

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 `TamanoUsurio` 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 TamanoUser (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 usurio es elsa.ruiz y el
   tamaño es 9
2. Introduce el correo electronico:    juan.silvano@cactussystem.com
2. El correo es: juan.silvano@cactussystem.com, el usurio es juan.silvano y el
   tamaño es 12
3. Introduce el correo electronico:    olvido.bineka@cactussystem.com
3. El correo es: olvido.bineka@cactussystem.com, el usurio es olvido.bineka y
   el tamaño es 13
```

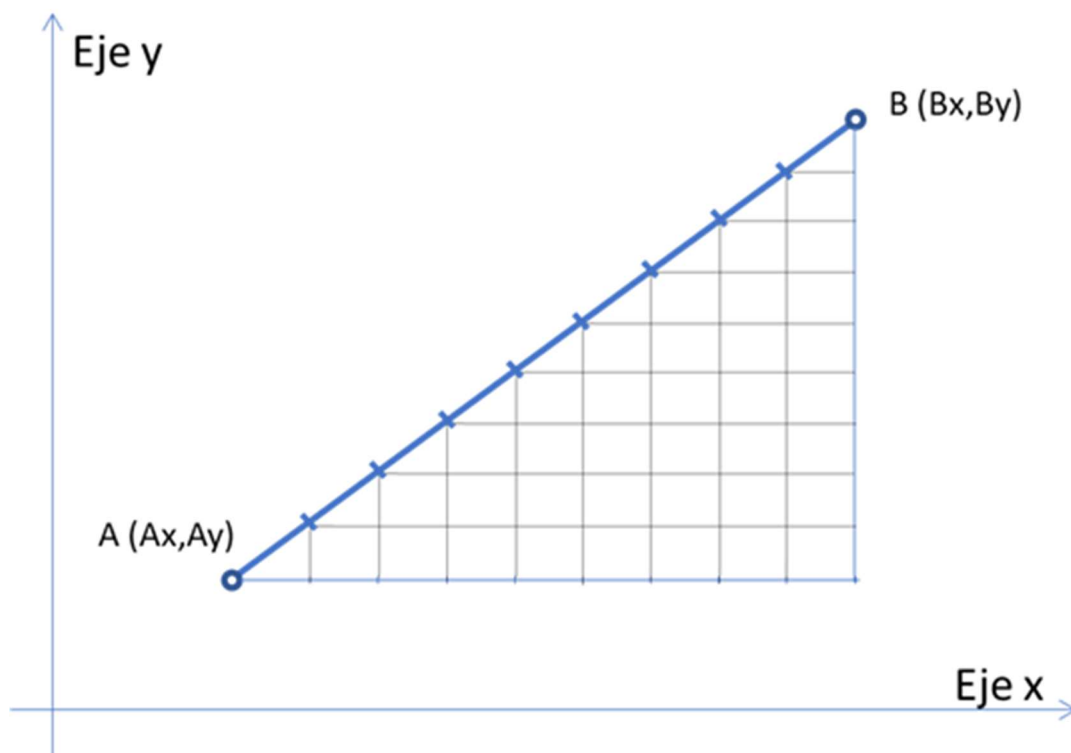
B. (4 puntos de la nota total)

Dados dos puntos (A y B) en un espacio 2D, identificados por sus coordenadas $\langle X, Y \rangle$, (A_x, A_y) (B_x, B_y) , 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_INTERMEDIOS 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 raíces 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 parámetros 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);  
}
```