**Estadísticas de CCE**

Esta actividad está compuesta por 17 ejercicios muy pequeñitos y guiados para aquellos alumnos que han estudiado menos, han entrenado menos su resiliencia y **serán estupendos programadores,** quizá algo más lentos, pero **seguros.** No es tan importante el ritmo como la calidad. Vamos indicando, además, el apartado del libro: “Resolución de problemas con Python estructurado”, el apartado que debes estudiar antes de hacer la tarea indicada.

Poco a poco vamos a construir una tabla con estadísticas de la Comunidad Económica Europea. Partimos de los siguientes datos localizados en internet:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **País** | **PIB** Millones€ | **Población** personas |
| 1 | Alemania | 3.144.050 | 82.605.000 |
| 2 | Francia | 2.228.857 | 64.770.000 |
| 3 | Reino Unido | 2.393.134 | 65.893.000 |
| 4 | Italia | 1.680.523 | 60.721.000 |
| 5 | España | 1.118.522 | 46.427.000 |

La tabla contiene el nombre, el producto interior bruto y la población de 5 países de la CEE. Para probar nos basta. Cuando la tengamos hechos los programas completaremos los datos:

1. Repasa la construcción de secuencias (apartado 6.1.1 del libro) y construye una tupla constante que contenga el nombre de los 5 primeros países de la lista (es decir que el nombre de los 5 estén en una única variable). Recuerda el convenio de nombres: Las constantes van en mayúsculas.
2. Repasa la sintaxis y semántica de la creación de subprogramas (apartado 4.4.1) y la del bucle definido (apartado 6.2.1) y haz un subprograma que liste estos nombres. Cada uno de los países en un reglón. El subprograma recibe la variable que contiene todos los nombres. En el apartado 4.4.2 se explica cómo documentar un subprograma. Deja una línea en blanco separando cada pieza (sección de declaración de constantes, cada subprograma, el cuerpo ejecutivo, etc.).
3. Repasa la llamada a procedimientos y funciones en el apartado 4.3.1 y haz un probador que ejecute el subprograma anterior. A partir de aquí harás siempre un probador para el subprograma que estás haciendo y no pasarás al siguiente subprograma hasta que el que tienes entre manos está perfecto. Los probadores se escriben justo detrás del subprograma que prueban. Van precedidos por “#PROBADOR” e incluyen los casos de prueba. Deben ser cómodos para el programador. Cuando hayas probado un subprograma, deshabilita el probador encerrándolo entre triples comillas sencillas ‘’’.
4. Haz un subprograma que devuelva (no que imprima) una cadena con el nombre del país que ocupa la posición pos, donde pos y la tupla de nombres son argumentos de entrada. Recuerda que las personas, a diferencia de Python, empezamos a contar por el 1. Como es habitual, indica si hay precondiciones.
5. Estudia el operador in (apartado 6.1.3) y, usándolo haz un subprograma que indique si un país (digamos España) está entre los cinco primeros. Simplemente ha de devolver True o False. Prueba uno que esté y otro que no.
6. Construye una estructura de datos que contenga (en una sola variable) los datos del PIB y en otra la población de los 5 primeros países de la CEE en el mismo orden que la tupla de los nombres de países. Estos datos cambian cada año, de modo que no son una constante. Elige nombre adecuado para cada una.
7. Repasa el condicional simple (apartado 5.1) y el esquema contador (apartado 6.2.1). Tras lo cual, haz un subprograma que cuente cuantos entre los países tienen más de un número de habitantes indicado como argumento. Date cuenta de que no necesitas usar la función range.
8. Junto al esquema contador has visto el esquema acumulador o totalizador. Haz un subprograma que devuelva la población total de todos (por tanto un for) los países de la CEE. Tampoco necesitas range.
9. Haz un subprograma que construye una tabla con el nombre de cada país junto a su población. Para ello has de recorrer las dos tuplas a la vez, y por tanto necesitas un contador para que selecciones los valores que están en la misma posición en ambas. Estudia el apartado 6.2.2 que te explica cómo generar un contador. Por si en un futuro hay que considerar mayor número de países, no uses el número 5, si no len(tupla). Alternativamente, crea una constante que indique el número de elementos de la tupla. Hazlo así con cualquier “constante” que pueda variar en un futuro. Como es una constante, puedes hacer un uso global de ella, pero acuérdate de documentarlo como una PRE:. Para controlar el formato de la tabla estudia los patrones de formato en el apartado 3.3.1 del libro. Ponle un rótulo a la tabla para que quede vistosa. Por ejemplo:

listado de paises

===================

PAIS POBLACION

Alemania 30100

Francia 28800

Reino Unido 27600

Italia 26200

España 26500

Recuerda que si un subprograma toca la interfaz de usuario (excepto que pida datos) no tiene return. Revisa la convención de nombres de subprogramas en el apartado 4.4.3.

1. Sin embargo, es costumbre ajustar los campos alfabéticos a la derecha y los numéricos a la izquierda. Para ello, después de imprimir el nombre del país, imprime ‘ ‘\*n, donde n es el número de blancos que necesitas

PAIS POBLACIÓN

Alemania 82605000

Francia 64770000

Reino Unido 65893000

Italia 60721000

España 46427000

1. Modifica el subprograma del apartado 10 para que añada una nueva columna a la tabla que contendrá el porcentaje de población de cada país respecto al total de los cinco (el porcentaje es: población del país/población total\*100) reusa el subprograma que hiciste en 9.
2. Añade la columna del PIB a la tabla
3. Probablemente te ha salido muy largo el print. Procura que quepan todas las instrucciones en la pantalla, sin necesidad de desplazar la pantalla. Puedes leer como se hace en el apartado 5.5.2.
4. La renta percápita de un país es el PIB/número de habitantes. Haz un subprograma que devuelva la renta per cápita media de la CEE. Si piensas que debes poner un for, es que no has pensado suficiente. Pensar antes ahorra mucho trabajo. Advierte que el PIB está en millones.
5. En el ejercicio 5 empleaste el operador in de Python. La mayoría de los lenguajes no tienen equivalente, de modo que repasa el bucle while (apartado 7.1.1). Tu while tendrá dos condiciones. Fuera del bucle te plantearás por cuál de las dos has salido. Realmente no necesitas un if.
6. Recorre la tupla de poblaciones para saber si hay al menos un país que tenga menos población que una determinada. En cuanto haya una que la tenga ya no has de recorrer la tupla. OJO con salirte de la secuencia. Diseña los casos de prueba, necesitas 2. Revisa el esquema “bucle con condición doble y discriminación a la salida” (apartado 7.1.3). Es muy parecido al anterior.
7. Resuelve este mismo problema con la técnica del centinela (apartado 7.1.3). Esta técnica evita la doble condición en el while. Esta alternativa es rentable en listas largas, especialmente si la probabilidad de que exista el elemento buscado es pequeña. No es rentable en tuplas. Para ello, duplica la tupla de poblaciones, esta vez como lista, añade al final un elemento que cumpla la condición. Estudia los servicios específicos de listas en el apartado 6.15.
8. En el ejercicio anterior habrás usado append(). En el probador, añade imprimir la lista y su longitud. ¿Tiene los mismos elementos que antes de hacer la llamada? ¿Qué ha pasado? ¿Pasaría lo mismo con tuplas? Corrige el ejercicio de listas para que no tenga efectos laterales.
9. Vamos a generalizar este trabajo. Revisa cómo podemos crear un bucle 1 a n en Python ya que no tiene “DO WHILE” (apartado 7.1.5). Haz un subprograma que pida al usuario el nombre de un país, podrá introducir tantos como desee, hasta que el usuario introduzca “un país” llamado “fin”. El subprograma devuelve una única estructura que contendrá los nombres de los países, similar a la que tú creaste en el ejercicio 1. Es decir, una tupla, no una lista. Por supuesto, no se almacena el país “fin”.
10. Revisa la gestión de excepciones (apartado 5.6.1) y modifica el ejercicio anterior para que el usuario, tras introducir un nombre de país (distinto de “fin”) introduzca su población y su PIB. El subprograma devolverá 3 tuplas en que los datos correspondan a los países en el mismo orden. Advierte que el subprograma ha de pedir n+1 nombres de país, pero solo n poblaciones y pibs, de modo que usa un solo bucle while y pides antes un país.

Para desarrollar los subprogramas escribiste unas tuplas de prueba. Comprueba que puedes llamarlos también con las creadas con los datos que introdujo el usuario. Ahora vas a construir el cuerpo ejecutivo de tu programa.

1. Un grupo de científicos locos está convencido de que hay un número mágico, p (de población) y otro q (de PIB), por debajo de los cuales un país está en riesgo de consanguinidad y pobreza respectivamente. Introduce las constantes P=20000.0, y Q=1500000.0 en la sección del programa que le corresponde. Estudia en orden de las secciones de un programa en la tabla 4.1 del libro.
2. El cuerpo ejecutivo va detrás de todas las definiciones. Haz un programa que pinta la tabla de las estadísticas de la CEE tal como muestra la figura, pidiéndole al usuario los datos de los países, hasta introducir “fin” en nombre de País. Por cada país, el usuario introducirá nombre, población y PIB. Tras lo cual, indica si (no) hay riesgo de consanguinidad y si (no) hay riesgo de pobreza. Se desea también saber cuántos países tienen PIB mayor que la mitad del primero de ellos, indicando el nombre de éste.
3. Todo archivo.py comienza con la documentación. Como mínimo, objetivo y posiblemente nombre del programador y contexto.

**Enunciado de examen**

En versión enunciado de examen podría quedar así:

Un grupo de científicos está convencido de que hay un número mágico, p =20000.0 (de población) y otro q=1500000.0 (de PIB), por debajo de los cuales un país está en riesgo de consanguinidad y pobreza respectivamente. Desean un programa que pide al usuario los datos de los países, hasta introducir “fin’ en nombre de País. Por por cada país el usuario introducirá nombre, población y PIB. Tras mostrar la tabla de resumen estadístico de una serie de países, indica si (no) hay riesgo de consanguinidad y si (no) hay riesgo de pobreza. Se desea también saber cuántos países tienen PIB mayor que la mitad del primero de ellos, indicando el nombre de este. Tal como se muestra a continuación:

PAIS POBLACIÓN %POBLAC PIB

320416000.0

Alemania 82605000 0.26 3144050.00

Francia 64770000 0.20 2228857.00

Reino Unido 65893000 0.21 2393134.00

Italia 60721000 0.19 1680523.00

España 46427000 0.14 1118522.00

TOTAL 320416000 10565086.00

hay riesgo de consanguinidad

no hay riesgo de pobreza

5 países tienen PIB mayor que Alemania

EJEMPLO DE SALIDA