UNIVERSIDAD DE MALAGA
DPTO. DE LENGUAJES Y C. DE LA COMPUTACION
E.T.S. DE INGENIERIA INFORMATICA

FUNDAMENTOS DE LA PROGRAMACIÓN

SEGUNDA RELACIÓN DE PROBLEMAS

1.- Dadas las siguientes declaraciones de variables en la función `main` de un determinado programa:

```
int a,b,c;
bool fin;
```

Si disponemos de los siguientes subprogramas (se muestra sus cabeceras):

```
bool uno(int x, int y)
void dos(int& x, int y)
int tres(int x)
```

¿Cuáles de las siguientes llamadas desde la función `main` a lo subprogramas son válidas? ¿Cuáles son incorrectas y por qué?

```
a) if (uno(a,b)) {      b)  dos(a,b+3)            c) fin = uno(c,5)
d) fin = dos(c,5)       e)  dos(a,tres(a))        f) dos(tres(b),c)
g) if tres(a) {         h)  b = tres(dos(a,5))    i) dos(4,c)
```

2.- Escribe un programa que lea un número natural N por teclado y dibuje un triángulo de asteriscos con base y altura N. Por ejemplo, si N=5 debería dibujarse:

```
    *
   * *
  * * *
 * * * *
* * * * *
```

3.- Escribe un programa que tome como entrada desde teclado dos números naturales (mayores que cero) "N" e "i", e imprima en pantalla el dígito que ocupa la posición i-ésima del número N. Si i es mayor que el número de dígitos de N, se escribirá en pantalla -1. Por ejemplo, para N = 25064 e i = 2, el resultado es el dígito 6, y para i = 7, el resultado es -1.

4. Escribe un programa que acepte como entrada desde teclado un número natural mayor que cero y dé como salida el resultado de sumar dos a dos los dígitos que aparecen en posiciones simétricas respecto al dígito central dentro del número dado como entrada. Por ejemplo:

para el número: 2354869
la salida es:
2 + 9 = 11
3 + 6 = 9
5 + 8 = 13
4
para el número: 6582
la salida es:
6 + 2 = 8
5 + 8 = 13

5.- Escribe un programa que acepte como entrada desde teclado un número natural mayor que uno (> 1) y dé como salida el resultado de realizar la descomposición en factores primos de dicho número.

Para realizar la descomposición en factores primos se procede de la siguiente forma:

- Paso 1: Se toma como primer primo el 2 (p = 2).
- Paso 2: Se va dividiendo n por p mientras que el resto de la división sea 0. Cada vez que se hace esto, se muestra por pantalla el valor de p y se actualiza el valor de n al valor del cociente de la división entera de n entre p.
- Paso 3: Cuando deja de cumplirse que el resto de la división de n entre p sea 0, se pasa al siguiente primo.
- Se repiten los pasos 2 y 3 mientras que n sea mayor o igual que p.

Ejemplo:

```
Introduce un numero (>1): 40
Los primos divisores de 40 son: 2 2 2 5
```

n	P
40	2
20	2
10	2
5	5
1	

```
Introduce un numero (>1): 300
Los primos divisores de 300 son: 2 2 3 5 5
```

n	P
300	2
150	2
75	3
25	5
5	5

6.- La conjetura de Goldbach dice que todo número par mayor que 2 tiene la propiedad de que es la suma de dos números primos. Diseña un algoritmo que compruebe si dicha conjetura es cierta para todos los números pares comprendidos entre dos números leídos por teclado.

Ejemplo. Para los números pares comprendidos entre 3 y 12, se cumple la conjetura:

4 = 2 + 2
6 = 3 + 3
8 = 3 + 5
10 = 3 + 7
12 = 5 + 7

Por lo que la ejecución del programa mostraría por pantalla :

```
Introduzca límite inferior: 3
Introduzca límite superior: 12
Todos los pares en el rango elegido cumplen la conjetura
```

7.- Diseña la versión iterativa del siguiente procedimiento recursivo:

```
void Rec(int n) {
    if (!f(n)) {
        {cualquier grupo de sentencias que no modifiquen el valor de n}
        Rec(g(n))
    }
}
```

donde las cabeceras de f y g se definen como:

```
bool f(int n)
int g(int n)
```

8.- Considera la siguiente función recursiva:

```
int p(int x) {
    int res;
    if (x < 3) {
        res = x;
    } else {
        res = p(x-1) * p(x-3);
    }
    return res;
}
```

Sea A(n) el número de productos realizados al ejecutar la función p cuando se llama con el argumento n. Diseña la función recursiva de A(n).

9.- El máximo común divisor (mcd) de dos números naturales p y q es el mayor entero d que divide a ambos. Un algoritmo muy conocido para calcularlo es el de Euclides. Éste utiliza dos variables, que contienen inicialmente a cada uno de los números, y trata de hacer que su contenido sea el mismo. Para ello, irá restando la menor a la mayor hasta que ambas contengan el mismo valor. En dicho momento, el valor obtenido en cualquiera de ellas es el máximo común divisor de los dos números iniciales. Por ejemplo, si P = 18 y Q = 12, el algoritmo hará que P y Q vayan tomando los siguientes valores:

Inicialmente	P == 18	y	Q == 12	(P > Q => P = P - Q)
Después	P == 6	y	Q == 12	(Q > P => Q = Q - P)
Después	P == 6	y	Q == 6	(P == Q => El mcd es 6)

Diseña una función con la siguiente cabecera para el algoritmo anterior siguiendo un enfoque recursivo:

```
int mcd(int P, int Q)
```

Diseña también un algoritmo principal (`main`) para probar la función. Para ello, se leerán de teclado los valores P y Q (asegurándose que son números > 0), se invocará a la función implementada y se mostrará por pantalla el valor devuelto por la misma.

10.- Diseña un procedimiento recursivo que lea por teclado una secuencia de caracteres de longitud arbitraria terminada en un punto, y la imprima en orden inverso (el carácter punto no se escribe, sólo es un carácter terminador). El procedimiento no tiene parámetros. Diseña también un algoritmo principal (`main`) para probar el procedimiento. Por ejemplo, si se introduce por teclado la secuencia `hola`, la salida por pantalla será `aloh.`