**EJERCICIOS DE REPASO Y PREPARACIÓN PT1**

**Instrucciones generales**

* No se podrán utilizar funciones, listas o métodos no explicados en clase. No se evaluarán las líneas de código en las que se usen.
* Se entregará un archivo en formato .py con el enunciado copiado y a continuación el ejercicio realizado.
* Se valorará NEGATIVAMENTE el uso innecesario de variables y estructuras de control (Optimizar código).

1. Solicitaremos al usuario un número mayor que 1 y menor que 20. Este número serán los bloques. Si el número no cumple estos requisitos, volveremos a solicitarlo.

Haremos una pirámide usando el número de bloques como los bloques de la base. En cada piso habrá un bloque menos que en el anterior. Tras mostrar la pirámide, el programa nos indicará cuantos bloques se han utilizado.

- Ejemplo de pirámide para 6 bloques:  
  
\*  
\*\*  
\*\*\*  
  
- Ejemplo de pirámide para 10 bloques:  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
  
Mostrar la pirámide anterior de forma inversa.  
- Ejemplo:

\*\*\*  
\*\*  
\*

1. El programa solicita 4 múltiplos de 3 por teclado e imprime el cuadrado de cada elemento por pantalla.
2. Se quiere jugar al juego “adivina el número”, en el programa se define una constante de dos dígitos, el usuario tendrá cinco oportunidades para adivinarlo e informará al usuario de las oportunidades que le quedan indicando si el valor a buscar es mayor o menor que el introducido. Si el usuario lo adivina le pondrá nota dependiendo de la cantidad de veces que ha jugado finalizando el programa sin consumir los intentos restantes e indicando el número de intentos que se han utilizado. (Cada jugada vale 2 puntos)

Nota: Obligatorio utilizar estructura principal con while

1. Programa que muestra un menú con las siguientes operaciones:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1-Tabla interés de un capital

2-Salir

* 1. Opción 1: Imprime la lista de intereses producidos (I) y el capital acumulado anualmente (CA), por un capital inicial ‘C’, impuesto con un rédito o interés ‘R’ durante ‘N’ años.

Ejemplo: para un capital inicial de 1000 y un rédito del 10% durante 5 años.

|  |  |  |  |
| --- | --- | --- | --- |
| AÑO | %r | I | CA |
| 1 | 10 | 100 | 1100 |
| 2 | 10 | 110 | 1210 |
| 3 | 10 | 121 | 1331 |
| 4 | 10 | 133,1 | 1464,1 |
| 5 | 10 | 146,41 | 1610,51 |
|  |  |  |  |

1. Algoritmo que escribe en orden decreciente los 100 primeros números naturales (un número en cada línea)
2. Algoritmo que escribe en orden decreciente los 100 primeros números pares (un número en cada línea)
3. Algoritmo que pide al usuario su nombre y un número entero mayor que 1, y escribe su nombre tantas veces como indique el número. Ej. Si la entrada es

V

3

La salida será

V V V

1. Algoritmo que pide dos números al usuario y escribe todos los números entre el primero y el segundo. Se debe tener en cuenta los casos en que el primer número sea más grande o pequeño que el segundo.
2. Algoritmo que pide un número al usuario y escribe la suma de todos los números entre el introducido y el 1. Ej.

Si la entrada es

4

La salida será

10

Dado que 10 = 1 + 2 + 3 + 4

1. Algoritmo que pide al usuario, la edad y el año actual. El algoritmo escribirá la edad que tenía el usuario en cada año desde su nacimiento y se despedirá diciendo Adiós <edad actual>Ej.

Si la entrada es

5

2003

El programa escribirá

En 1998 nació

En 1999 tenía 1 año

En 2000 tenía 2 años

En 2001 tenía 3 años

En 2002 tenía 4 años

Adiós tienes 5 años

1. Algoritmo que va leyendo números enteros desde el teclado hasta que se introduzca un número múltiplo de 7. La salida por pantalla será la suma de todos los números impares y positivos leídos sin contar el último.
2. Programa que lee ‘n’ números y cada vez que se introduce un número par y positivo muestra por pantalla el mensaje: “el número introducido es par y positivo”. El primer número introducido por teclado indica el número de números que se van a introducir.
3. Algoritmo que tiene como datos de entrada (por teclado) la fecha de nacimiento de 10 personas (cada fecha consta de 3 números enteros: día, mes y año), y como dato de salida el número de personas que nacieron entre el 1-1-1980 y el 31-12-2000.
4. Determinar cuántos años no bisiestos hay comprendidos entre 1 y n, siendo n un número entero entrado por teclado.
5. Determinar si un número entero entrado por teclado es primo o no. La entrada será un número entero y la salida será “PRIMO” o “NO PRIMO”.
6. Mostrar los números primos que hay entre 1 y n, siendo n el dato de entrada. Debe mostrar cada número primo en una línea.
7. Programa que muestra la lista de los ‘n’ primeros números primos, siendo ‘n’ el dato de entrada. Debe mostrar cada número primo en una línea.
8. Programa que muestra las soluciones de una ecuación de segundo grado del tipo: ax2+bx+c=0. Los coeficientes a, b y c deben ser datos reales leídos por teclado.

x=

1. Programa que obtenga y muestre el producto de dos números enteros positivos entrados por teclado mediante sumas sucesivas.
2. Programa que obtenga el cociente y el resto de dos números enteros positivos entrados por teclado mediante restas. Primero se pide el dividendo y después el divisor.
3. Programa que pide al usuario un número positivo mayor que 0. El programa escribirá una línea por número entre el 1 y el número introducido (es decir, crecientemente). En cada línea escribirá todos los números desde el número correspondiente a la línea hasta el 1. Ej.

Si la entrada es

5

La salida será

1

2, 1

3, 2, 1

4, 3, 2, 1

5, 4, 3, 2, 1

1. Programa que pide al usuario un número positivo mayor que 0. El programa escribirá una línea por número entre el número introducido y el 1 (es decir, decrecientemente). En cada línea escribirá todos los números desde el 1 hasta el número correspondiente a la línea. Ej.

Si la entrada es

5

La salida será

1, 2, 3, 4, 5

1, 2, 3, 4

1, 2, 3

1, 2

1

1. Algoritmo que pide un número de pirámides p y otro número n del 1 al 9 y escriba lo siguiente (suponiendo que p es 2 y n es 9)

.........9.........

........898........

.......78987.......

......6789876......

.....567898765.....

....45678987654....

...3456789876543...

..234567898765432..

.12345678987654321.

0123456789876543210

........9.........

........898........

.......78987.......

......6789876......

.....567898765.....

....45678987654....

...3456789876543...

..234567898765432..

.12345678987654321.

0123456789876543210

1. Tenemos dos listas: antonio = [0, 5, 4, 2, 10] y manu = [2, 5, 3, 5, 1]

El primer valor es la nota de la Pt1, el segundo de la Pt2, etc.

Queremos calcular la media de nota de la asignatura de cada uno de nuestros alumnos.

Salida:

Antonio. Nota media: \_\_

Manu. Nota media: \_\_

Compararemos cada una de las notas e indicaremos quien de los 2 tiene la nota más alta en cada una de las pt.

Salida:

Pt1 – manu – 2

Pt2 – Empate

Pt3 – antonio – 4

Etc.

1. Crearemos una lista de números aleatoria de 1.000 números del 0 al 9.

Nota:

import random

r1 = random.randint(0, 11) # Numeros aleatorios entre el 0 y el 5

Ordenaremos la lista y contaremos el número de veces que aparece cada número.

Salida:

Lista ordenada:

El 0 aparece \_\_ veces

El 1 aparece \_\_ veces

Etc.