FECHA: 29/01/2020 (Ordinario)

A. (2, 5 puntos de la nota total)

Desarrollar una función para convertir coordenadas cartesianas en polares. En el programa hay dos vectores inicializados con las coordenadas `x,y` de 5 puntos del plano. Se pide convertirlas a polares usando la función `CartAPolares()` e imprimir el resultado con un bucle fuera de la función. Para cada par x,y de coordenadas cartesianas, las polares equivalente son radio y ángulo, según las siguientes fórmulas.

```
radio = sqrt(x^2+y^2) // El radio es la distancia al origen
angulo = arctan(y/x) // El ángulo en radianes es el arcotangente del
                     //cociente y/x
```

Puedes usar la librería `math` de C que proporciona las funciones `sqrt()` y `atan()` el esqueleto del programa es:

```
#include <stdio.h>
#include <math.h>
#define NUMPUNTOS 5
// Prototipo de la función que tienes que desarrollar, con este nombre y estos
parámetros
void CartAPolares (float x[NUMPUNTOS], float y[NUMPUNTOS], float *radio, float
*angulo);
void main()
{
   float coord_x[NUMPUNTOS] = {1,1,-1,1.2,8};
   float coord_y[NUMPUNTOS] = {1,9,0,-1,2.7};
   …
}
```

Ejemplo de salida

```
El punto x=1,y =1 en coordenadas cartesianas equivale en polares a radio =
1.414 y ángulo = 0.785
…
```

B. **(2,5 puntos de la nota total)**

Para realizar cualquier trámite con las administraciones públicas es necesario identificarse de manera unívoca. Para ello, se utiliza el documento nacional de identidad (DNI). Este identificador está formado por 8 dígitos (del 0 al 9) y por una letra mayúscula (de la A a la Z), unidos por el símbolo guion (-). Un ejemplo de DNI es: 12345678-A
Para una nueva aplicación web se va a tomar como identificador de usuario el documento nacional de identidad precedido de los caracteres DNI/, con la particularidad de que los 8 dígitos estarán en orden inverso. El identificador de usuario del ejemplo anterior es: DNI/87654321-A
Cread un programa que a partir de un DNI genere el nuevo identificador. Para ello

A. Introducir por teclado una cadena de caracteres con el número de caracteres del formato de un documento nacional de identidad. Mientras la cadena no tenga ese número de caracteres se debe seguir pidiendo al usuario.
B. Cread una función que compruebe que los caracteres introducidos se corresponden con el formato de un documento nacional de identidad.
C. El prototipo de la función que genera el nuevo identificador es el siguiente prototipo (recordad que se debe invocar desde el programa principal)

```
void NuevoIdentificador(char* p_Cadena, char* p_nueva_Cadena);
```

D. Escribir por pantalla el identificador de la nueva aplicación.

C. **(2,5 puntos de la nota total)**

Cálculo de métricas estadísticas en una matriz: crear un programa que reciba dos enteros N y M, y use una matriz de tipo `float` de N filas y M columnas rellenando dicha matriz con números aleatorios entre 0 y 1. El programa debe también imprimir la matriz y a continuación debe calcular e imprimir los valores de mínimo, máximo, suma, promedio y desviación estándar para cada una de las columnas. Todos los `printfs` de números deben estar formateados utilizando 4 decimales. Utilizar `pow(valor,2)` si necesitas calcular el cuadrado y `sqrt(valor)` si necesitas la raíz cuadrada. Para generar los números aleatorios flotante entre 0 y 1 simplemente utilizar:

```
(float) rand() / (float) RAND_MAX
```

```
Ejemplo de la salida para llamada: calcula_estadisticas(5,3);
0.0013  0.5636  0.1933
0.8087  0.5850  0.4799
0.3503  0.8960  0.8228
0.7466  0.1741  0.8589
0.7105  0.5135  0.3040
Coluna 0 - min: 0.0013, max: 0.8087, suma: 2.6174, promedio: 0.5235, desviacion estandard: 0.3062
Coluna 1 - min: 0.1741, max: 0.8960, suma: 2.7322, promedio: 0.5464, desviacion estandard: 0.2296
Coluna 2 - min: 0.1933, max: 0.8589, suma: 2.6590, promedio: 0.5318, desviacion estandard: 0.2687
```

Acordaros que la fórmula de la desviación estándar es:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{N}}$$

La desviación estándar de la columna 0 del ejemplo anterior sería:

- promedio = 0.5235
- RAIZ_CUADRADA((0,0013-0,52348)^2 + (0,8087-0,52348)^2 + (0,3503-0,52348)^2 + (0,7466-0,52348)^2 + (0,7105-0,52348)^2)) / 5) = 0.3062

FECHA: 03/02/2021 (Ordinario)

A. (3 puntos de la nota total)

En la compañía CactusSystem todos sus empleados tienen un correo electrónico que se ha generado con el siguiente criterio: `nombre.apellido@cactussystem.com`. Para una nueva aplicación de mensajería, el equipo de sistemas necesita un identificador de usuario para cada empleado. Para ello va a utilizar los correos electrónicos, tomando solo la parte de `nombre.apellido`.

Se pide:

- Desarrollar una función `GenerarUsuario` que a partir del correo de un empleado genere su usuario para la nueva aplicación.
- Desarrollar una función `TamanioUsurio` que devuelva el número de símbolos del usuario.
- Imprimir el correo electrónico, el usuario y el tamaño del usuario.

Para ello programa pedirá por teclado uno por los correos electrónicos e imprimirá el correo

El esqueleto del programa es:

```
#include <stdio.h>
#define NUM_EMPLEADOS 5    /* Numero de empleados */
#define NUM_CHAR_CORREO 50 /* Numero maximo de caracteres del correo */
#define NUM_CHAR_USER 30   /* Numero maximo de caracteres del usuario */

// Prototipo de las funciones que tienes que desarrollar, con este nombre y
estos parámetros
void GenerarUsuario (char correo[NUM_CHAR_CORREO], char user[NUM_CHAR_USER]);
int TamanioUser (char *usuario);
void main(){
      …
}
```

Ejemplo de salida

```
1. Introduce el correo electronico:    elsa.ruiz@castussystem.com
1. El correo es: elsa.ruiz@castussystem.com, el usurios es elsa.ruiz y el
tamanio es 9
2. Introduce el correo electronico:    juan.silvano@cactussystem.com
2. El correo es: juan.silvano@cactussystem.com, el usurios es juan.silvano y
el tamanio es 12
3. Introduce el correo electronico:    olvido.bineka@cactussystem.com
3. El correo es: olvido.bineka@cactussystem.com, el usurios es olvido.bineka y
el tamanio es 13
```

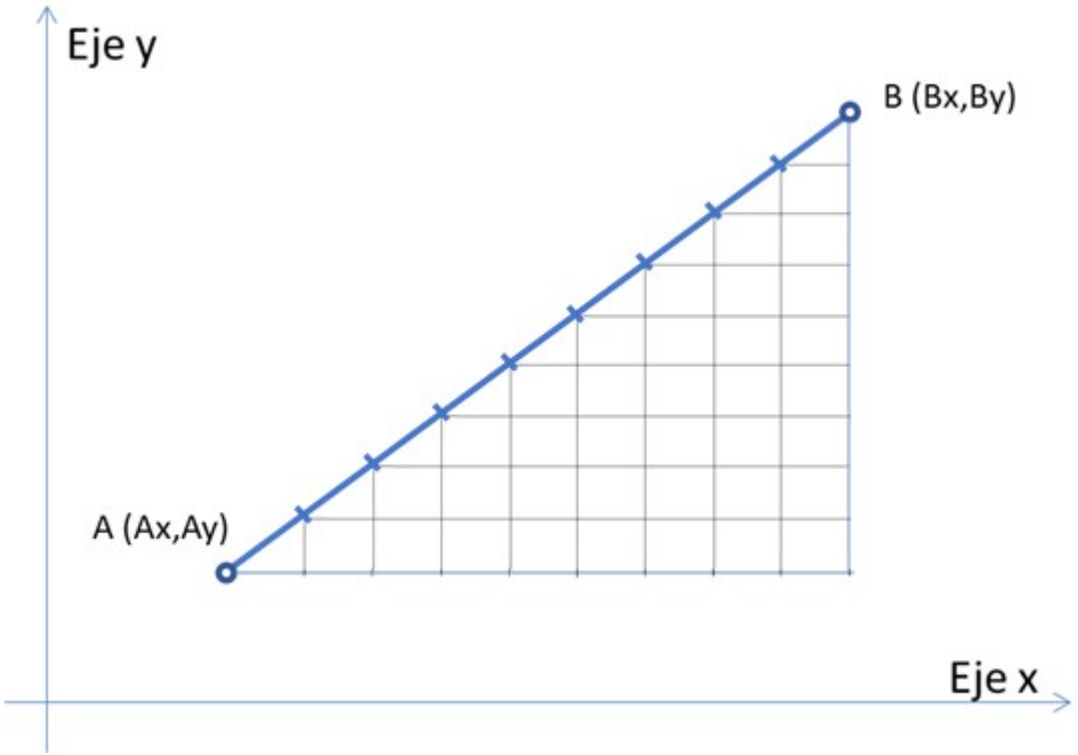
B. **(4 puntos de la nota total)**

Dados dos puntos (A y B) en un espacio 2D, identificados por sus coordenadas <X,Y>, (Ax,Ay) (Bx, By), se pide implementar un programa que permita calcular las siguientes operaciones:

- Función1: Discretización de las coordenadas del segmento del vector que les une, usando una resolución de 10 puntos. (IE, calcular los 10 puntos intermedios separados por intervalos regulares, incluyendo el inicio y el final del vector).

Esta función recibirá por parámetros las coordenadas X.Y de los dos puntos, y los arrays donde se almacenarán las coordenadas de los puntos intermedios calculados

- Función 2: Cálculo de la distancia que hay entre los dos puntos introducidos por el usuario. Dadas las coordenadas X,Y de dos puntos en un espacio 2D, devolverá el valor de la distancia entre esos dos puntos.
- Pedir consecutivamente al usuario que le proporcione las coordenadas de los dos puntos. Por simplicidad, se a considerar que las coordenadas son números enteros positivos.
- Para concluir el programa deberá mostrar por pantalla los puntos intermedios y la distancia entre los dos segmentos.



El esqueleto del programa es:

```
#include <stdio.h>
#include <math.h>
#define NUM_PUNTOS_INTERMDDIOS 10

// Prototipo de las funciones que tienes que desarrollar, con este nombre y
estos parámetros
void CalculoPtsIntermedios (float Ax, float Ay, float Bx, float By, float
*listaInterX, float *listaInterY);
void mostrarDatos(float* listaInterX,float* listaInterY);
float distanciaXY (float Ax, float Ay, float Bx, float By);

void main(){
```

```
        …
}
```

Ejemplo de salida

```
Introduzca coordenadas <x,y> del primer punto
2.5 3.6
Introduzca coordenadas <x,y> del segundo punto
25.0 18.3
Punto 0 <2.500000 , 3.600000>
Punto 1 <4.750000 , 5.070000>
Punto 2 <7.000000 , 6.540000>
Punto 3 <9.250000 , 8.010000>
Punto 4 <11.500000 , 9.480000>
Punto 5 <13.750000 , 10.949999>
Punto 6 <16.000000 , 12.420000>
Punto 7 <18.250000 , 13.889999>
Punto 8 <20.500000 , 15.359999>
Punto 9 <25.000000 , 18.299999>
La distancia entre los puntos es 26.876383 unidades.
```

APENDICE

Para el cálculo de raices cuadradas, se dispone de la function sqrt, que está implementada en la librería matemática de gcc. Para poder usarla:

- Paso 1: Incluir “math.h”
- Paso 2: usar la función sqrt, recibe por pará metros un número en format float/double y devuelve el cálculo de su raíz cuadrada.
- Paso 3: Al compilar añadir el flag “-lm”, para que no haya errores de linkado

Ejemplo:

```
#include <math.>
#include  <stdio.h>

void main()
{
   float res=sqrt(9);
}
```

FECHA: 19/02/2021 (Versión COVID)

A. (3 puntos de la nota total)

En la compañía IntegratedCircuits se ha desarrollado un nuevo componente que realiza cálculos trigonométricos. Para comprobar el funcionamiento se ha realizado un programa en c que calcula el seno y el coseno de un ángulo a partir de su serie de Taylor.

Se pide desarrollar el programa que:

- Pida al usuario el ángulo en grados y la precisión con la que se va a aproximar la serie de Taylor
- Desarrollar una función que convierta el ángulo de grados a radianes
- Desarrollar la función iterativa `CalcularSeno` que a partir del valor del ángulo en radianes y de la precisión devolverá los valores aproximados por la serie de Taylor del seno.
- Desarrollar una función `factorial` que sea utilizada por la función `CalcularSeno`. Se valorará el uso de recursividad en esta función.
- Una vez realizado los cálculos el programa debe presentar los resultados. Todos los cálculos deben realizarse dentro de funciones, pero los resultados se muestran desde el programa principal.

El desarrollo de Taylor del seno es:

$$x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ldots+(-1)^n\frac{x^{2n+1}}{(2n+1)!}+\ldots,\forall x\in R$$

donde x es el ángulo en radianes y n la precisión (número de términos de la sucesión que se toman)

El esqueleto del programa es:

```
#include <stdio.h>
#define PI 3.1416

// Prototipo de las funciones que tienes que desarrollar, con este nombre y
estos parámetros
float GradosToRadianes (int grados);
int factorial (int entero);
float CalcularSeno (float anguloRad,int precision);

void main(){ …… }
```

ALUMNO: _____ GRUPO:_____

B. (4 puntos de la nota total)

Se pide implementar un programa que pida una frase al usuario por consola, escrita en una sola línea, y que calcule una serie de estadísticas sobre la misma, y algunas modificaciones. Las condiciones de la frase de entrada son las siguientes:

- Máximo tamaño de 200 caracteres.
- Solo se podrán utilizar letras minúsculas y espacios

Se pide implementar un programa que calcule las siguientes estadísticas:

- Pedir al usuario que introduzca la frase haciendo todas las comprobaciones que estime oportunas.
- Cálculo de número total de caracteres (incluyendo los espacios en blanco) que tiene la frase, número de vocales y consonantes que hay en la frase. Mostrarlo al usuario
- Cálculo de número de palabras que hay en la frase.

Prototipos de funciones a implementar:

```
//funciones que devuelven true o false en caso de que se cumplan sus condiciones (caracter es vocal, consonante, no es ni vocal ni consonante
int esVocal(char caracter);
int esConsonante(char caracter);
int niVocalNiConsonante(char caracter);
int esCaracterValido (char caracter);

//Función que, dada una cadena de caracteres (variable “frase”), calcula el número de palabras encontradas en la misma. La varible numeroPalabras es una variable de salida (lectura/escritura) que se usará para guardar el número de palabras
void calculaPalabras(char *frase, int *palabras);
```

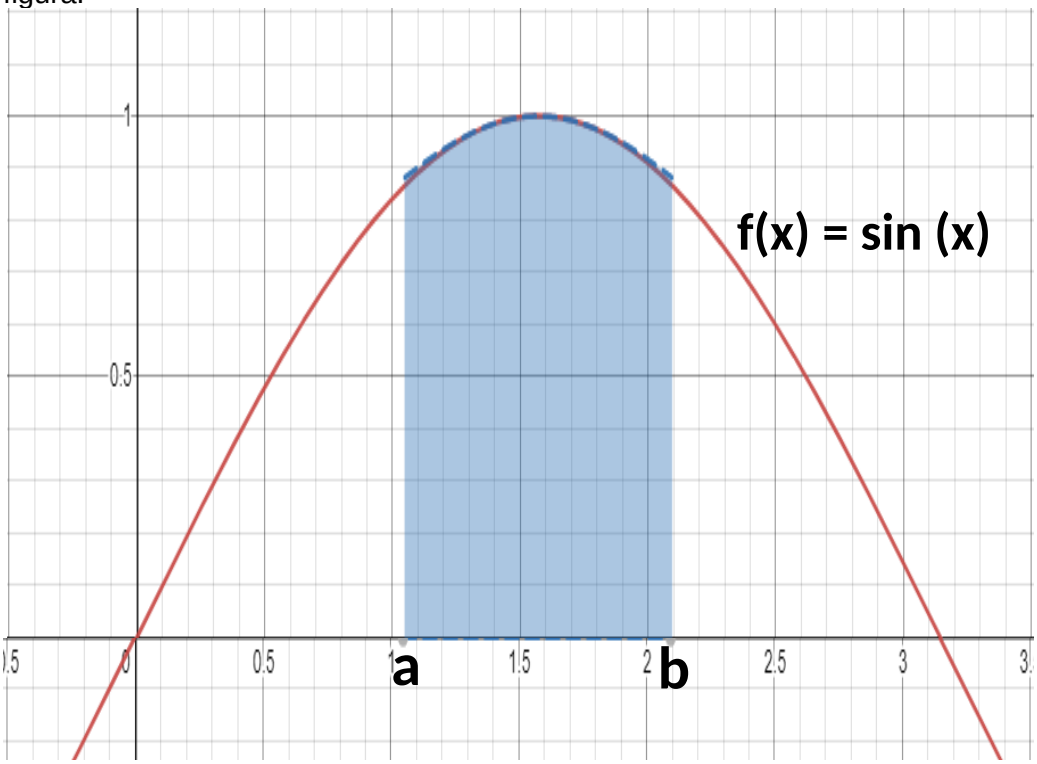
Ejemplo de uso:

```
Introduce una frase de menos de 200 caracteres (solo minusculas):
En un lugar de la mancha
Texto introducido no valido
Introduce una frase de menos de 200 caracteres (solo minusculas):
en un lugar de la mancha
Tiene 24 caracteres, 8 vocales y 11 consonantes
El numero de palabras es 6
```

Fecha: 30/06/2021 (extraordinaria)

A. Problema 1 (4 puntos de la nota total) [60 minutos]

Se desea calcular el área comprendida entre la curva que define la función seno, dos abscisas y el eje. X.
Para ello se dividirá el segmento a y b en N partes, y el área se calculará como la suma de las áreas de los N rectángulos delimitados por la curva dada y el eje x, tal y como se muestra en la figura.



Como condiciones, el intervalo a, b deberá estar incluido en los cuadrantes primero y segundo es decir que a debe ser mayor que o y b debe ser menor que PI. Además, b debe ser mayor o igual que a.

Se pide:

- Desarrollar una función `AreaSeno` que, a partir de los valores de a, b y el número de rectángulos determine el área buscada. Esta función devolverá el resultado del área (suma de las áreas de los rectángulos)
- Desarrollar una función `ComprobarValores` que a partir de los valores de a y b compruebe que estos cumplen las condiciones del enunciado el número de símbolos del usuario. Esta función devolverá si hay o no error en la introducción de los valores a y b.
- En caso de que los valores no sean correctos el programa debe volver a pedirlos.
- Imprimir el área.

El esqueleto del programa es:

```
#include <stdio.h>
#include <math.h> //uso de la función sin
```

```
#define PI 3.1416

// Prototipo de las funciones que tienes que desarrollar, con este nombre y
estos parámetros
int ComprobarValores (float valorA, float valorB);
float AreaSeno (float valorA, float valorB, int N);


void main(){

}
```

Ejemplo de salida

```
Introduzca el extremo inferior del intervalo:
 0
Introduzca el extremo superior del intervalo:
 1.5708
Introduzca el numero de rectangulos:
 1000
Los valores de a y b son correctos.
El area de la función seno en el intervalo (0.000000, 1.570800) es 1.000789
```

ALUMNO: _____ GRUPO:_____

B. Problema 2 (3 puntos de la nota total) [60 minutos]

Se pide desarrollar un programa que realice la siguiente operación:

- Pedirá al usuario que escriba una frase, que tendrá como máximo 15 letras.
- Una vez introducida, mostrará la siguiente información:
  - El número de caracteres que contiene.
  - Misma frase introducida por el usuario, pero escrita del revés.

El alumno deberá crear dos funciones con las siguientes características:

- Función `cuentaLetras`. Recibe por parámetros una cadena de caracteres inicializada con la frase introducida por el usuario, y un puntero a una variable `integer` llamada `numLetras`. La función contará el número de letras que tiene la cadena "`fraseIn`", y almacenará en "`numLetras`" el resultado. El prototipo/cabecera es el siguiente:

```
void cuentaLetras(char* fraseIn, int* numLetras);
```

- Función "`delReves`" que recibe por parámetros dos cadenas. La primera cadena "`fraseIn`" contiene la frase introducida por el usuario. La segunda cadena es un array de caracteres inicializado previamente, que será rellenado por la función con la frase original escrita del revés. El parámetro de entrada "`numLetras`" contiene el tamaño de la frase original. El prototipo/cabecera es el siguiente

```
void delReves(char* fraseIn, char* fraseOut, int numLetras);
```

Para desarrollar el programa, se ofrece el siguiente esqueleto que el alumno deberá rellenar. Se permite usar arrays de tamaño estático para la lectura/tratamiento de datos. Se valorará positivamente evitar el uso de variables globales y el uso correcto de las cabeceras suministradas. El programa debe funcionar sin errores de memoria.

```
#include <stdio.h>
#include <stdlib.h>

void cuentaLetras(char* fraseIn, int* numLetras);
void delRevés(char* fraseIn, char* fraseOut, int numLetras);
int main(int argc,char** argv){

// Rellenar por el alumno.
// - Declaración de variables
// - Pedir datos al usuario
// - LLamar a cuentaLetras
// - Mostrar numero de letras
// - Llamar a delReves
// - Mostrar frase del revés

return 0;
}

void cuentaLetras(char* fraseIn, int* numLetras){
// rellenar por el alumno
}
void delRevés(char* fraseIn, char* fraseOut, int numLetras){
//rellenar por el alumno
}
```

Ejemplo de salida

A continuación, se muestra un ejemplo de ejecución, donde el usuario ha introducido la frase “¡Hola mundo!”

- `Introduzca una frase, máximo 15 letras`
- `¡Hola mundo!`
- `La frase contiene 12 letras`
- `La frase del revés es: ¡odnum aloH¡`

FECHA: 18/01/2022 (Ordinaria)

A. (3,5 puntos de la nota total)

Escribe un programa que lea dentro de un bucle números enteros positivos del 1 al 9 por teclado hasta que se introduzca el valor cero. El programa debe dar como resultado el número introducido mayor número de veces de forma consecutiva indicando cuál es dicho número.

Adicionalmente, el programa tiene que mostrar el número total de dígitos introducidos (todos menos el valor cero) y el número total de cada uno de los dígitos introducidos (independientemente si son consecutivos o no).

Ejemplo de ejecución:

Introduce número: 3
Introduce número: 3
Introduce número: 3
Introduce número: 5
Introduce número: R
El dato introducido no es correcto
Introduce número: 1
Introduce número: 1
Introduce número: 9
Introduce número: 7
Introduce número: 7
Introduce número: 7
Introduce número: 7
Introduce número: 7
Introduce número: 3
Introduce número: 7
Introduce número: 0

Resultado:

El número que se ha introducido mayor número de veces (número repetido de forma consecutiva) es el 7 y se ha escrito 5 veces.

El número total de dígitos introducidos es: 14

El número de veces que se ha introducido cada digito es:

Número 1: 2
Número 3: 4
Número 5: 1
Número 7: 6
Número 9: 1

ALUMNO: _____ GRUPO:_____

B. **(3,5 puntos de la nota total)**

Sabiendo que la letra que acompaña a los dígitos de un DNI se calcula mediante un algoritmo basado en dichos dígitos, se pide realizar un programa que verifique si existe o no concordancia entre los datos de un DNI, es decir entre los datos numéricos del DNI y la letra que los acompaña. El DNI debe ser solicitado al usuario y éste debe introducirlo por pantalla. Para ello:

1. **Solicitar los números del DNI (sin letra)**. Se debe comprobar que el valor introducido es positivo y tiene a lo sumo 8 dígitos. En caso de no cumplir estas condiciones se ha de indicar que el DNI introducido es incorrecto y se pedirá un nuevo valor. Para esta verificación se debe implementar una función que reciba como parámetro el número del DNI y, tras hacer las comprobaciones pertinentes, devuelva el valor 1 (VERDADERO) o el valor 0 (FALSO).

2. **Una vez introducido un DNI correcto, el programa solicitará una letra mayúscula** y el programa debe determinar si DNI y letra son concordantes. Para ello el alumno debe:

   a) Implementar una función que determine si la letra introducida es mayúscula, es decir, es decir, una función que reciba como parámetro una variable tipo char y devuelva un 1 (VERDADERO) si el carácter es mayúscula y 0 (FALSO) en caso contrario.

   b) Implementar una función que reciba los dígitos del DNI y devuelva la letra que le corresponde. Para calcular dicha letra hay que tener en cuenta que hay 26 letras mayúsculas y que si el resultado de "dígitos_DNI%26"es igual a cero la letra correspondiente es la 'A'. Para resultado igual a 1 la letra es la 'B' y así sucesivamente.

   c) Implementar otra función que reciba los dígitos del DNI, la letra del DNI introducida por el usuario y devuelva si dicha letra corresponde o no a los dígitos introducidos.

Al final, el programa debe mostrar si el DNI introducido es correcto (dígitos y letra concuerdan) o no (no concuerdan).

```
// Prototipo de las funciones que tienes que desarrollar, con este nombre y estos parámetros

// Devuelve 1 si la letra es MAY y 0 en caso contrario.
int esMayuscula(char car);

// Devuelve 1 si el DNI es correcto y 0 en caso contrario. EL DNI es positivo y a lo sumo tiene 8 dígitos.
int validarDni(int n);

// Devuelve 1 si la letra es concordante con el DNI recibido como parámetro. Devuelve 0 en caso contrario.
int verificarLetraNif(int dni, char letra);

// Cálcula la letra asociada a DNI recibido y la devuelve como resultado de la función.
char calcularLetraNif(int n);
```

```
#include <stdio.h>

int main( ) {
 …
 …
}
```

Ejemplo de ejecución:

Introduce los dígitos de tu DNI: 1051054
Introduce la letra asociada: E

Los dígitos y la letra introducidos SON concordantes. El NIF 1051054-E es correcto

Ejemplo de ejecución:

Introduce los dígitos de tu DNI: 50234389
Introduce la letra asociada: X

Los dígitos y la letra introducidos SON concordantes. El NIF 50234389-X es correcto

Ejemplo de ejecución:

Introduce los dígitos de tu DNI: 123456789
El dato es incorrecto. Máximo 8 dígitos.

Introduce los dígitos de tu DNI: 1051053
Introduce la letra asociada: T

Los dígitos y la letra introducidos NO SON concordantes. El NIF 1051053-T es incorrecto.

FECHA: 08/02/2022 (Ordinario versión COVID)

A. **(2,5 puntos de la nota total)**

Se quiere averiguar el número de la suerte o número "mágico" asociado a la fecha de nacimiento de una persona. Para calcularlo se suman todos los números de su fecha de nacimiento y a continuación se reducirán a un solo dígito.

Ejemplo:
Fecha de nacimiento: 05/02/1973 ➔ 5 + 2 + 1973 = 1980 ➔ 1 + 9 + 8 + 0 = 18 ➔ 9

Realizar un programa que:

a) Solicite al usuario el año, el mes y el día que componen su fecha de nacimiento y compruebe que la fecha introducida es una fecha válida.
   La fecha se considerará válida si: año >0; 1<=mes<=12; 1<=dia<=DIAS_MES. Se considerará que, con independencia del año, el número de días del mes de Febrero es 28.
   Los días que tiene cada mes son:
   Enero: 31, Febrero: 28, Marzo: 31, Abril: 30, Mayo: 31, Junio: 30, Julio: 31, Agosto:31, Septiembre: 30, Octubre: 31, Noviembre: 30, Diciembre: 31
b) Mostrar por pantalla la Fecha de Nacimiento leída
c) Calcular el número mágico asociado a esa fecha.
d) Mostrar el número mágico por pantalla

Ejemplo de ejecución:

Introduce tu Fecha de Nacimiento.

Año: 1998
Mes: 6
Dia: 5

Tu fecha de Nacimiento es: 05/06/1998
El número mágico asociado a tu fecha de nacimiento es: 2

Ejemplo de ejecución:

Introduce tu Fecha de Nacimiento.

Año: 1998
Mes: 14
El mes introducido no es correcto.

Mes: 6
Dia:5

Tu fecha de Nacimiento es: 05/06/1998
El número mágico asociado a tu fecha de nacimiento es: 2

Ejemplo de ejecución:

Introduce tu Fecha de Nacimiento.

Año: 1998
Mes: 6
Dia: 31
El día del mes introducido no es correcto.

Dia: 5

Tu fecha de Nacimiento es: 05/06/1998
El número mágico asociado a tu fecha de nacimiento es: 2

ALUMNO: _____ GRUPO:_____

B. **(2,5 puntos de la nota total)**

La sala de un minicine tiene 4 filas y cada una de ellas tiene 8 asientos. Por las condiciones sanitarias para evitar la propagación de un virus si un asiento se encuentra ocupado los contiguos en la misma fila deben estar reservados Desarrollad un programa que simule la venta de localidades para este minicine con esta restricción.
El programa debe

- Considerar inicialmente que todos los asientos están libres. Marcados con una 'O'.
- Mostrar el aspecto de la sala de cine, es decir, mostrar el estado de las butcas
- Pedir consecutivamente al usuario que le proporcione la localidad (fila y asiento) que desearía ocupar el cliente, hasta que le minicine esté lleno.
- Comprobar si. los valores de fila y asiento son números enteros válidos y comprobar si se puede reservar dicha butaca
- Después de cada petición debe mostrar por pantalla un esquema en el que se visualice el estado de ocupación del minicine marcando con una X las localidades ocupadas, con un * las localidades reservadas que no se puede ocupar por las condiciones de seguridad y marcadas con una O las localidades libres.
- Cuando el minicine se haya llenado, el programa debe indicarlo, señalando el número de localidades que se han ocupado y el número de localidades que han quedado reservadas por la restricción.

```
//inicializa las butacas como libres
void inicializaCine(char sala[FILAS][COLUMNAS], int filas, int columnas);

//muestra por terminal los datos de la sala
void mostrarCine(char sala[FILAS][COLUMNAS], int filas, int columnas);

//Pide los datos por pantalla (fila, columna), evaluando si son correctos
void leeAsiento(int *f, int *c);

//comprueba si un asiento dado (fila/columna) está libre y se puede reservar. Devuelve
1 si se puede reservar, 0 en otro caso
int asientoLibre(char sala[FILAS][COLUMNAS], int filas, int columnas);

//Comprueba si no se pueden reservar más butacas. Devuelve 1 si no se pueden
reservar, 0 si hay butacas disponibles
int cineCompleto(char sala[FILAS][COLUMNAS], int filas, int columnas);
```

FECHA: 21/06/2022 (Extraordinaria)

A. (X puntos de la nota total)

Se desean tener guardados en un array los números premiados en el “gordo” de Navidad de los últimos 12 años.

Para inicializar el array el usuario debe introducir los datos por teclado asegurando que los números son válidos. Dichos números serán válidos si son enteros positivos y no superan el valor 90.000.

Se quieren obtener una serie de estadísticas sobre los números premiados que han salido en estos últimos años, para ello se pide:

a) Calcular el total de números premiados de cada terminación, es decir cuántos números premiados han acabado en 0, cuantos han acabado en 1, en 2, etc. (hasta los acabados en 9).
b) La terminación más popular, es decir, la cifra en la que ha acabado el gordo mayor número de veces.
c) El número más alto que ha sido premiado en todos los sorteos.

Ejemplo de resultado de salida:

Supongamos que los números premiados en el gordo de Navidad de los últimos 12 años han sido:

24.215	11.348	55.965	75.642
14.212	25.625	23.267	4.025
9.671	89.973	35.700	6.005

Números premiados acabados en 0: 1
Números premiados acabados en 1: 1
Números premiados acabados en 2: 2
Números premiados acabados en 3: 1
Números premiados acabados en 4: 0
Números premiados acabados en 5: 5
Números premiados acabados en 6: 0
Números premiados acabados en 7: 1
Números premiados acabados en 8: 1
Números premiados acabados en 9: 0

La terminación más popular en estos años ha sido el “5”.

El número premiado más alto premiado en estos últimos años ha sido el: 89.973

B. (X puntos de la nota total)

Los empleados de la empresa JustBuy trabajan por objetivos de ventas que deben cumplir cada trimestre. Cada año seleccionan a los 5 mejores empleados y les premian por su trabajo. Este año las ventas (en €) de los 5 mejores empleados obtenidas en cada trimestre han sido las siguientes:

	Q1	Q2	Q3	Q4
E1:	700	234	250	960
E2:	895	800	100	432
E3:	790	634	850	560
E4:	600	854	550	460
E5:	590	987	750	660

Declarar una matriz de números enteros de dos dimensiones e inicializarla con dichos datos. Realizar un programa, en bucle infinito, con un menú que tenga las siguientes opciones:

1. Calcular las ventas acumuladas, suma de la de los 4 trimestres del año, de cada uno de los empleados y mostrar el resultado por pantalla.
2. Calcular el mínimo de todas las ventas realizadas e indicar a que empleado y trimestre corresponde.
3. Salir

Ejemplo de resultado de salida:

Ventas por trimestre de cada empleado:

Ventas anuales del Empleado1 = 2.144 €

Ventas anuales del Empleado2 = 2.227 €

Ventas anuales del Empleado3 = 2.834 €

Ventas anuales del Empleado4 = 2.464 €

Ventas anuales del Empleado1 = 2.987 €

Mínimo de las ventas realizado es 100€. Corresponden al empleado 2 en el Q3.