

Nombre y Apellidos: \_\_\_\_\_

Grupo: \_\_\_\_\_

DNI: \_\_\_\_\_

Firma: \_\_\_\_\_

### Examen CONVOCATORIA EXTRAORDINARIA – JUNIO 23

#### Instrucciones para el examen:

- Duración: **180 minutos**
- Se deben programar los enunciados descritos más abajo y entregar según las indicaciones que se den durante el examen.
- Rellena tus datos personales antes de comenzar a realizar el examen.
- No se permite el uso de libros o apuntes, ni tener teléfonos móviles u otros dispositivos electrónicos encendidos.
- Incumplir cualquiera de estas normas puede ser motivo de expulsión del examen.

## A. (5 puntos/10)

La empresa de delivery WeAreTheWorld ha decidido donar los céntimos de cada una de las 1000 primeras transacciones que ha realizado en 3 días consecutivos. Se trata de realizar un programa que calcule la cantidad donada.

Para simplificar la magnitud del problema, en nuestro caso particular supondremos que la donación se realizará de las 5 primeras transacciones realizadas en 3 días. Para ello contaremos con una matriz de 3 filas x 5 columnas en las que se almacenarán los importes de cada transacción. Si el importe no contiene decimales, no se donará nada. Si por el contrario el importe contiene decimales, se deberá incluir el importe donado (parte decimal) en un array/vector que será utilizado para calcular la cantidad total.

Se pide:

Escribir un programa completo siguiendo los siguientes pasos:

### 1. (2 puntos)

Leer desde teclado los valores de las transacciones de los 3 días y los almacene en la matriz **float transacciones[NUM\_DIAS][NUM\_TRANSACC]**, pidiendo al usuario dato a dato. Para ello, se definirán las constantes NUM\_DIAS y NUM\_TRANSACC con los valores que se han indicado más arriba en el enunciado.

Los datos leídos para rellenar la matriz serán los valores de las transacciones realizadas en tres días consecutivos. Si alguno de los días no se ha llegado a realizar 5 transacciones, entonces los valores restantes tendrán valor 0.

Una vez leída la matriz completa imprimir el número de transacciones realizadas así como los valores de la matriz completa utilizando dos decimales. Ejemplo de matriz de transacciones:

29.4	32	12.5	0	0
14.6	11	12.2	85.6	0
22	12.3	60	31.1	25

**FAST TRACK:** Pierdes los dos puntos de este apartado pero no te quedas atascado si defines e inicializas la matriz transacciones de la manera siguiente y pasas al punto 2:

```
float transacciones[NUM_DIAS][NUM_TRANSACC]= {  
    {29.4, 32, 12.5, 0, 0},  
    {14.6, 11, 12.2, 85.6, 0},  
    {22, 12.3, 60, 31.1, 25}  
};
```

### 2. (0,5 puntos)

Declarar un array **float donaciones[XXXXX]**, donde XXXXX debe ser el valor que permita guardar la parte decimal de las transacciones si todas las transacciones realizadas tuvieran parte decimal.

### 3. (2 puntos)

Recorrer la matriz **transacciones** mediante la función:

```
void buscaTrConDecimales (float transacc[][NUM_TRANSACC], float donaciones[],  
int tam, int *total_tr_decimales);
```

y rellenar el array donaciones con la parte decimal de todas las transacciones realizadas cuya parte decimal es distinta de 0. Devolver además el número de transacciones que tienen parte decimal distinta de 0.

En el momento que el valor de una transacción es 0, se entiende que no existen más transacciones en esa fila (en ese día).

Imprimir desde el programa principal el array donaciones y el número de transacciones que tienen parte decimal distinta de 0.

4. **(0,5 puntos)**

Calcular el total de las donaciones invocando la siguiente función:

```
void calcularDonacion(float donaciones[],int tam, float *total_donado);
```

Parámetros: array de las donaciones, total de transacciones realizadas y total donado.

La función debe recorrer el array de las donaciones y devolverá el importe total de la donación (total\_donado).

Imprimir desde el programa principal el valor de la cantidad donada.

Ejemplo: dada la matriz del punto 2, el número total de transacciones realizadas es 12 y el número de transacciones con decimales es: 7. El total donado es: 2.7€.

**Ejemplo Resultado de la ejecución del programa después de leer los datos de las transacciones:**

```

$ ./1.exe
Introduzca los valores de las transacciones del día 1:

Transacción número 1: 29.4

Transacción número 2: aa
El valor introducido no es correcto. Inténtelo de nuevo.
32

Transacción número 3: 12.5

Transacción número 4: 0

Transacción número 5: 0
Introduzca los valores de las transacciones del día 2:

Transacción número 1: 14.6

Transacción número 2: jj
El valor introducido no es correcto. Inténtelo de nuevo.
11

Transacción número 3: 12.2

Transacción número 4: 85.6

Transacción número 5: 0
Introduzca los valores de las transacciones del día 3:

Transacción número 1: 22

Transacción número 2: 12.3

Transacción número 3: 60

Transacción número 4: 31.1

Transacción número 5: 25

```

```

====> Valores de las transacciones:

      29.40   32.00   12.50   0.00   0.00
      14.60   11.00   12.20   85.60   0.00
      22.00   12.30   60.00   31.10   25.00

El número total de transacciones realizadas es: 12
====> El array de donaciones es:
      0.40   0.50   0.60   0.20   0.60   0.30   0.10

====> Numero de transacciones con decimales es: 7

====> El total donado es: 2.70

```

## B. (5 puntos/10)

Hacer un programa que genere la Password asociada al login de un usuario cuando éste se registra por primera vez en una aplicación.

1. (2 puntos) El programa pedirá el login al usuario que debe tener un formato determinado, a saber: debe tener un total de entre 8 y 15 caracteres (combinación de letras, números y guiones) teniendo en cuenta que debe contener al menos una mayúscula, al menos una minúscula, al menos 3 números y al menos un carácter especial ('-').

Mientras el login no cumpla las características de formato indicadas más arriba, el programa seguirá pidiendo una nueva cadena al usuario hasta que dicha cadena sea correcta.

**FAST TRACK:** Pierdes los dos puntos de este apartado pero no te quedas atascado si defines e inicializas la variable login de la manera siguiente y pasas al punto 2:

```
char myArray[] = "MyLogin12-9";
```

2. (0.5 puntos) Cuando el login sea correcto, el programa lo imprimirá por pantalla, indicando su longitud
3. (2.5 puntos) A continuación generará la password partiendo del login que se transformará del siguiente modo:
  - Cada número del login será transformado en una letra minúscula teniendo en cuenta que el 9 se transformará en 'z', el 8 en 'y', el 7 en 'x' y así sucesivamente. Es decir, la asignación del 0 al 9, empieza transformando el 9 en la última letra del abecedario en minúscula y continuando en orden inverso hasta llegar al 0, que se transformaría en la letra 'q'.
  - Cada letra minúscula del login se transforma en la misma letra pero en MAY.
  - Cada letra mayúscula del login se transforma en la misma letra pero en minúsculas.
  - El guión se transformará en '\*'.

El programa imprimirá entonces la Password generada para que el usuario pueda registrarse por primera vez en la aplicación con su login y password.

Se deben utilizar los siguientes prototipos de funciones:

```
// Funcion que valida el formato del login (debe tener al menos 3 números, 1 letra MAY, 1 letra min y un guión y como resultado devolverá un 1 si el formato es válido y 0 en caso contrario,  
int validaLogin(char login[NUM_MAX_CARACTERES]);
```

```
// Genera la password según el algoritmo de transformación indicado en el enunciado  
void generaPassword(char login[NUM_MAX_CARACTERES], char password[NUM_MAX_CARACTERES]);
```

### Ejemplo Resultado de la ejecución del programa después de leer los datos de las transacciones:

```
$ ./2.exe
Introduce el login de usuario:
Debe tener entre 8 y 15 caracteres, y tener mínimo tres números, una MAY, una MIN, y un '-'
Lola
===>ERROR: El login debe tener mínimo 8 caracteres. Intentalo de nuevo
LolaLopez

===>El login introducido es: LolaLopez
===> Su longitud es: 9

===>El login NO ES valido, intentalo de nuevo
Lola4321

===>El login introducido es: Lola4321
===> Su longitud es: 8

===>El login NO ES valido, intentalo de nuevo
Lola9876-

===>El login introducido es: Lola9876-
===> Su longitud es: 9

===>El login ES válido.
===>La password de acceso es: l0LAzyxw*
```