

PROGRAMACIÓN COMPETITIVA EN PYTHON

Universidad Rey Juan Carlos



PROGRAMACIÓN COMPETITIVA

URJC - 2020

Organizadores:

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

Motivación

- Mejora en nuestras habilidades como programadores
 - Diseño de algoritmos
 - Estructuras de datos
 - Nociones de complejidad
 - Aprobar asignaturas!!!
 - Diversión (la más importante)

Motivación

- Empresas patrocinadoras
 - Cazatalentos
 - Concursos internos
 - Entrevistas de trabajo



Motivación

- Participación en concursos nacionales e internacionales
 - 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, Team Burton, SQLete, etc)
- Representar tu institución, país...
- Contacto con estudiantes de informática de universidades de todo el mundo

PLANIFICACIÓN DE LA SESIÓN
































Objetivos:

- Familiarizarse con jueces
- Aprender nociones básicas de 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	  ilIUSlon 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)

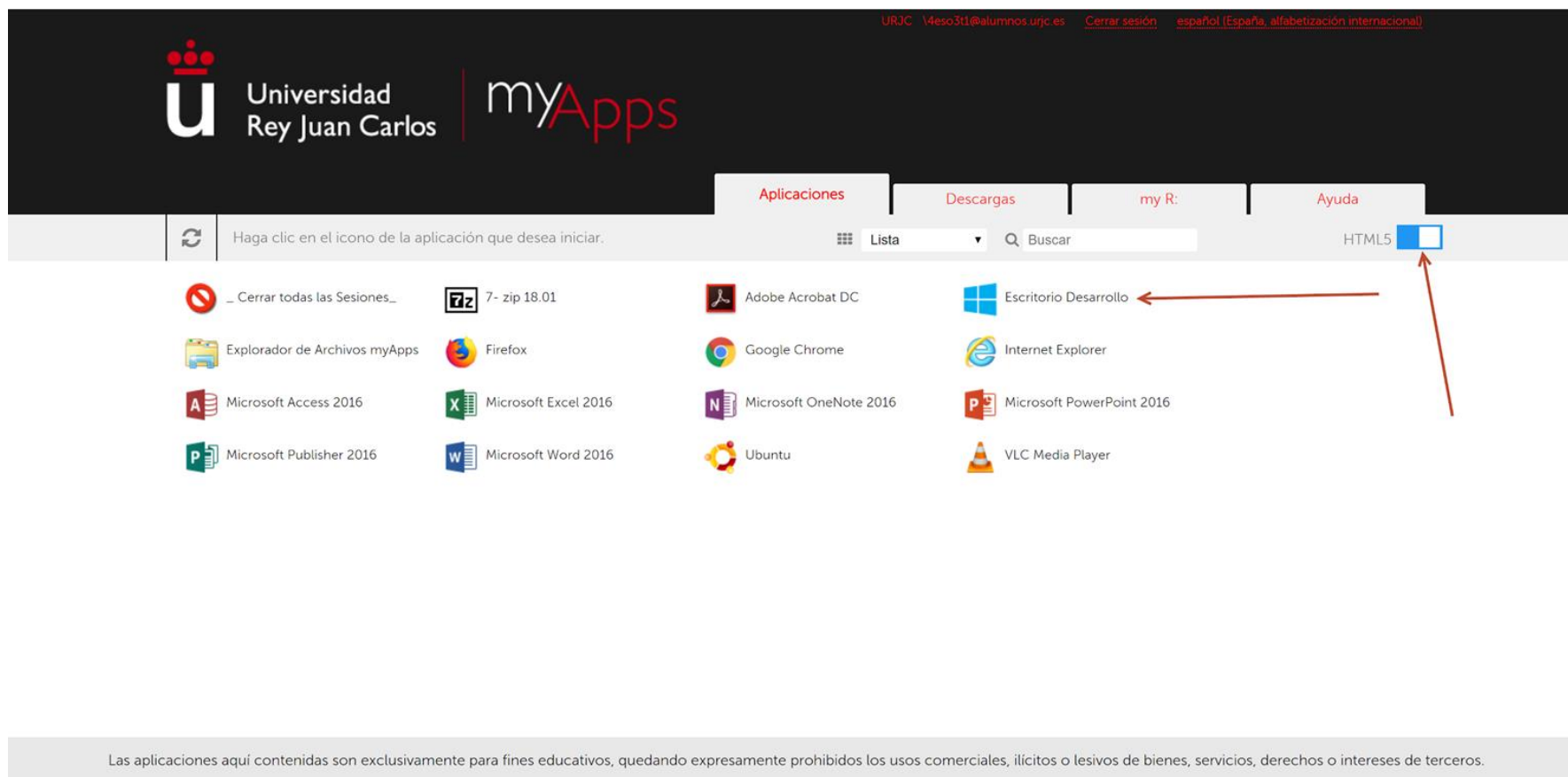
Entrada: Se especifica lo que nuestro programa debe leer

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

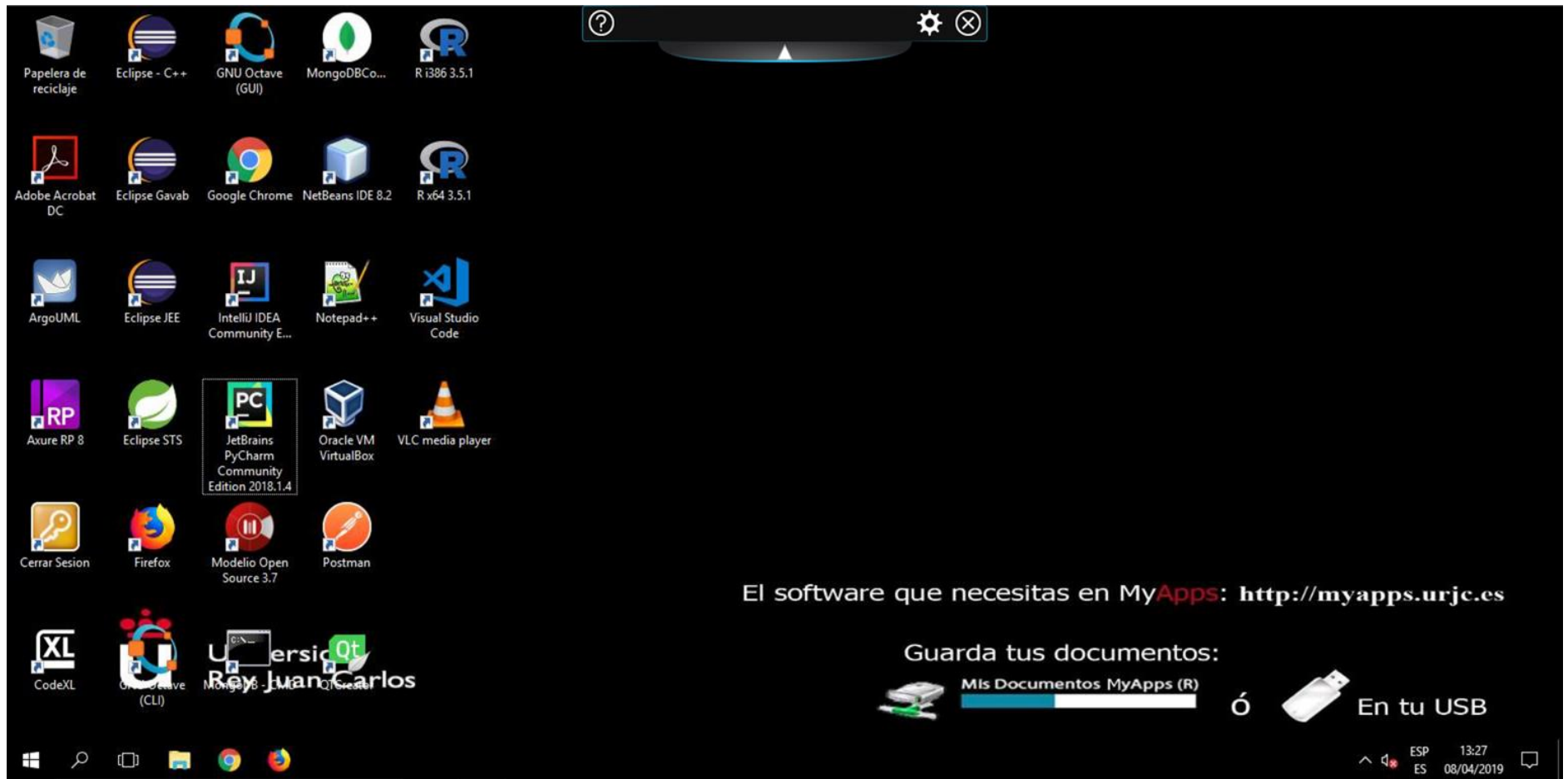
ABRIR MYAPPS URJC



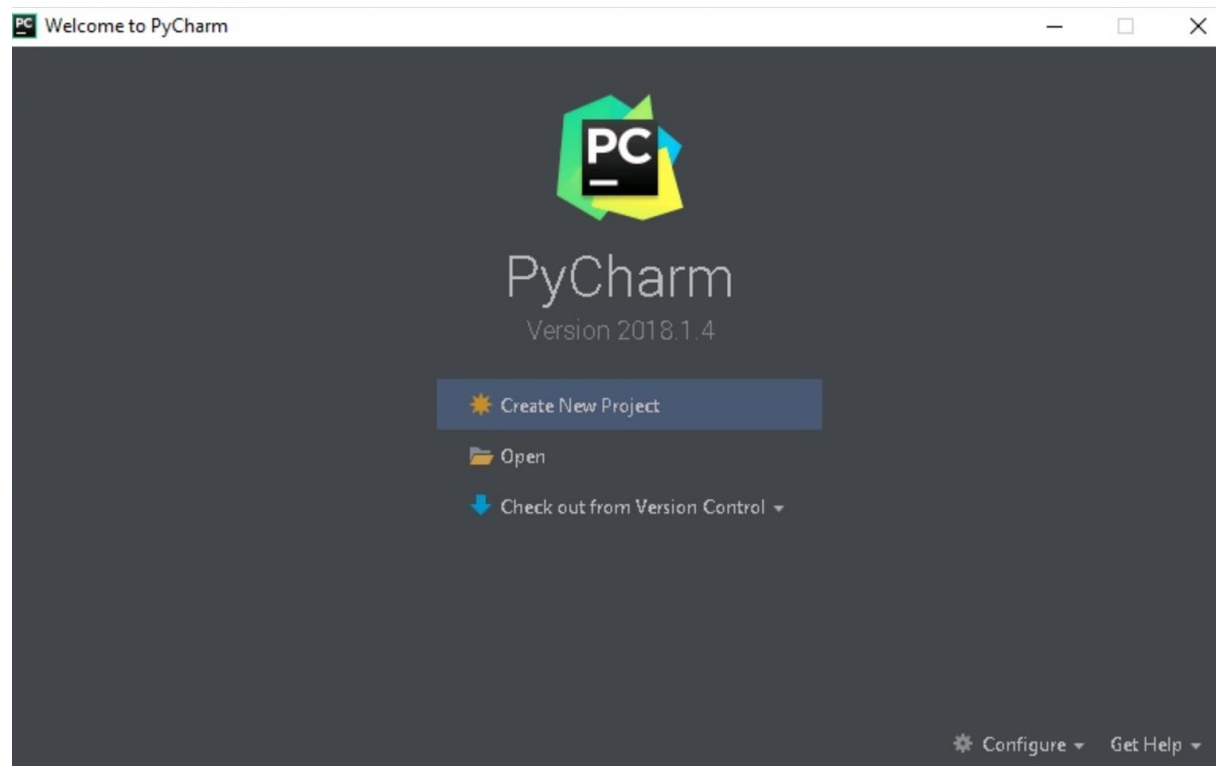
The screenshot shows the myApps URJC interface. At the top, there is a header with the Universidad Rey Juan Carlos logo and the myApps text. Below the header, there are navigation tabs: Aplicaciones, Descargas, my R, and Ayuda. A search bar is also present. The main area displays a grid of application icons, 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. A red arrow points to the 'Escritorio Desarrollo' icon. At the bottom, there is a disclaimer: '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>

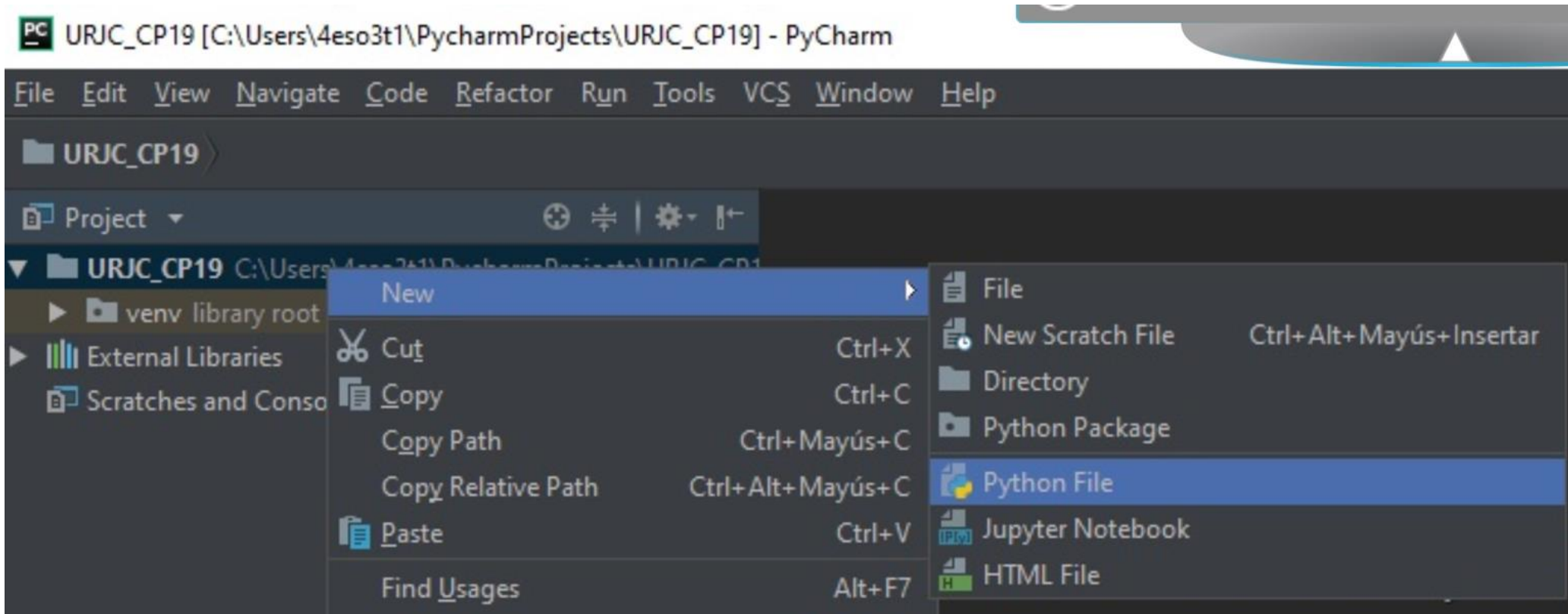
Escritorio de Desarrollo



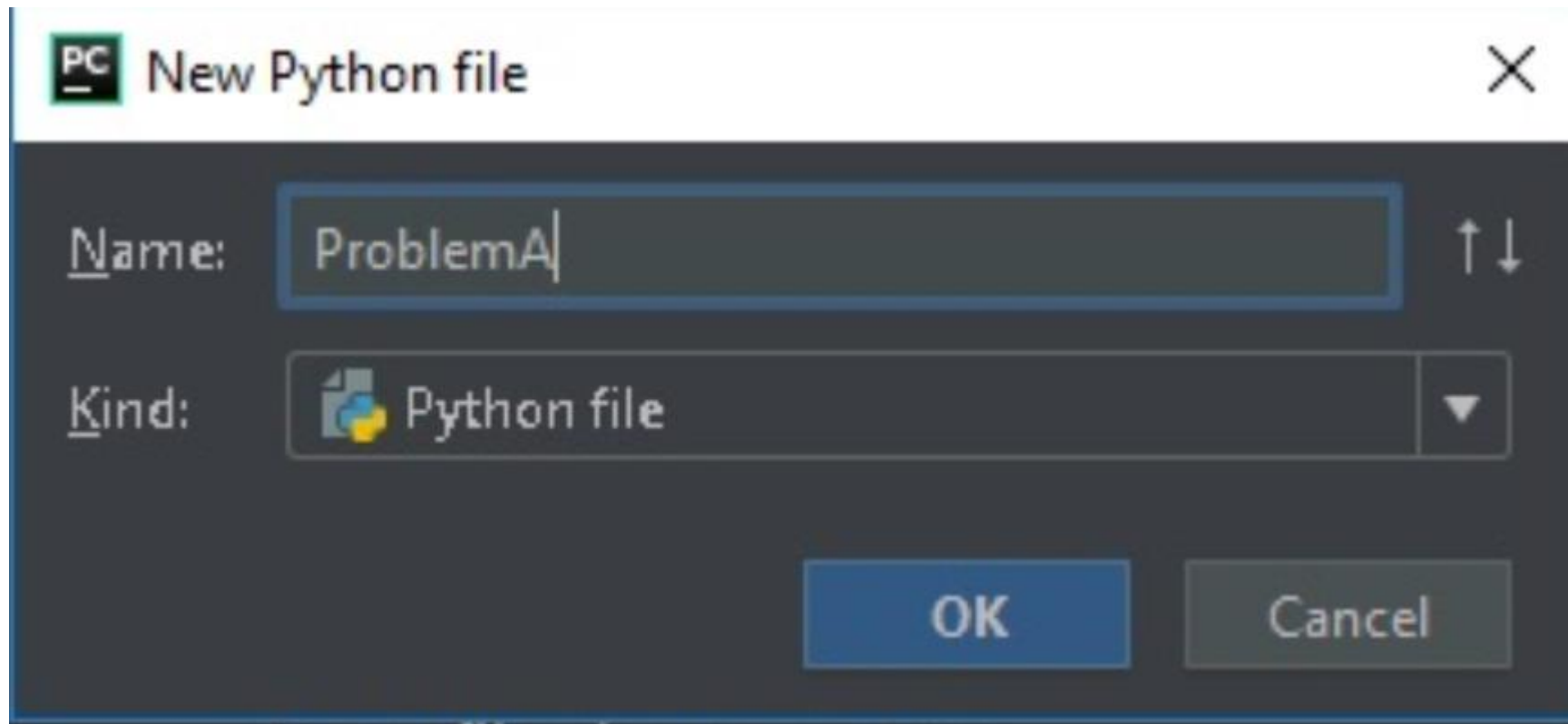
PyCharm - Crear Proyecto



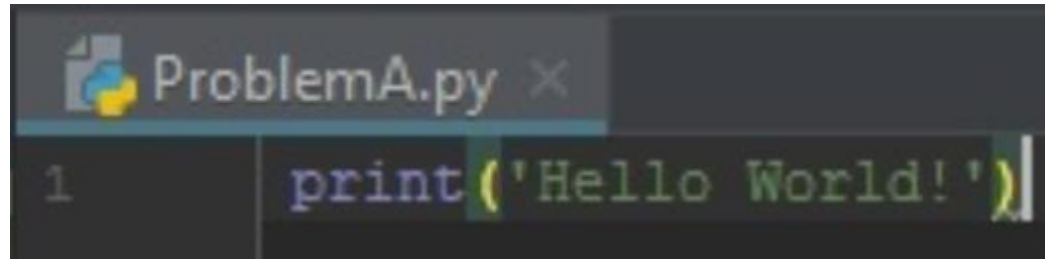
PyCharm - Crear Archivo .py



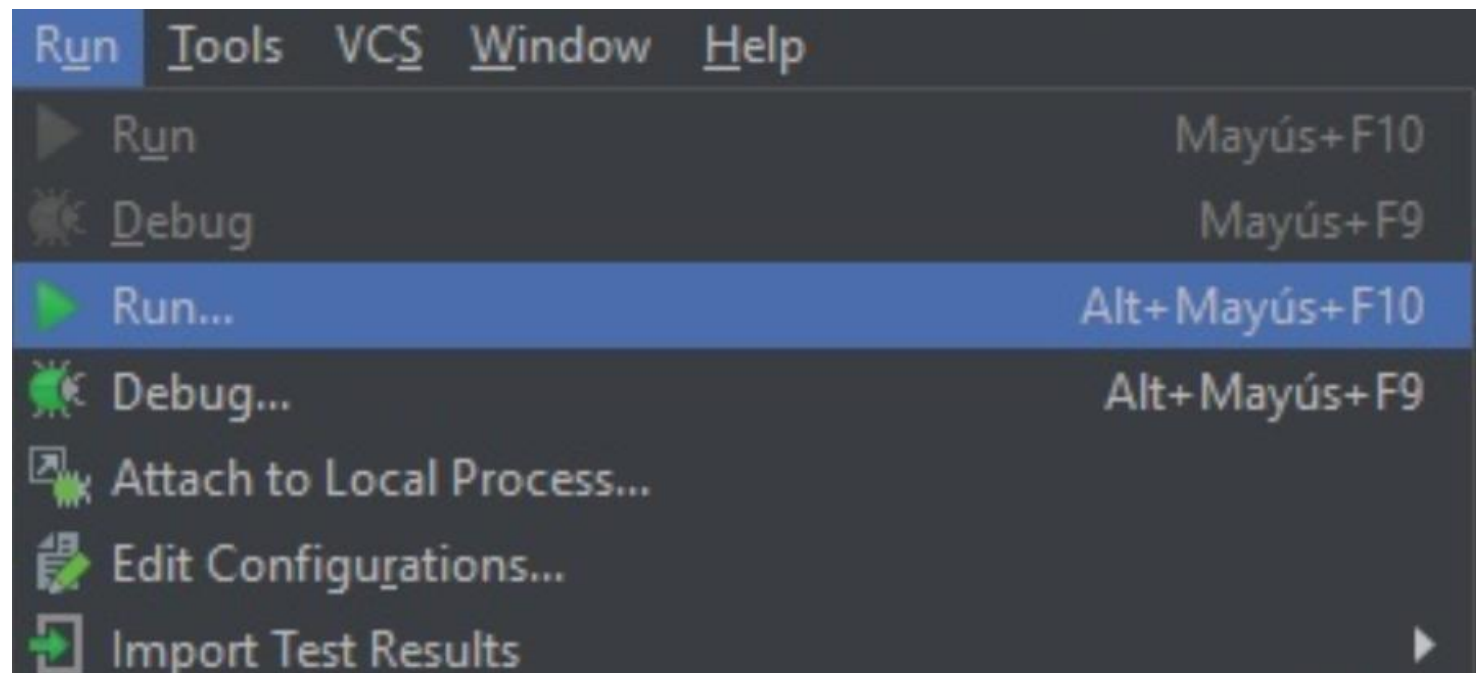
PyCharm - Crear Archivo .py



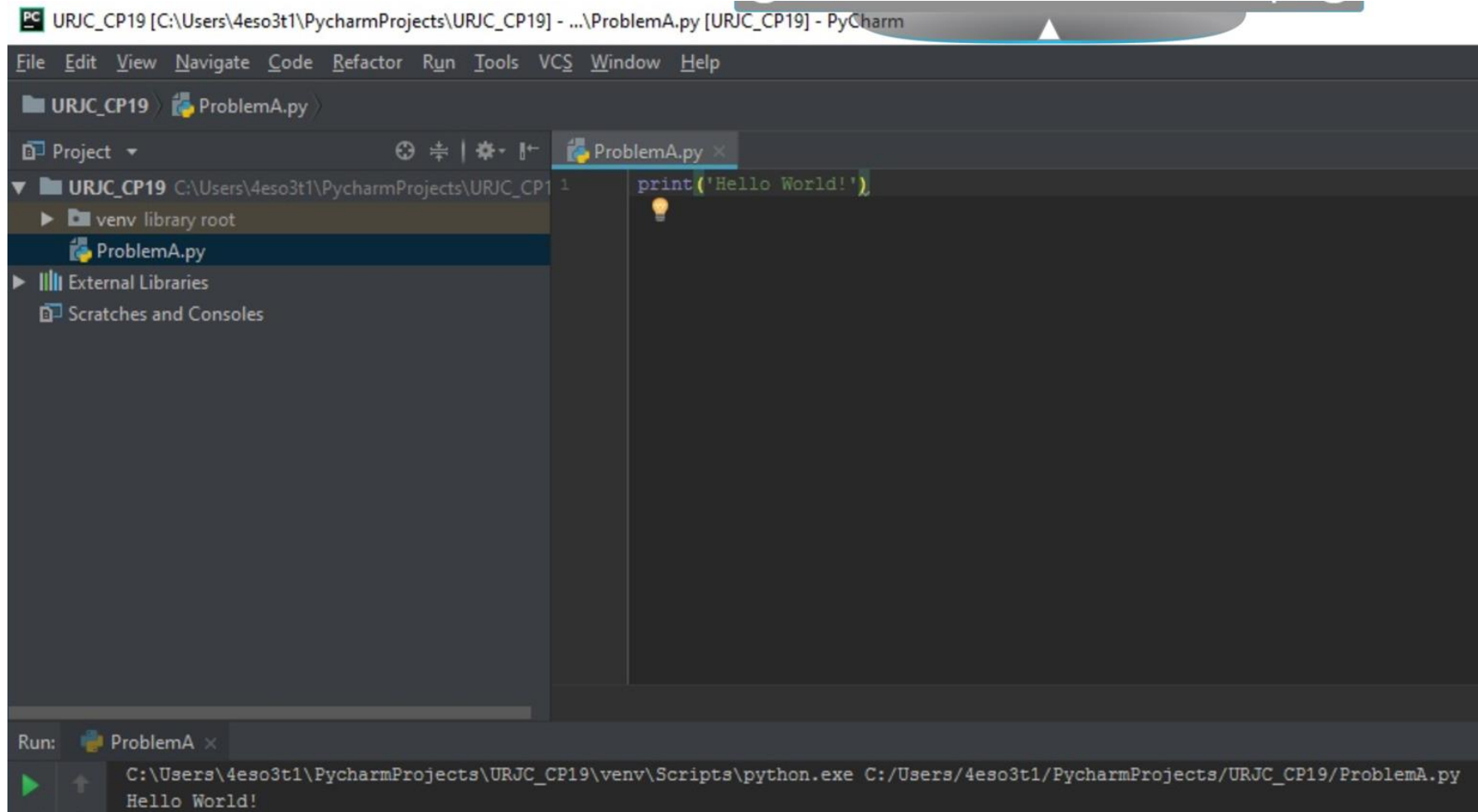
PyCharm - Primer Ejemplo!



```
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 espacio en memoria que almacena datos que utilizamos en un 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

```
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

- 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

- 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

- 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

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

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

- Condiciones: Si número 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

- 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

- 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

- 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

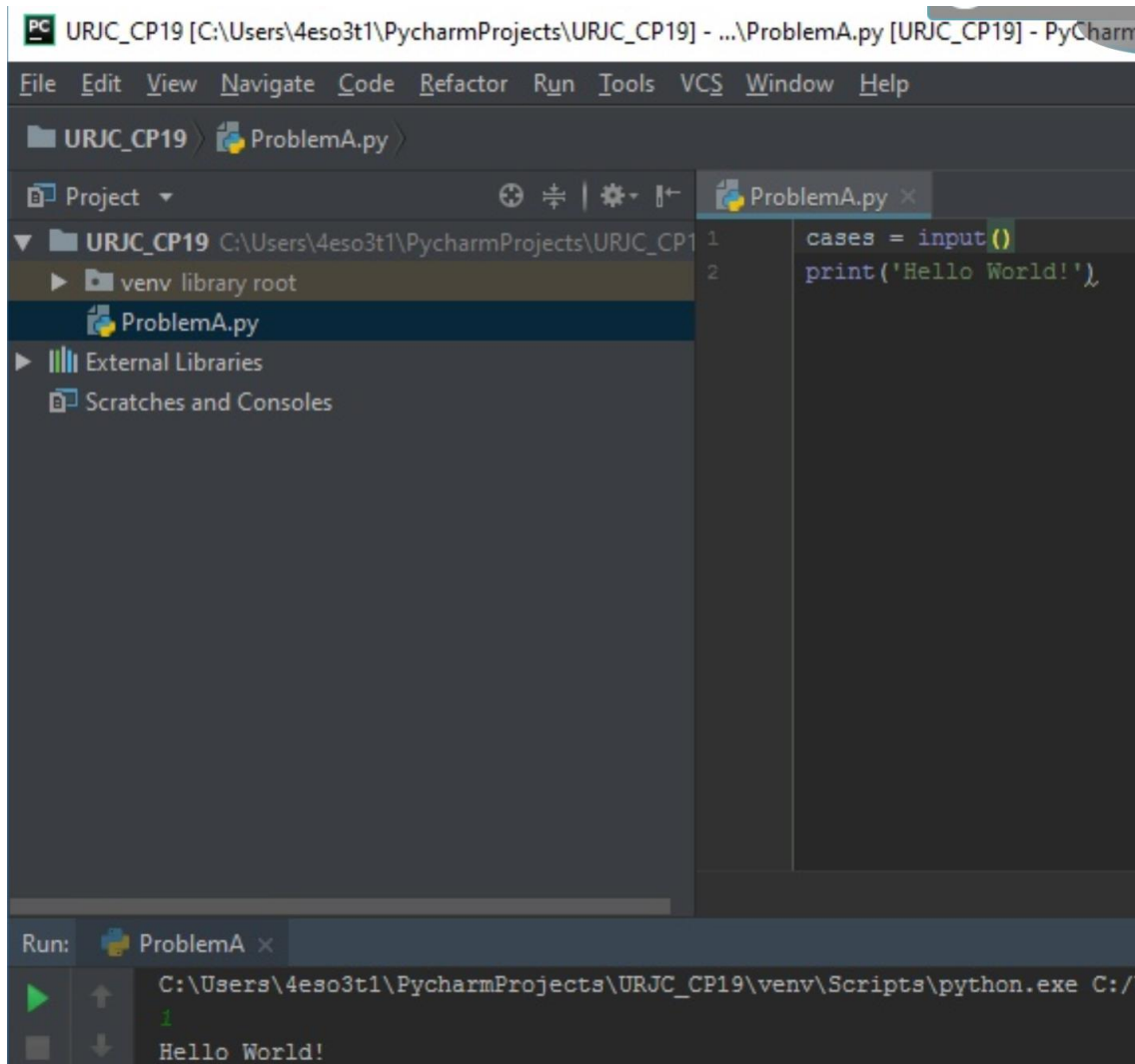
You are to write the most basic program; it should just output “Hello world!” on a single line, no matter what the input.

Sample input/output

Sample input and output for this problem:

Input	Output
1	Hello world!

Ejemplo Problema: ¡Hola mundo!

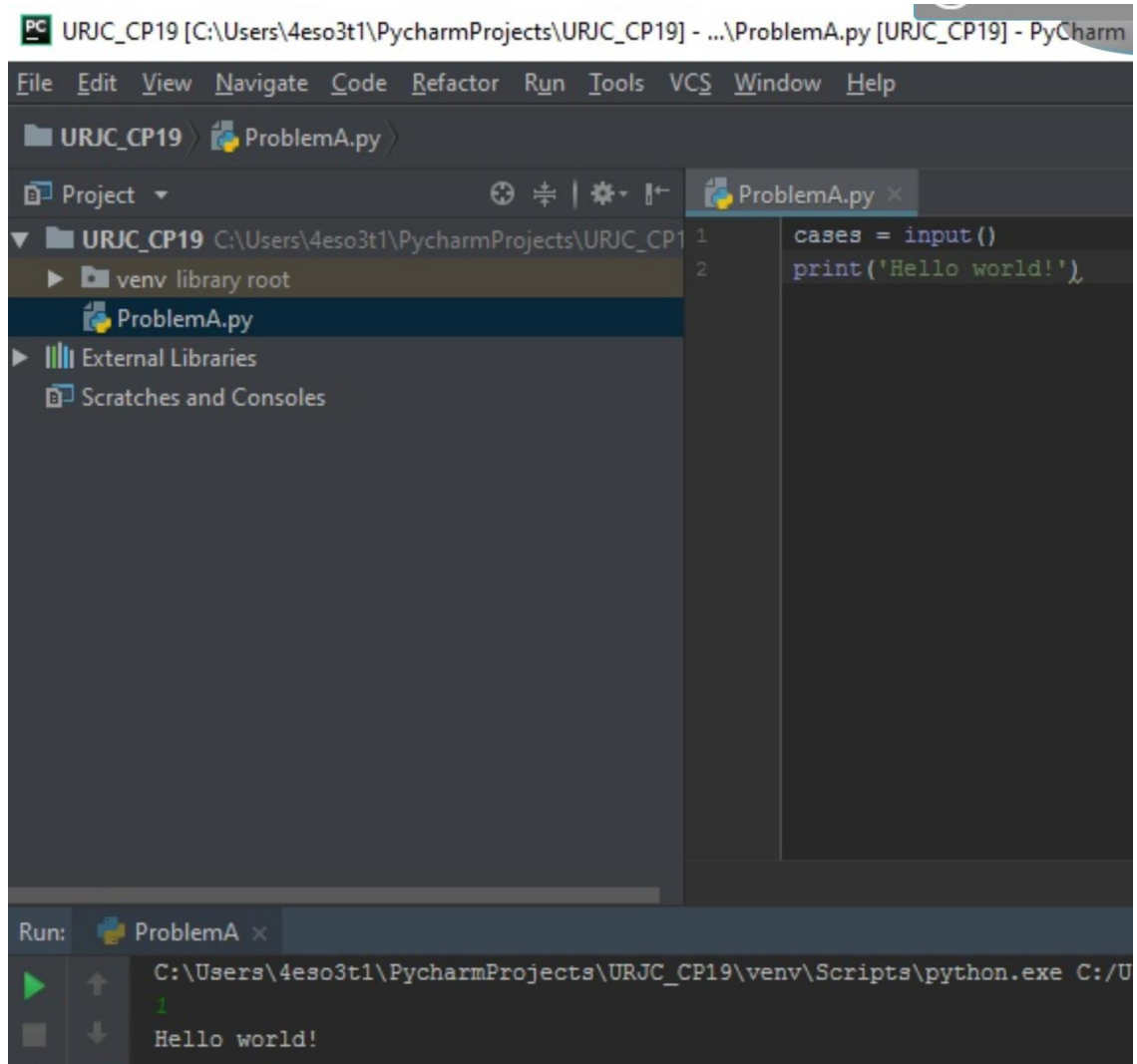


```
PC URJC_CP19 [C:\Users\4eso3t1\PycharmProjects\URJC_CP19] - ...\ProblemA.py [URJC_CP19] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
URJC_CP19 > ProblemA.py
Project
URJC_CP19 C:\Users\4eso3t1\PycharmProjects\URJC_CP19
venv library root
ProblemA.py
External Libraries
Scratches 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 image shows a PyCharm IDE window titled 'URJC_CP19 [C:\Users\4eso3t1\PycharmProjects\URJC_CP19] - ...\ProblemA.py [URJC_CP19] - PyCharm'. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the project structure with 'URJC_CP19' expanded, containing 'venv library root' and 'ProblemA.py'. The main editor displays the code for 'ProblemA.py':

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

The bottom panel shows the 'Run' output for 'ProblemA'. It displays the command: `C:\Users\4eso3t1\PycharmProjects\URJC_CP19\venv\Scripts\python.exe C:/U` and the output: `1` followed by `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!
```

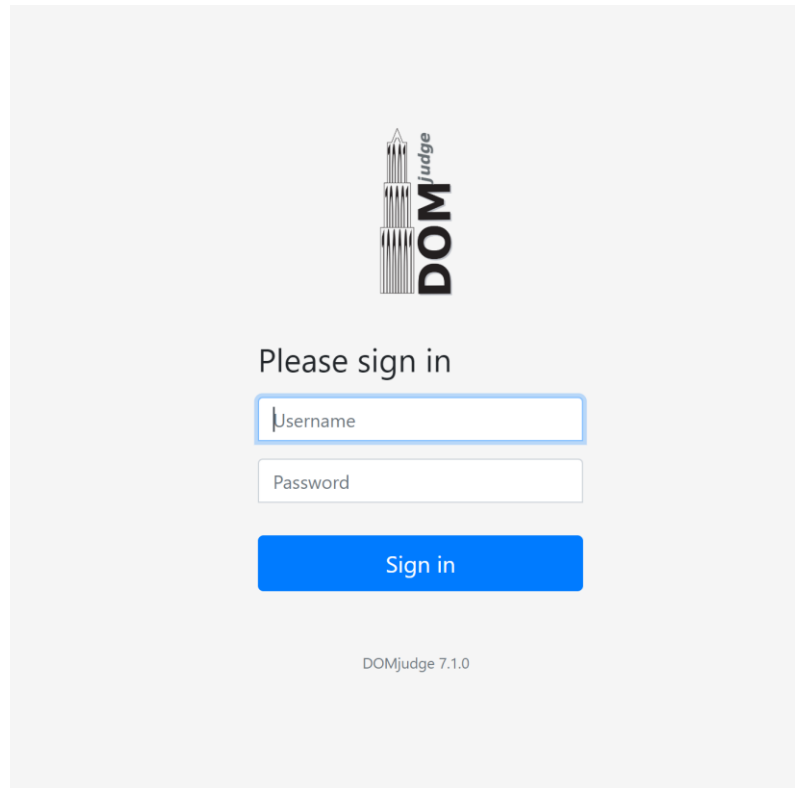
Importante la indentación (tabulaciones) además de recoger el elemento como un entero.

Domjudge

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

Domjudge

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



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: the first is labeled 'Username' and the second is labeled 'Password'. Below these fields is a blue button with the text 'Sign in'. At the bottom center of the page, the version number 'DOMjudge 7.1.0' is displayed.

Domjudge

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

[Submit](#)

[Logout](#)

[test_contest](#) 148d 17:19:42

RANK	TEAM	SCORE	ACONTRASEÑAS	ACTIVIDADES	CHERLOJOLMS	DARKSOULS	DORAENAMORADORA	DS_MOCHILA	FINALSTORY	GODOFWAR	HACKERMAN	MERGESORT	PILOTES	SEPTIEMBRE	SKYLINE
1	Equipo de pruebas	12 3761458		321420 1 try	328763 2 tries	309779 1 try	302468 1 try	321489 1 try	321422 1 try	311048 1 try	301186 1 try	311062 2 tries	302785 2 tries	320147 1 try	309829 1 try

Submissions

time	problem	lang	result
2020-03-02 17:30	CHERLOJOLMS	PY3	CORRECT
2020-03-02 17:29	CHERLOJOLMS	PY3	OUTPUT-LIMIT
2020-02-27 16:14	DS_MOCHILA	PY3	CORRECT
2020-02-26 17:02	DS_MOCHILA	PY3	CORRECT
2020-02-26 16:18	DS_MOCHILA	PY3	WRONG-ANSWER
2020-02-26 16:16	DS_MOCHILA	PY3	CORRECT
2020-02-26 15:09	FINALSTORY	JAVA	CORRECT
2020-02-26 15:09	FINALSTORY	PY3	CORRECT
2020-02-26 15:09	FINALSTORY	PY3	CORRECT
2020-02-26 15:07	ACTIVIDADES	PY3	CORRECT
2020-02-25 17:54	SEPTIEMBRE	JAVA	CORRECT

Clarifications

No clarifications.

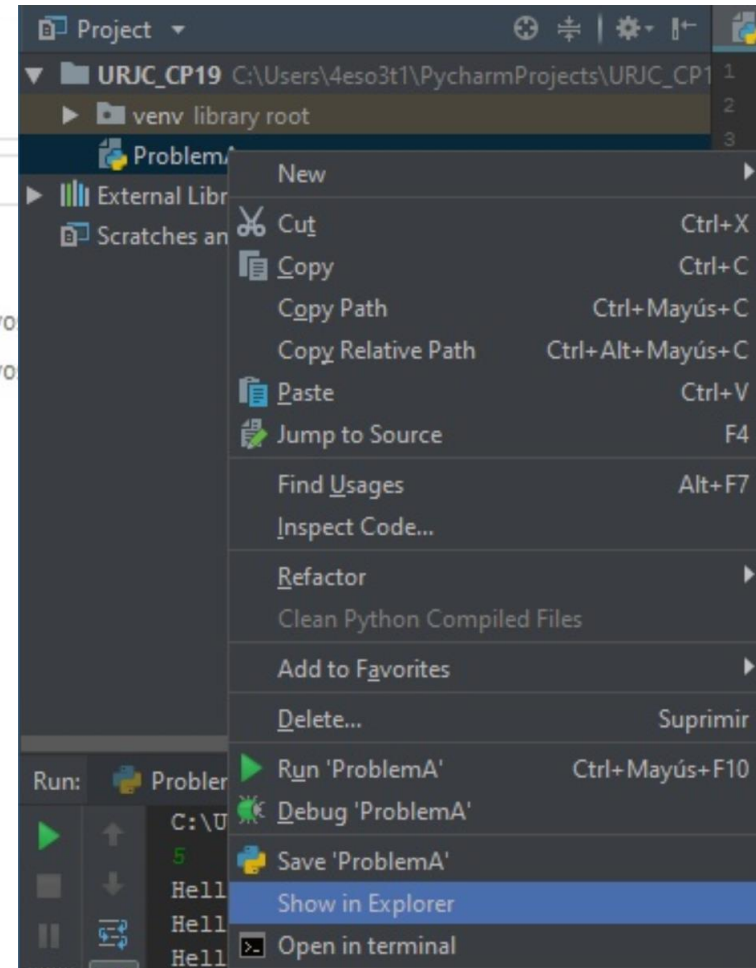
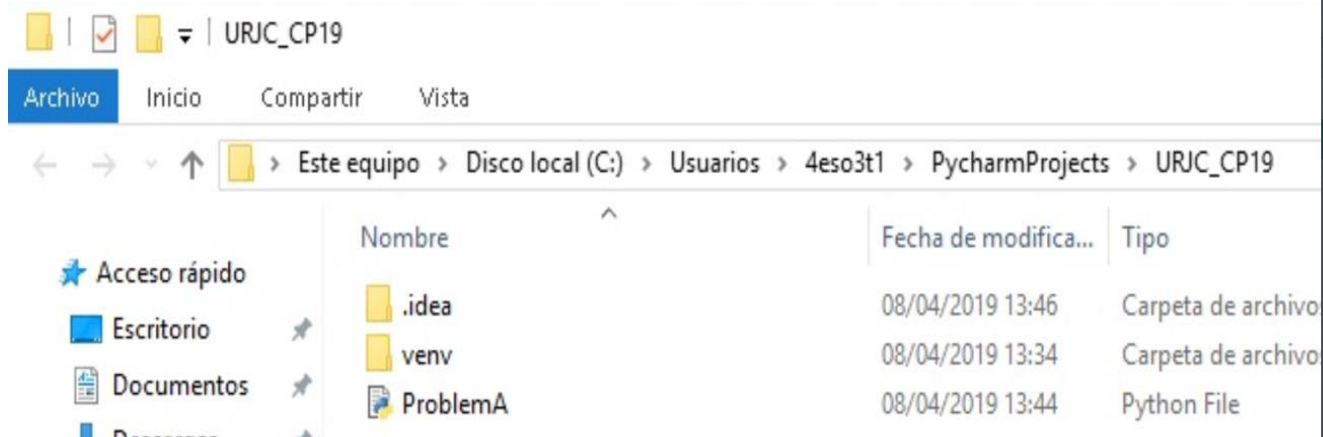
Clarification Requests

time	from	to	subject	text
2019-07-18 11:46	Equipo de pruebas	Jury	problem SN	No entiendo la entrada

[request clarification](#)



Domjudge - Seleccionar Archivo



C:\Users\NOMBRE_DE_USUARIO\PycharmProjects\NOMBRE_DE_PROYECTO

Domjudge




DOMjudge [judgehosts](#) [clarifications](#) [submissions](#) [rejudgings](#) [scoreboard](#) [team](#)

admin [AJ_GDDV_1](#) 156d 19:25:27

AJ_GDDV_1

starts: 2020-02-07 13:10 - ends: 2020-08-07 13:15

Filter

RANK		TEAM	SCORE	A 	B 	C 
1		m.sebastianv.2016_ajgddv Universidad Rey Juan Carlos	3 250	56 1 try	87 1 try	107 1 try
2		p.casas.2017_ajgddv Universidad Rey Juan Carlos	3 341	45 1 try	49 1 try	187 4 tries
3		j.lopezn.2016_ajgddv Universidad Rey Juan Carlos	3 392	108 1 try	34 1 try	210 3 tries
4		d.maria.2016_ajgddv Universidad Rey Juan Carlos	3 573	61 1 try	75 2 tries	377 3 tries
5		a.casadoce.2017_ajgddv Universidad Rey Juan Carlos	3 2713	1290 1 try	108 1 try	1315 1 try
6		f.domenech.2017_ajgddv Universidad Rey Juan Carlos	3 3914	1285 1 try	1296 1 try	1333 1 try
7		m.marquezb.2016_ajgddv Universidad Rey Juan Carlos	3 4246	1360 1 try	1425 1 try	1461 1 try
8		a.fernandezsan.2018_daagis Universidad Rey Juan Carlos	3 5113	4706 5 tries	130 1 try	177 2 tries



Contest problems

Problem A

MaquinaExpendedora
Limits: 1 second* / 512 MB

[problem text](#)

Samples

input #1

output #1

Problem B

Testigos
Limits: 1 second* / 512 MB

[problem text](#)

Samples

input #1

output #1

Problem C

Triangulares
Limits: 0.05 second* / 512 MB

[problem text](#)

Samples

input #1

output #1

¿Preguntas?



Solución de problemas - Hello World

```
N= int(input())  
for i in range(N):  
    print("Hello World!")
```

Solución de problemas - Victoria Magistral

```
N = int(input())  
for i in range(N):  
    n1,n2 = map(int,input().split())  
    print(n1//n2)
```

Solución de problemas - Hola Pepito

```
N= int(input())
for i in range(N):
    name,gen,name2,gen1 = input().split()
    if gen == "M":
        print("Hola Don " + name)
    else:
        print("Hola Donya " + name)
    if gen1 == "M":
        print("Hola Don " + name2)
    else:
        print("Hola Donya " + name2)
```

Solución de problemas - Contar

```
n = int(input())
cnt = 0
for i in range(n):
    name = input()
    if name == "AULA":
        cnt = cnt + 1
print(str(cnt) + " " + str(n-cnt) )
```