Funciones.

Fun1. Hacer una función tipo void, que determine cual de tres valores pasados como parámetros es el mayor.

Fun2. Hacer una función tipo void, para dibujar en la pantalla un recuadro según 4 coordenadas pasadas como parámetros, Fila Inicial, Columna Inicial, Fila Final, Columna Final. Colorear el área del interior del cuadro, el color también se ha de pasar con un parámetro. La función ha de controlar que el rectángulo se pueda hacer o no, ver programa R11.

Fun3. Hacer una función tipo void, que muestre todos las números primeros que hay entre dos valores que se pasan como parámetros.

Fun4. Hacer una función Tipo int, que devuelva el factorial del número que se le pasa como parámetro.

Fun5. Hacer una función tipo int, que devuelva la potencia de una base elevada a un exponente. Base y exponente se pasan como parámetros.

Fun6. Hacer una función tipo short int, que devuelva 1 si valor del parámetro es primo, y 0 si no lo es.

Fun7. Hacer una función tipo char, que devuelva el carácter mayúscula del parámetro. Ha de controlar que la conversión sólo se haga si el carácter es una minúscula.

Fun8. Hacer una función tipo int, que devuelva el valor más grande de tres números pasados como parámetros.

Fun9. Hacer un programa que devuelva el M.C.D de dos números que se le pasan como parámetros. Utilizar el algoritmo de Euclides.

Fun10. Hacer una función tipo int que devuelva los segundos. Los argumentos que se le pasan son las horas y los minutos.

Fun11. Hacer un programa que muestre un menú con las siguientes opciones: 1) Calcular área de un rectángulo; 2) Calcular área de un triángulo; 3) Calcular área de un círculo; 4) Acabar. La función menú ha de ser de tipo char y las funciones que calculen las áreas han de ser de tipo float.

Fun12. Hacer un programa con las tres funciones siguientes: una función para entrar elementos en un array de una dimensión, otra para mostrar los elementos del array y otra para sumar 2 a los elementos del array. La función main() ha de tener el aspecto siguiente:

```
EntrarArray(t);
MostrarArray(t);
SumarDosArray(t);
MostrarArray(t);
```

Observe que cuando se pasa un array a una función, se trabaja con el array que se pasa.

Fun13. Hacer una función tipo int que devuelva el elemento mayor de un array que se le pasa como parámetro.

Fun14. Hacer una función tipo int que devuelva la posición del elemento más pequeño de un array que se le pasa como parámetro.

Fun15. Hacer una función de tipo int para buscar que posición ocupa un elemento dentro de un array. El array y el número buscado se pasan como parámetros. La función ha de retornar la posición del elemento, en caso que lo encuentre; si no lo encuentra ha de retornar -1.

Fun16. Hacer una función que ordene un array por el método de selección directa.

Fun17. Hacer una función que ordene un array por el método de la burbuja.

Fun18. Hacer una función que ordene un array por el método de inserción directa.

Fun19. Hacer una función que ordene un array por el método de la burbuja mejorado.

Fun20. Hacer una función que ordene un array por el método Shell.

Fun21. Hacer una función que devuelva que posición ocupa un elemento determinado en un array. Utilizar el método de búsqueda dicotómica o binária.

Fun22. Hacer un programa que pase los elementos de dos tablas ordenadas T1 y T2, a una tercera tabla T3. Los elementos de T3 han de quedar ordenados.

Fun23. Hacer una función tipo void para entrar caracteres en un array hasta que se pulse ENTER, o bien hasta que se llegue al número máximo de elementos del array. La sequencia de caracteres ha de acabar con el carácter '\0'.

Fun24. Hacer una función tipo int que devuelva el número de caracteres de un array de caracteres. No utilizar la función strlen.

Fun25. Hacer una función tipo int a la cual se le passen dos cadenes de caracteres. La función ha de retornar 0 si las dos cadenes son iguales, -1 si cad1 > cad2, y 1 si cad1 < cad2.

Fun26. Hacer un programa para contar cuantas veces aparece una sequencia de caracteres dentro de otra. Por ejemplo, que cuente cuantas veces aparece la seqüència 'LA' dentro la cadena 'La Balanguera fila fila, la Balanguera filará'.

Fun27. Hacer un programa que permita sustituir una sequencia de caracteres en una cadena por otra. Por ejemplo, en la cadena 'Hola tio como estás, Hola muy bien', buscar la sequencia 'Hola' y sustituirla por la sequencia 'Adios', 'Adios tio como estàs, Adios muy bien'. Tener en cuenta los siguientes casos:

Secuencia Buscada = Secuencia Sustituida Secuencia Buscada > Secuencia Sustituida Secuencia Buscada < Secuencia Sustituida.

Sugeréncia: Solucionar primero el caso donde las dos secuencias son iguales.

Fun28. Hacer un programa para jugar al juego del ahorcado. El programa ha de ir dibujando las diferentes partes del muñeco cada vez que se cometa un error. El programa acaba cuando ha terminado el dibujo del muñeco, o bien cuando se ha acertado la palabra.

Fun29. El juego de la vida. Este programa consta de un tablero formadoo por diferentes casillas. Cada casilla puede ser ocupada por un sol individuo. Dada una situación inicial, el juego se va desarrolando según las reglas siguientes:

Cuando una casilla está vacia y tiene tres vecinas, hay un nacimiento.
Cuando una casilla tiene menos de dos vecinois, o bien más de tres, hay una defunción.
Cuando una casilla tiene dos vecinas se mantiene la situación.

Se ha de definir un array bidimensional de 10x10 elementos, por ejemplo, y dar una situació inicial. Después, el programa ha de ir evolucionando hasta que se haya generado un número determinado de generacions que se puede entrar desde teclado. Ver el esquema siguiente:

```
Definir situació inicial;
Leer(Número_de_Generaciones);
Para y:=1 hasta Número_de Generacions Hacer
        Inici
                Nacimientos;
                Defunciones;
                Nueva_Situación;
        Fin;
```

Se ha de tener en cuenta que cuando haya un nacimiento en una casilla, este individu no se teiene en cuenta hasta la siguiente generación. De la misma manera, aunque en una casilla haya una defunción, este individuo se ha de contar como a vecino de las casillas

que tiene alrededor hasta la siguiente generación, que serà cuando desaparecerà. Vea el siguiente dibujo y el anàlisis posterior:

	1	2	3	4	5
1					
2	X	X			X
3	X			X	
4					X
5					

Si se tiene esta situación inicial:

Nacimeintos, en la casilla 2,2 porque tiene tres vecinas.
Defunciones, las casillas 2,5 y 4,5 sólo tienen un vecino,

	1	2	3	4	5
1					
2	X	X			
3	X	X		X	
4					
5					