



Semana 5: Bucles

Programación

2020 – 2021

Grado en Ingeniería Informática

Doble Grado en Ingeniería Informática y Administración de Empresas

Ejercicio 1. Defina un programa que le permite simular el comportamiento de un cajero automático. Para ello:

- Tanto el pin, cuatro números entre 0 y 9, como la cantidad de dinero en la cuenta, que estará entre 50 y 5000 euros, se generarán aleatoriamente. Téngase en cuenta que el PIN se debe generar como cadena de texto, en caso contrario los ceros a la izquierda se perderán. Se puede comprobar su valor en el explorador de variables de Spyder para depurar correctamente el programa.
- Se solicitará al usuario el correspondiente código PIN. Dispone de hasta tres intentos para introducir el número correcto. En caso de agotar el número de intentos muestre un mensaje por pantalla y finalice la ejecución del programa.

Ejercicio 2. Extienda el Ejercicio 1 para que una vez se ha introducido correctamente el PIN, se muestre un menú por pantalla con las siguientes operaciones en el cajero:

Bienvenido

- 1- Ingreso efectivo
- 2- Retirada efectivo
- 3- Salir

Indique la operación a realizar:

Después de cada operativa indique por pantalla el saldo disponible en la cuenta. Si al solicitar una operación de retirada de efectivo se sobrepasa la cantidad disponible en cuenta se denegará dicha operación y se mostrará el mensaje correspondiente.

Ejercicio 3. Genere un número aleatorio entre 1 y 20 y, sin mostrarlo por pantalla, trate de averiguar su valor. A medida que introduzca valores el programa debe indicarle si el número introducido es mayor o menor del buscado y el número de intentos.

Ejercicio 4. Solicite por pantalla dos números y realice una multiplicación entre ellos. Para realizar el cálculo solicitado emplee únicamente operaciones de suma

Ejercicio 5. Cree un programa que le solicite al usuario un número. Calcule el número de números pares que hay comprendidos entre el 0 y el número introducido por el usuario.

Ejercicio 6. Muestre por pantalla todos los números comprendidos entre el 0 y 30 exceptuando los números primos.

Ejercicio 7. Pedir al usuario 2 números enteros A y B tales que $A + 5 < B$. Se debe seguir pidiendo hasta que el usuario introduzca los números correctos. A continuación se deben generar aleatoriamente e imprimir 5 números enteros en el intervalo [A,B], de tal forma que se alternen pares e impares. No es necesario que los números sean distintos, sólo la alternancia par-impar.

Ejemplo: para el intervalo [3,9] una posible secuencia válida es 6, 7, 6, 3, 4

Ejercicio 8. Partiendo del rango 1 a 10, solicite al usuario de forma consecutiva y en el mismo orden que se detalla la siguiente información:

- El número ¿es mayor o menor de 5? (mayor/menor)
- ¿Es par o impar? (par/impar)
- ¿Extremo superior o inferior? (superior/inferior)

De acuerdo a las respuestas obtenidas imprima por pantalla el número/s que cumpla dichas condiciones.

Ejemplo 1:

El número ¿es mayor o menor de 5? (mayor/menor) mayor

6 7 8 9 10

¿Es par o impar? (par/impar) par

6 8 10

¿Extremo superior o inferior? (superior/inferior) superior

10

Ejemplo 2:

El número ¿es mayor o menor de 5? (mayor/menor) menor

1 2 3 4

¿Es par o impar? (par/impar) impar

1 3

¿Extremo superior o inferior? (superior/inferior) inferior

1

Ejercicio 9. Escribir un programa que genere e imprima los N números perfectos menores de un cierto número introducido por teclado. Un número es perfecto si es igual a la suma de todos sus divisores positivos sin incluir el propio número. Por ejemplo el número 6 es perfecto ya que tiene como divisores: 1, 2, 3 y 6 (pero el 6 no se cuenta como divisor para comprobar si es perfecto). Si sumamos $1 + 2 + 3 = 6$. Ejemplo:

Indica el límite superior para generar números perfectos y pulsa Enter:

10000

El número 6 es perfecto

El número 28 es perfecto

El número 496 es perfecto

El número 8128 es perfecto

Ejercicio 10. Como sabréis, para leer un número por teclado en Python hay que usar `input` y luego convertir la cadena leída a entero o flotante. De esta forma podremos operar con el número leído. Pero si recibimos algo que no sea un número el programa fallará. Una forma de resolver este problema es usar la función `isdigit()` del tipo `str`, que devuelve `True` si la cadena es un número (por ejemplo `'342432'.isdigit()` devuelve `True`). Esto funciona para los enteros, pero no para los flotantes puesto que el punto decimal no se considera un dígito (`'343.32'.isdigit()` devuelve `False`). Crear un programa que pide un número por teclado y continúa pidiéndolo hasta que lo que recibe es un número. Una vez recibido debe imprimir su cuadrado. El programa debe funcionar tanto para enteros como para flotantes.

Normas de entrega

Los ejercicios se deben **subir a Aula Global** hasta las 7:00 de la mañana del día de la siguiente clase de laboratorio. Se deberá subir un fichero comprimido **zip** con un fichero para cada programa, nombrados ejercicio1.py, ejercicio2.py, etc. El nombre del fichero será “s5-iniciales-del-alumno.zip” (por ejemplo Lucía Pérez Gómez subirá un archivo llamado s5-lpg.zip)