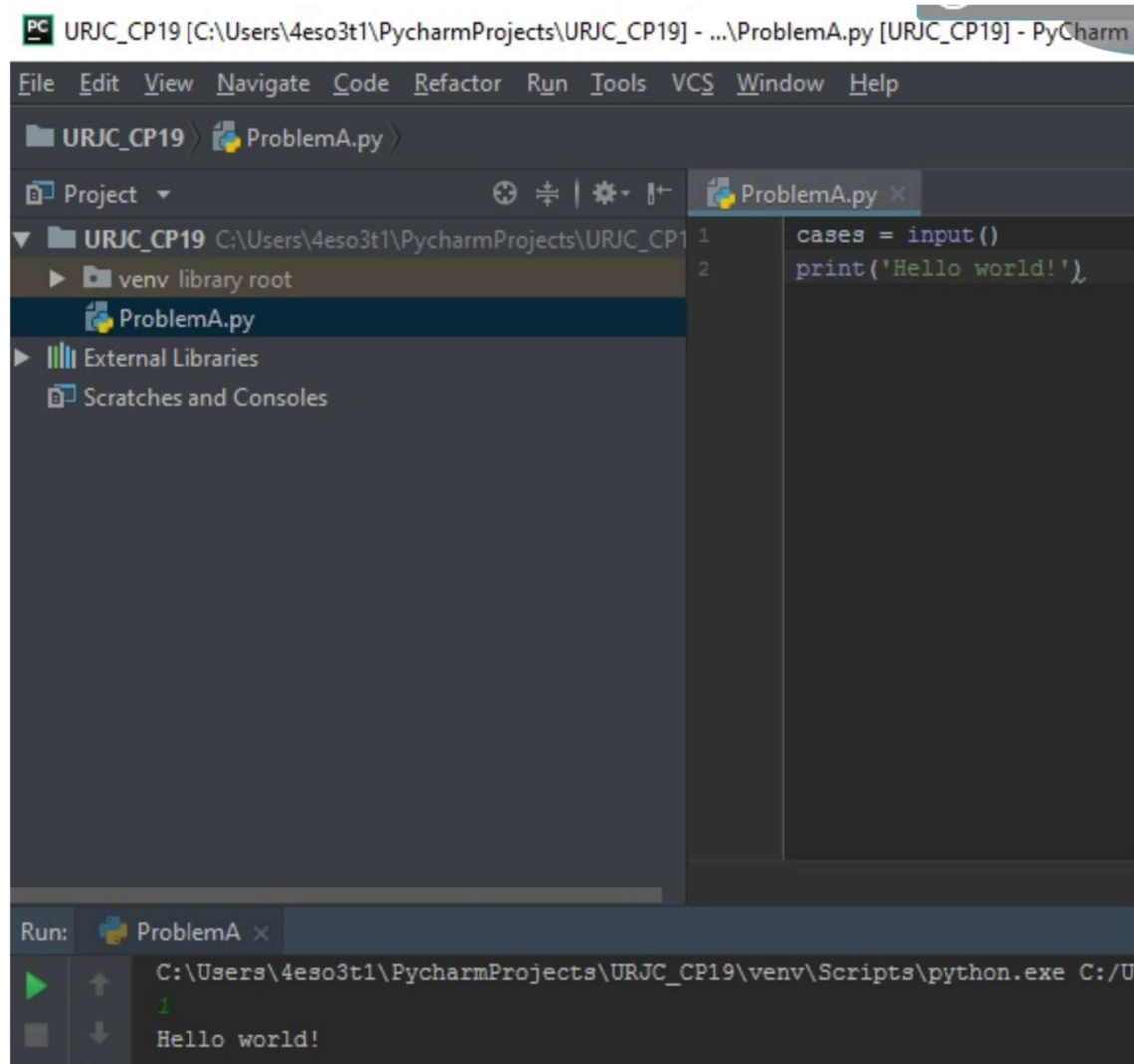


```
PC URJC_CP19 [C:\Users\4eso3t1\PycharmProjects\URJC_CP19] - ...\ProblemA.py [URJC_CP19] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
URJC_CP19 > ProblemA.py
Project
URJC_CP19 C:\Users\4eso3t1\PycharmProjects\URJC_CP19
venv library root
ProblemA.py
External Libraries
Scratches and Consoles
1 cases = input()
2 print('Hello World!')
Run: ProblemA x
C:\Users\4eso3t1\PycharmProjects\URJC_CP19\venv\Scripts\python.exe C:/
1
Hello World!
```



Hello World! no
es lo mismo que
Hello world!

Ejemplo Problema: ¡Hola mundo!



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project browser on the left shows the project 'URJC_CP19' with a subdirectory 'venv library root' and a file 'ProblemA.py'. The main editor window displays the code for 'ProblemA.py':

```
1 cases = input()
2 print('Hello world!')
```

The Run tool bar at the bottom shows a green play button, and the Run console displays the output: 'Hello world!'.



Ejemplo Problema: ¡Hola mundo! N casos

Entrada

La primera línea contiene un entero N denotando un número entero.
La entrada debe ser leída de forma estándar.

Salida

Se debe imprimir la cadena “Hello World!” tantas veces como indique el número N .

La salida debe ser escrita de forma estándar.

Entrada ejemplo	Salida ejemplo
5	Hello World! Hello World! Hello World! Hello World! Hello World!

Límites

- $0 \leq N \leq 10$

Ejemplo Problema: ¡Hola mundo! N casos

```
ProblemA.py x
1 cases = int(input())
2 for i in range(cases):
3     print('Hello World!')
```

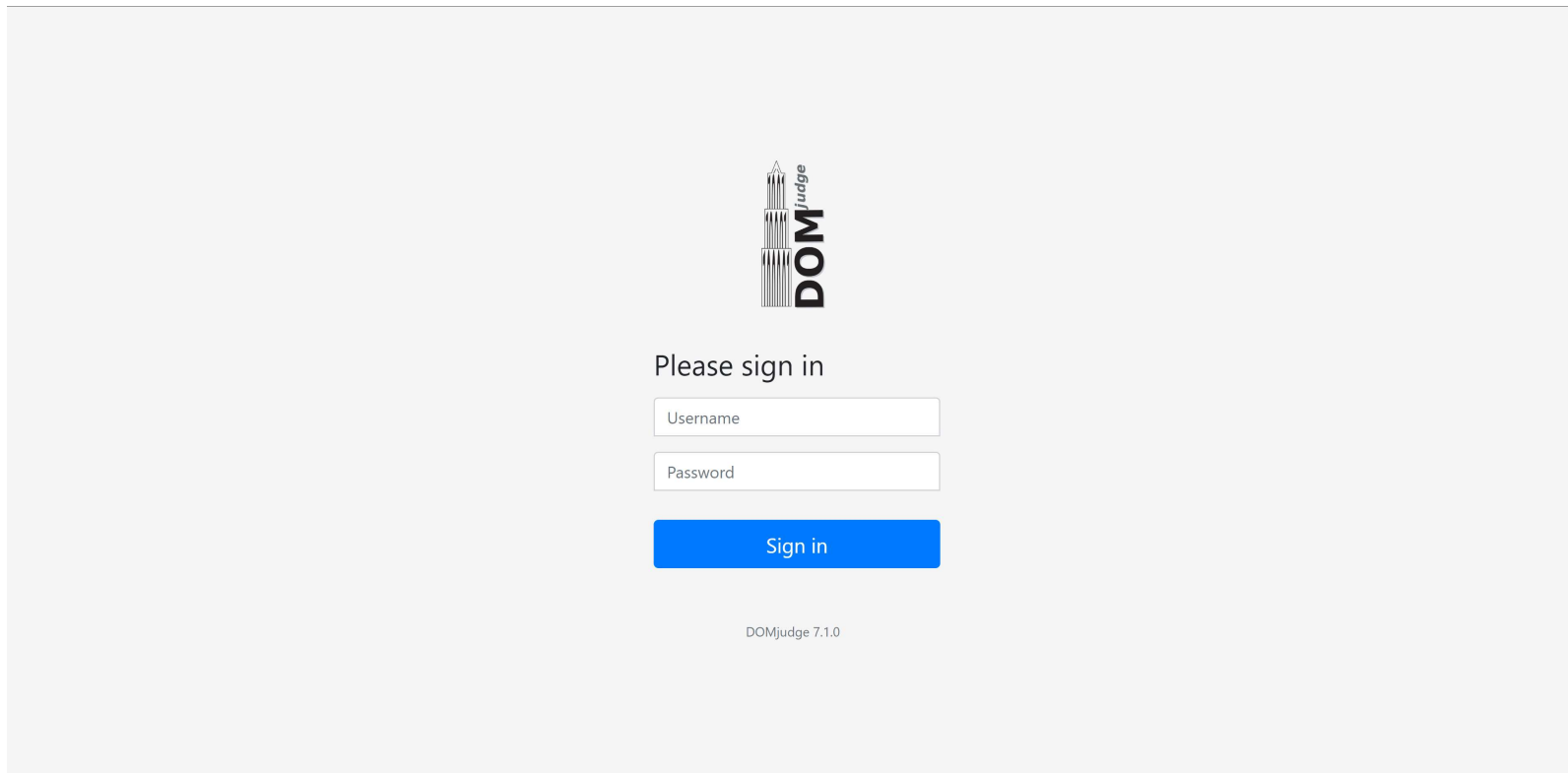
```
C:\Users\4eso3
5
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

Importante la indentación (tabulaciones) además de recoger el elemento como un entero.

Domjudge

- Enlace: <http://urjc-cp.numa.host/public>
- Nicks y contraseñas las tenéis en el papel

- Enlace: <http://urjc-cp.numa.host/login>



The image shows a screenshot of the DOMjudge login page. At the top center is the DOMjudge logo, which consists of a stylized building icon and the text 'DOMjudge'. Below the logo, the text 'Please sign in' is displayed. Underneath this text are two input fields: 'Username' and 'Password'. Below the password field is a blue button labeled 'Sign in'. At the bottom center of the page, the version number 'DOMjudge 7.1.0' is displayed.

Domjudge

DOMjudge

[Home](#) [Problemset](#) [Scoreboard](#)

[Submit](#) [Logout](#) URJC_SWERC_2018 4008d 21:10:14

RANK	TEAM	SCORE	P81-COMETAS	P82-DISTOTED	P83-TELESCOPIO	P84-GALAXY	P85-GALILEO	P86-CONVERTER	P87-ROVER	P88-GUERRA
1	i.lozano_CP Universidad Rey Juan Carlos	8 4870	28 1 try	30 1 try	519 5 tries	655 4 tries	1798 1 try	21 1 try	1510 8 tries	29 1 try

Submissions

time	problem	lang	result
2019-11-03 21:28	P85-GALILEO	CPP	CORRECT
2019-11-03 16:40	P87-ROVER	CPP	CORRECT
2019-11-03 16:09	P87-ROVER	CPP	WRONG-ANSWER
2019-11-03 02:25	P84-GALAXY	CPP	CORRECT
2019-11-03 02:25	P84-GALAXY	CPP	NO-OUTPUT
2019-11-03 00:58	P84-GALAXY	CPP	WRONG-ANSWER
2019-11-03 00:18	P83-TELESCOPIO	CPP	CORRECT
2019-11-03 00:15	P83-TELESCOPIO	CPP	WRONG-ANSWER
2019-11-03 00:12	P83-TELESCOPIO	CPP	WRONG-ANSWER
2019-11-03 00:09	P83-TELESCOPIO	CPP	CORRECT
2019-11-02 23:09	P83-TELESCOPIO	CPP	WRONG-ANSWER
2019-11-02 23:04	P83-TELESCOPIO	CPP	WRONG-ANSWER

Clarifications

No clarifications.

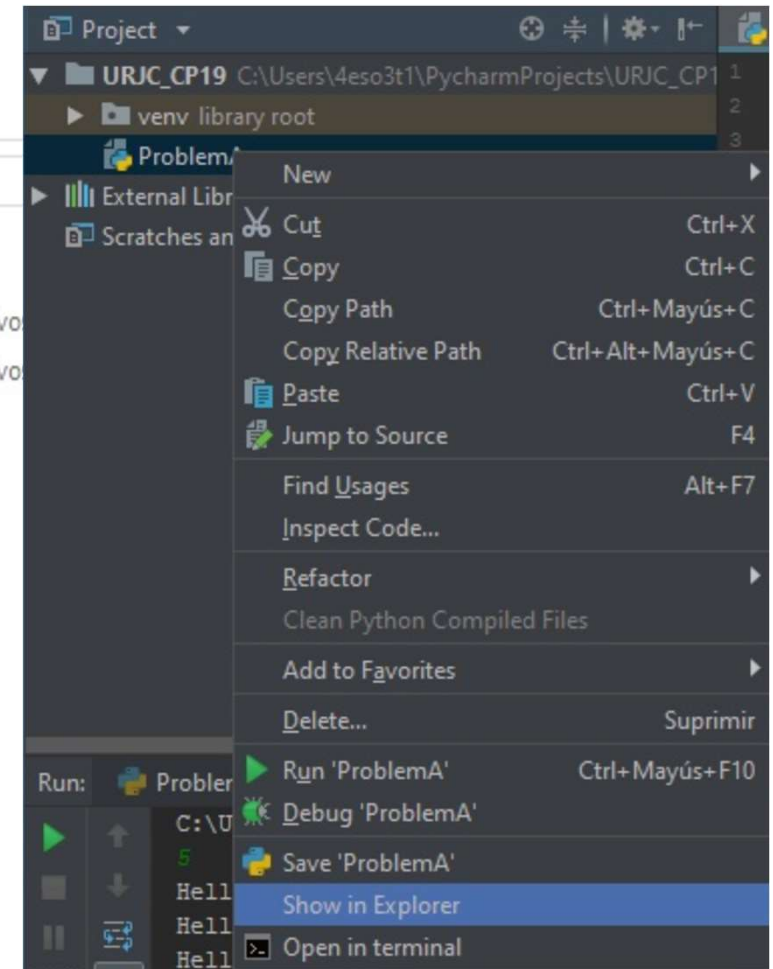
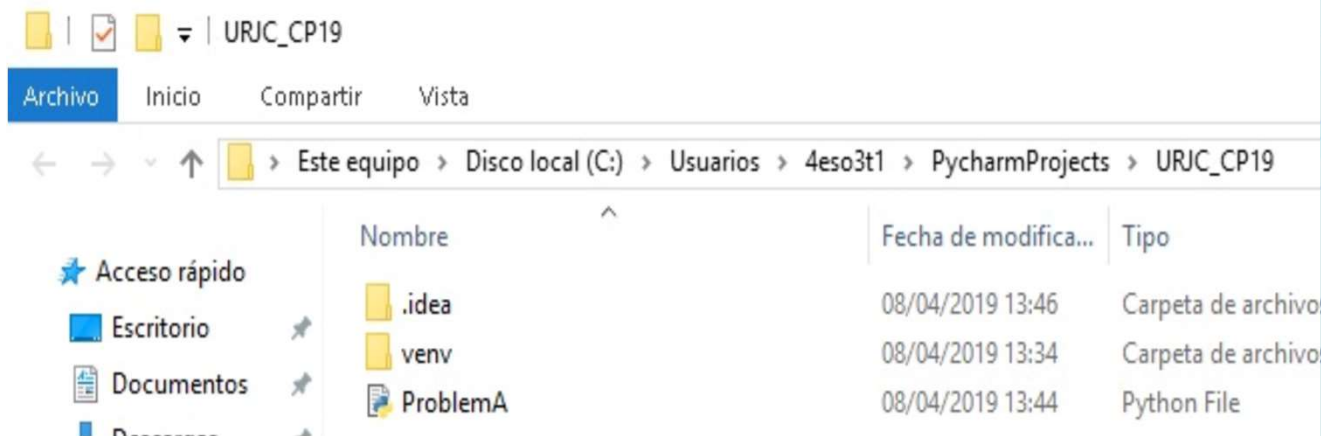
Clarification Requests

No clarification request.

request clarification



Domjudge - Seleccionar Archivo



C:\Users\NOMBRE_DE_USUARIO\PycharmProjects\NOMBRE_DE_PROYECTO

Domjudge

DOMjudge [Home](#) [Problemset](#) [Scoreboard](#)

[Submit](#)

[Logout](#)

[URJC_SWERC_2018](#) [4008d 21:09:34](#)

Concurso de practica para el Clasificador SWERC de la URJC 2019

starts: 2019-11-02 15:30 - ends: 2030-11-04 17:00

[Filter](#)

RANK	TEAM	SCORE	P81-COMETAS	P82-DISTOTED	P83-TELESCOPIO	P84-GALAXY	P85-GALILEO	P86-CONVERTER	P87-ROVER	P88-GUERRA
1	Competitive Programming i.lozano_CP Universidad Rey Juan Carlos	8 4870	28 1 try	30 1 try	519 5 tries	655 4 tries	1798 1 try	21 1 try	1510 8 tries	29 1 try
2	fj.klaiber.2017_CP Universidad Rey Juan Carlos	4 5028	1362 1 try	1225 2 tries	3 tries		1198 1 try	1223 1 try	1 try	1 try
3	q.ye_CP Universidad Rey Juan Carlos	3 6729	1831 4 tries	1940 1 try	6 tries		2878 2 tries	2 tries		
4	f.tortola.2018_CP Universidad Rey Juan Carlos	2 5685		2840 3 tries			2805 1 try			
5	a.garciaca.2016_CP Universidad Rey Juan Carlos	1 10421				10361 4 tries			9 tries	
6	a.arlandis.2019_ipgic Universidad Rey Juan Carlos	0 0								
	a.arribasp.2017_ip_mat Universidad Rey Juan Carlos	0 0								
	a.campos.2019_CP Universidad Rey Juan Carlos	0 0								
	a.campos.2019_ipgicib Universidad Rey Juan Carlos	0 0								
	a.centeno.2018_ipgicib Universidad Rey Juan Carlos	0 0								
	a.crespos_ip_mat Universidad Rey Juan Carlos	0 0								
	a.dejesus.2019_ipgicib Universidad Rey Juan Carlos	0 0								



Contest problems

Comets Night
Limits: 2 seconds / 512 MB

[problem text](#)

Samples

[input](#)

[output](#)

[zip with all samples](#)

Distorted
Limits: 1 second / 512 MB

[problem text](#)

Samples

[input](#)

[output](#)

[zip with all samples](#)

blackhole
Limits: 2 seconds / 512 MB

[problem text](#)

Samples

[input](#)

[output](#)

[zip with all samples](#)

Galaxy Detector
Limits: 1 second / 512 MB

[problem text](#)

galileo
Limits: 1 second / 512 MB

[problem text](#)

Converter
Limits: 1 second / 512 MB

[problem text](#)

rover
Limits: 2 seconds / 512 MB

Galactic War
Limits: 1.5 seconds / 512 MB

¿Preguntas?



Comienza el concurso a las 16:00
SEMANA_DE_LA_CIENCIA



COMPETITIVE
PROGRAMMING