### Puntos obtenidos: 96pts

| Apartado | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Puntaje | 15 | 10 | 15 | 5 | 10 | 10 | 15 | 20 |
| Obtenido | 12 | 10 | 15 | 5 | 10 | 10 | 12 | 20 |

### Observaciones: Muy bien! A continuar trabajando así.

| 1. Hacer una función que permita ingresar manualmente los valores de compra del producto PlayStation 5 a una Pila.  El dato ingresado por el usuario debe estar entre 500 y 650 USD, Si es válido, apilar el precio; si no, repetir la lectura hasta recibir un dato correcto o hasta que la pila se llene.  Recordar que la pila tiene un límite de 50 elementos, por lo tanto la función debe validar la cantidad de valores de compra ingresados, además de verificar previamente si tiene espacio para poder ingresar más valores.  La función debe devolver la cantidad de elementos que posee una vez finalizada la carga de los precios de compra de la PlayStation.  **Modularizar en funciones lo que considere apropiado.** | Desarrollo parcialmente correcto.  Falta contemplar el escenario en que la Pila ya tenga datos cargados previamente. Además, v<500 es incorrecto, la Pila como máximo puede tener 50 elementos. |
| --- | --- |
| *2.* Hacer una función que muestre los precios almacenados en la pila. Dado que la moneda aprobada para realizar transacciones de compra y venta es el peso argentino (ARS), cada precio deberá mostrarse convertido a dicha moneda.  Para ello, antes de mostrar los valores, la función debe calcular el equivalente en pesos tomando como referencia la cotización:  USD 1 = ARS 1180.60 | Desarrollo correcto. |
| **3. Hacer una función que cargue en un arreglo los valores de compra del paquete de yerba Playadito de forma aleatoria.**  Para generar los precios aleatorios, se deberá utilizar la función rand(), asegurando que los valores estén dentro del rango de **$3000 a $5500 ARS**, inclusive.  La cantidad máxima de registros que pueden almacenarse en el arreglo es de **50 elementos**.  La función debe devolver la **cantidad de elementos cargados** en el arreglo. | Desarrollo correcto. |

| 4. Hacer una función que muestre los elementos del arreglo. | Desarrollo correcto. |
| --- | --- |

| 5. Hacer una función de tipo **float** que calcule el promedio los valores del arreglo y lo retorne. Modularice. | Desarrollo correcto. |
| --- | --- |
| 6. Hacer una función que reciba el arreglo, la cantidad de elementos, el promedio y elimine todos aquellos valores mayores al promedio. Debe ajustar el tamaño del arreglo en consecuencia. | Desarrollo correcto. |
| 7. De los valores almacenados en la Pila, buscar y devolver el precio más bajo registrado. **Debe eliminar el dato de la Pila.** | Desarrollo parcialmente correcto, la función no contempla si la pila recibida esta vacía. |
| 8. Crear una función que reciba como parámetro una pila **(pasada por referencia)** y un arreglo vacío. Esta función debe recorrer la pila y, por cada elemento, llamar a la función del punto anterior.  Si el valor devuelto por dicha función es par, debe almacenarse en el arreglo.  La función debe devolver la cantidad de elementos insertados en el arreglo, ya sea por valor de retorno o mediante un parámetro por referencia. **Importante: al finalizar la ejecución de la función, la pila debe conservar únicamente los datos impares.** | Desarrollo correcto. |