

Completar sus datos con letra de imprenta y bien prolijo.

Nro. de DNI \_\_\_\_\_ Nombre y Apellido: \_\_\_\_\_

Cantidad de hojas entregadas (sin contar el enunciado): \_\_\_\_\_

- Entregar cada problema en hojas separadas.
- Poner nombre, DNI y número de problema a toda hoja entregada.
- La solución a cada problema deberá respetar la misma salida que se muestra en los ejemplos.

**Problema 1 - Evento Estudiantil:** La inscripción a un evento estudiantil del DCIC se realizará la primera semana de Junio, de lunes a jueves. Para organizar la inscripción cada persona deberá inscribirse un día determinado según su carrera y el último dígito de su DNI. Tanto para las personas anotadas en la Ingeniería en Computación (C) como en Licenciatura en Ciencias de la Computación (L) se deberán inscribir al evento el lunes si su DNI termina en 0, 1, 2, 3 o 4, y lo deberán hacer el martes si su DNI termina en 5, 6, 7, 8, o 9. Las personas anotadas en la Ingeniería en Sistemas de Software (S) lo harán el miércoles si su DNI termina en 0, 1, 2, 3 o 4, o el jueves si su DNI termina en 5, 6, 7, 8, o 9. Suponemos que cada persona solamente estará anotada en una carrera.

- a) Escriba todos los casos de prueba significativos que considere necesarios.
- b) Escribir un programa en Pascal que solicite el DNI de la persona, y la letra que identifica a la carrera que está anotado (C, L o S) validando que los datos ingresados sean correctos y luego muestre en pantalla el día que debe inscribirse. Un DNI es correcto si es un número mayor a cero, la carrera es correcta si es una letra 'C', 'L' ó 'S' mayúscula o minúscula.

El programa en Pascal debe respetar este formato de ingreso y salida de datos:

```
Carrera que está anotado (L) Licenciatura (C) Ing. en Computación (S) Ing. en Sistemas: A
Error ingreso de datos
Carrera que está anotado (L) Licenciatura (C) Ing. en Computación (S) Ing. en Sistemas: a
Error ingreso de datos
Carrera que está anotado (L) Licenciatura (C) Ing. en Computación (S) Ing. en Sistemas: c
Ingrese su DNI: 54222999
Su DNI termina en el dígito 9
Debe concurrir el día: martes
```

**Sigue atrás** – Sigue al dorso – Continúa en la página siguiente – **Sigue en la hoja 2**  
Turn the page – Snu siden - Umblättern - 次のページを見る- Перегорнути сторінку - 翻页

---

**Problema 2 - Montaña Rusa:** En la montaña rusa "Tycoon" se quiere saber cuánto se debería haber recaudado luego de que hayan ingresado todas las personas al tren y se hayan sentado. El tren tiene una capacidad que depende de la cantidad de carros que tenga, y cada carro tiene varias filas. Igualmente en cada fila hay un único asiento y todos los asientos son individuales. Por lo tanto se solicita al usuario que ingrese la cantidad total de asientos disponibles para cada vuelta. Para fomentar la venta de entradas se cobra más barato a los menores (siempre respetando las normas de seguridad que permitan su ingreso), por lo cual el precio para los menores es de \$800 mientras que para los adultos es de \$2500. También se desea saber cuántos menores y adultos ingresaron y cuántos asientos vacíos hubo.

- Escriba todos los casos de prueba significativos que considere necesarios.
- Realice un programa en PASCAL que solicite la cantidad total de asientos disponibles para la vuelta, y luego la secuencia de personas tal como se ubicaron en el tren. En el caso de que la persona sea menor se la identifica con una 'm', si es adulto se la identifica con una 'a' y si el asiento quedó libre se identifica con una 'X'.

Solamente puede utilizar FOR como estructura repetitiva (no puede usar REPEAT ni WHILE), por ello no es necesario que realice ninguna validación de los datos de entrada (puede suponer en este problema que serán bien ingresados).

```
La cantidad de asientos disponibles (entre 1 y 25): 12
Ingrese la secuencia de ingreso: mmaXmaXXammX
Ingresaron 3 adultos y 5 menores
Se recaudó $11500
Quedaron 4 asientos vacíos
```

---

**Problema 3 - Secuencia Estable:** Se tiene una secuencia de números reales que termina en 0, y se dice que esta secuencia es estable si la diferencia absoluta entre el menor y el mayor elemento es menor o igual a una constante de tolerancia predefinida. Si se define un intervalo en la secuencia, se dirá que es estable si la diferencia absoluta entre el menor y el mayor elemento dentro de ese intervalo es menor o igual a la constante de tolerancia. Por ejemplo, si se tiene una tolerancia de 2.5 y la secuencia 4.5 2.3 3.2 4.1 0, se considera estable porque la diferencia absoluta entre el mayor (4.5) y el menor (2.3) elemento es menor a la tolerancia definida.

Por otro lado, si asumimos que el primer elemento de la secuencia está en la posición 1, el siguiente en la posición 2, y así sucesivamente, podemos decir que la secuencia es estable en el intervalo X e Y, si la diferencia absoluta entre el menor y el mayor elemento que se encuentren en el intervalo es menor o igual a la constante de tolerancia definida. Por ejemplo, para una constante tolerancia=1.2, el intervalo definido por las posiciones 3 y 6, si la secuencia es 1.4 2.8 3.8 3.4 3.6 4.1 4.5 -3.2 0 diremos que es estable porque  $4.1 - 3.4 = 0.7$  que es menor a 1.2.

- Escriba todos los casos de prueba significativos que considere necesarios.
- Realizar un programa en PASCAL que pida al usuario las posiciones del intervalo a definir y una secuencia de números reales terminada en 0, y determine **si la secuencia es estable dentro del intervalo definido**. El programa debe validar que las posiciones ingresadas sean números positivos y que el primer número sea menor o igual al segundo número. Se asume que las posiciones del intervalo siempre existen en la secuencia. El programa debe definir una constante para la tolerancia y seguir el siguiente formato de salida por pantalla:

```
Ingrese las posiciones del intervalo: 3 6
Ingrese la secuencia (terminada en 0): 1.4 2.8 3.8 3.4 3.6 4.1 4.5 -3.2 0

Considerando una tolerancia de 1.2 en el intervalo entre las posiciones 3 y 6
El mayor elemento es 4.1
El menor elemento es 3.4
La diferencia absoluta es 0.7
La secuencia es estable en el intervalo [3,6]
```