



501125

## Introducción a la Programación Laboratorio Programación Python Nivel Certamen

JAVIER VIDAL VALENZUELA

9 de mayo de 2023

- 1.- **Divertilandia** Un parque de diversiones tiene diferentes tarifas de entradas. Si el cliente es menor de 4 años puede entrar gratis, si tiene entre 4 y 18 años debe pagar \$5000, si es mayor de 18 años y menor de 65 paga \$10000 y si es adulto mayor (65 años o más) paga \$3000. Realice un programa Python donde debe ingresar el número de integrantes de una familia  $N$  ( $N > 0$ ) seguido de las edades de los  $N$  integrantes de ésta. El programa debe desplegar el valor total a pagar para entrar al parque.

Entradas: El programa tiene  $N + 1$  entradas. La primera entrada corresponde al valor de  $N$ , ( $N > 0$ ), y luego vienen  $N$  valores enteros mayores que cero y menores que 90 correspondientes a las edades de los clientes.

Salidas: La única salida es el valor a pagar por las  $N$  personas.

Ejemplo de Entradas: 6 2 10 18 33 45 76

Ejemplo de salida: \$33000

- 2.- **Carga de barco** El Terminal Internacional Puerto San Vicente recibe barcos con distintas capacidades de carga, los cuales deben transportar diferentes tipos de productos, las que tienen distinto peso. Para apoyar la tarea de carga evitando que un barco sea sobrecargado se pide, que escriba un programa al que se ingrese un valor entero  $p$ , correspondiente a la capacidad máxima de carga de un barco. A continuación se debe ingresar un conjunto de números enteros positivos, correspondientes al peso de cada producto llevado al barco, hasta que su suma supere el valor  $p$ .

Entradas: El programa tiene  $m$  entradas. La primera entrada corresponde al valor de  $p$ , ( $p > 0$ ), y luego vienen  $m - 1$  valores enteros positivos.

Salidas: La suma de los  $m - 2$  valores ingresados antes de que se supere el valor  $p$ .

Ejemplo de Entradas: 30 10 5 6 7 7

Ejemplo de salida: 28

- 3.- **Perímetro de un polígono** Desde pequeños aprendimos que el perímetro de un polígono corresponde a la suma de todos sus lados. Ayude a su propio yo del pasado creando un programa en Python que calcule el perímetro de un polígono cualquiera. El programa primero recibirá la cantidad de puntos que tiene el polígono, seguido de una línea por cada punto del polígono, la que contiene dos números enteros: uno para la coordenada  $x$  y otro para la coordenada  $y$ . Asuma que existe una línea recta entre cada par de puntos consecutivos y que ninguna de las líneas se cruza, además, asuma que existe una línea entre el último y el primer punto ingresado, con la cual se cierra el polígono. Recuerde que la distancia de dos puntos  $p$  y  $q$  está dada por la fórmula  $d_{pq} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$ .

Entradas: El programa tiene  $2 * n + 1$  entradas. La primera entrada corresponde al valor de  $n$ , ( $n > 0$ ), correspondiente al número de lados del polígono. Luego vienen  $2 * n$  valores enteros correspondientes a las coordenadas  $x$  e  $y$  de los vértices del polígono.

---

Salidas: La única salida del programa es un valor real correspondiente al perímetro del polígono.

Ejemplo de Entradas:

```
3
0.0  0.0
4.0  0.0
0.0  3.0
```

Ejemplo de salida: 12.0

- 4.- **Turistas en el Parque Huacachina** La agencia de turismo municipal de la ciudad de Ica, en el Perú, montó un puesto de control de jeeps de aventura que suben a las dunas del parque Huacachina. Como durante el día, son varios los off-roads que suben y descienden del parque nacional, y no siempre los turistas usan un mismo transporte para la ida y vuelta, el ayuntamiento necesitaba tener un mejor control y seguridad sobre el flujo de visitantes en el parque. Desarrolle un programa Python que reciba como entrada si un jeep está entrando o volviendo del parque y la cantidad de turistas que este vehículo está transportando. Al final del turno, el programa debe indicar la cantidad de vehículos y de turistas que permanecen dentro del parque.

Entrada: El programa debe recibir sucesivos pares de entrada. Cada par debe indicar el movimiento del jeep y la cantidad de turistas que este está transportando. La primera entrada debe ser "ENTRADA" o "SALIDA". "ENTRADA" indica que el jeep está ingresando en el parque y "SALIDA" que el jeep está retornando del paseo. Inmediatamente después, el programa recibe un entero  $T$  (donde,  $0 \leq T \leq 20$ ) que indica la cantidad de turistas que están siendo transportados por el jeep. La cadena "FIN" es el indicador de fin de procesamiento.

Salida: El programa debe presentar dos salidas: la cantidad de jeeps y la cantidad de turistas que aún quedan por volver del parque.

Observación: Incluya en su programa una función para leer y validar los valores recibidos por el programa

- 5.- **Secuencia "Look and Say"** Los problemas de secuencias numéricas son clásicos en las matemáticas recreativas. La secuencia de números enteros "look and say" fue introducida y analizada por el matemático John Conway. Esta secuencia tiene la forma: 1, 11, 21, 1211, 111221, 312211, .... Su construcción es sencilla, cada término describe al anterior:

- 1 contiene "un uno", o sea, 11;
- 11 contiene "dos unos", o sea, 21;
- 21 contiene "un dos y un uno", o sea, 1211;
- 1211 contiene "un uno, un dos y dos unos", o sea, 111221;
- 111221 contiene "tres unos, dos doses y un uno", o sea, 312211; y así se puede continuar sucesivamente

En esta secuencia solo se emplean los números 1, 2, y 3. Otra curiosidad de esta secuencia observada por Conway es que muestra similitud formal con el comportamiento de los elementos químicos.

Pero, ¿qué ocurre si en lugar de iniciar con un 1 se inicia con otros valores?

Escriba un programa en Python que dado un valor  $n$  y  $v$ , determine  $n$  términos de la secuencia "look and say" iniciándose en un valor inicial  $v$ .

Entrada: Este programa tiene dos entradas, un valor  $n$  ( $0 < n \leq 20$ ) y un valor  $v$  ( $0 \leq v < 10$ ).

Salida: El programa debe desplegar  $n$  valores de la secuencia de Conway.

Ejemplo de Entradas: 7 5

Ejemplo de salida:

---

5  
15  
1115  
3115  
132115  
1113122115  
311311222115