

**Informática**  
**Agenda 5**  
**(Francisco Mugica)**

## **I. INTRODUCCIÓN**

Esta agenda propone el trabajo a realizar entre el 21 y el 30 de octubre. La agenda pretende cubrir dos objetivos: practicar el tema de subprogramas (acciones y funciones) y comenzar la preparación del examen parcial. Los nombres de los ficheros .cpp que se entreguen deberán incluir el número del ejercicio resuelto, por ejemplo, II1c.cpp o III3ProbB.cpp.

Haré una recapitulación de los temas que entran en el parcial:

1. Objetos: Variables y Constantes
  - a. Definición de Objetos
    - i. Nombre, Tipo y Valor
    - ii. Sintaxis de nombres
    - iii. Tipos de datos elementales (int, float, char, bool, double, long long int)
    - iv. Rango de valores posibles, operadores, sintaxis
  - b. Expresiones, operaciones y evaluación, incluyendo jerarquías.
  - c. Instrucciones básicas:
    - i. Asignación =
    - ii. Conversión de tipos (casting).
    - iii. Entrada – cin >>
    - iv. Salida – cout <<
2. Estructura de un programa
  - a. Declaración de bibliotecas
  - b. Declaración de constantes
  - c. Declaración de subprogramas (prototipos)
  - d. Programa main()
  - e. Declaración de variables
  - f. Composición de instrucciones secuencial (por defecto).
  - g. Compilación, ejecución y juego de pruebas.
3. Composición alternativa
  - a. Instrucción if ()
  - b. Instrucción if - else
  - c. Anidamiento alternativas
4. Composición iterativa
  - a. Instrucción while ()
  - b. Instrucción for( ; ; )
  - c. Lectura y escritura con ficheros
  - d. Uso while(cin >>)
  - e. Anidamiento iterativas (doble bucle)
5. Subprogramas
  - a. Paso de argumentos por valor y por referencia
  - b. Funciones
  - c. Acciones

Los temas del 1 al 4 han tenido una primera evaluación con el control 1 y junto el tema 5 constituyen los temas a evaluar en el examen parcial.

Si los ejercicios de esta agenda resultaran complejos, sugiero revisar antes los guiones de laboratorio 6, 7 y 8 para preparar el tema de subprogramas, usar los recursos de minidosis de funciones y acciones y estudiar los apuntes correspondientes al tema de subprogramas.

Para los alumnos con un mal resultado en el control 1 sugiero no abordar directamente los ejercicios sino regresar a las agendas anteriores e intentar los ejercicios que consideréis más difíciles tratando de avanzar lo más rápidamente posible hasta llegar a esta agenda. La ventaja didáctica de los guiones de laboratorio, es que presentan cada tema de forma graduada y pausada con algunas explicaciones y problemas resueltos. Revisando los 5 guiones de laboratorio anteriores a medida que avancéis os ayudará a cubrir conceptos que no estén muy claros. Es más importante aseguraos que domináis cada concepto y/o ejercicio a que intentéis muchos ejercicios. Dominar un ejercicio significa que podéis hacerlo sin ningún apoyo externo y en un tiempo razonable de acuerdo a la dificultad del ejercicio. Luego ir al siguiente ejercicio hasta dominar el tema y entonces abordar el siguiente tema seleccionando aquellos ejercicios en donde no veamos la solución mentalmente. Finalmente, el recurso de minidosis es excelente para ir paso a pasito.

## II. Ejercicios de Funciones y Acciones

II.1 Escribir los prototipos de los subprogramas que resuelvan los problemas siguientes teniendo en cuenta los parámetros o información para controlar los posibles errores de los subprogramas. No es necesario desarrollar los programas.

- a. Calcular la suma de dos números reales.
- b. Calcular el producto de dos números reales.
- c. Calcular el promedio de dos enteros.
- d. Calcular el cociente y el residuo de dos números enteros.
- e. Calcular la raíz cuadrada de un número real.
- f. Resolver una ecuación lineal.
- g. Resolver una ecuación de segundo grado.
- h. Encontrar el valor más grande de una secuencia de enteros.
- i. Encontrar la media de las 5 notas de un alumno.
- j. Calcular la distancia en el plano entre dos puntos.
- k. Calcular la distancia en el espacio entre dos puntos.
- l. Mediante una letra, decir si dos líneas son iguales, paralelas y diferentes, o se cruzan.
- m. Simplificar una fracción.
- n. Decidir si dos fracciones representen el mismo número racional.
- o. Decir si un primer entero es divisible, o no, entre un segundo entero.

II.2 Resolver como mínimo: b), d), e), f) y h)

- a. Diseñar un subprograma que, dado un real, regrese su valor absoluto
- b. Diseñar un subprograma que, dados dos enteros, regrese el mayor.
- c. Diseñar un subprograma que, dados dos enteros, regrese el menor.
- d. Diseñar un subprograma que, dados dos enteros, regrese el menor y el mayor.
- e. Diseñar un subprograma que, dado un entero positivo  $i$ , calcule el  $i$ -ésimo número de Fibonacci  $F(i)$ . Recordad que:  $F(0) = 0$ ,  $F(1) = 1$ ,  $F(n) = F(n-1) + F(n-2)$
- f. Diseñar un subprograma que, dados dos positivos, calcule su cociente y su residuo sin usar los operadores '/' y '%'.
- g. Diseñar un subprograma que, intercambie los valores de dos enteros.
- h. Diseñar un subprograma que, dados dos enteros positivos, regrese su MCD.
- i. Diseñar un subprograma que, dados dos enteros positivos, calcule su MCM.
- j. Diseñar un subprograma que, dado un natural, indique si el número es primo o no.
- k. Diseñar un subprograma que, dado un natural, indique si es un número capicúa o no.

## II.3 Leer primero todos los enunciados de los ejercicios siguientes (A→N) y resolver como mínimo 3.

### Problema A

- 1) Escriure un subprograma que donat un natural que codifica un temps en el format *ssmmhh* on *ss*<60 denota segons, *mm*<60 denota minuts i *hh*<24 denota hores, retorni el seu temps equivalent en segons. Per exemple, si la funció té la entrada 020312 la funció ha de retornar 7392 segons.
- 2) Escriure un programa que donat una seqüència en el fitxer "Mod1\_Prim20.txt" on els elements són 4 naturals *t1, t2, t3* i *t4* que codifiquen temps de la forma descrita en l'apartat 1), i que a més a més representen dos intervals de temps [*t1, t2*] i [*t3, t4*] tals que *t1*<*t2* i *t3*<*t4*, escrigui el percentatge del total dels parells d'intervals que un d'ells està inclòs en l'altre. El programa ha d'utilitzar el subprograma de l'apartat 1). Per exemple, si el fitxer conté la seqüència

```
505013 505015 505011 505017
202006 202009 202007 202407
345612 455514 555515 555516
345612 555515 555513 444417
000010 000510 000010 000510
```

el programa ha d'escriure "Percentatge: 60%" doncs el primer, segon i cinquè parells d'intervals compleixen la condició esmentada.

### Problema B

- 1) Escriure un subprograma que donats dos naturals positius amb el mateix nombre de dígit calculi un natural construït dígit a dígit, on cada dígit de cada posició es calcula com la resta entre el dígit major dels dos naturals d'aquella posició i el dígit menor dels dos naturals de la mateixa posició. Per exemple, si *n* = 12345 i *m* = 54321 el resultat ha de ser 42024, si *n* = 11 i *m* = 12 el resultat ha de ser 1 i si *n* = 23 i *m* = 12 el resultat ha de ser 11.
- 2) Escriure un programa que donat el fitxer "Mod4\_Prim20.txt" que conté una seqüència de parells de naturals positius escrigui pel canal de sortida els naturals que s'obtenen al aplicar la funció de l'apartat 1), cada un en una línia diferent. A més a més, el programa ha d'escriure el percentatge de resultats respecte del total que són parell, i el percentatge de resultats també respecte del total que són múltiples de 3.

Per exemple si el fitxer conté

```
12345 54321
2468 8462
7 4
11 12
23 12
```

el resultat hauria de ser

```
42024
6006
3
1
11
```

a més a més

Percentatge de parells: 40%  
Percentatge múltiples de 3: 60%

### Problema C

Según los magos, mediante la Reducción Teosófica (RT) de un número se puede calcular el número de la suerte de una persona en base a su fecha de nacimiento (i.e. 15111999) que pondrá al descubierto una faceta importante de la persona para que pueda conocerse mejor a sí misma y tomar decisiones sobre su futuro con más sabiduría. Para calcularla necesitamos hacer uso repetido de la Suma Teosófica (ST) hasta tener una sola cifra: el número de la suerte.

La ST de un número es simplemente la suma de todas sus cifras, y se calcula de la siguiente manera:

1. Tomamos un número, (por ejemplo, el número 15111999)
2. Calculamos su ST (sumamos todos sus dígitos individuales: (1+5+1+1+1+9+9+9 = 36)
3. Mientras la ST tenga más de una cifra, como en nuestro caso, repetimos el paso 2: (3 + 6 = 9)  
→ Por lo tanto, la RT del número 15111999 es 9

Desarrollar un subprograma que calcule la RT en base al uso de la ST y utilizarlo para encontrar las RT del fichero "nums1.txt" informando por pantalla el dígito de la suerte más frecuente.

### Problema D

- 1) Escriviu un programa que donades les vendes diàries en una farmàcia de caps de d'un genèric determinat, durant un mes, pel cas de tres dosis 250, 500 i 750 mg, comptabilitzi les vendes mensuals de gener. Per exemple, l'entrada d'un dia podria ser:

5 6 9

on 5 serien les caps de 250 mg, 6 les de 500 mg i 9 les de 750 mg. D'aquestes ternes d'enters tindrem 31, una per a cada dia del mes de gener. El programa escriurà quina és la presentació de càpsules més venuda i per fer-ho utilitzarà el subprograma de l'apartat 2). Un exemple de sortida del programa podria ser:

Les caps de 500 mg són las mes venudes

- 2) Escriviu un subprograma que donades les dades (dosis, vendes mensuals) dels tres casos, retorni la més venuda.

### Problema E

- 1) Fer un subprograma que rebi un enter i calculi quants divisors té, sense comptar l'1 i ell mateix. Per exemple, el número 12 té 4 divisors (2, 3, 4, i 6).
- 2) Utilitzant el subprograma de l'apartat a), fer un programa que demani tres enters:
  - Límit inferior (Linf)
  - Límit superior (Lsup)
  - Nombre de divisors (D)

i mostri la llista dels números dins de l'interval entre el límit inferior i superior (ambdós inclosos) que tinguin un nombre de divisors major o igual que D. La llista ha de mostrar cada número i al costat, entre parèntesis, el nombre real de divisors.

Exemple: Entrades:

Limit inferior: 26  
Limit superior: 52  
Min. divisors: 6

Sortida:

36(7) 48(8)

### Problema F

- 1) Escriure un subprograma que donats cinc nombres enters retorni cert si els nombres estan ordenats en ordre estrictament creixent i fals en cas contrari. Per exemple si els nombres són -5 -1 2 4 55 el subprograma ha de retornar cert i si els naturals són -99 0 78 3 43 ha de retornar fals.
- 2) Escriure un programa que llegeixi una seqüència de quintets de enters del fitxer Mod3.txt i calculi la longitud mitjana dels intervals (major-menor) formats entre el primer i últim element d'aquells quintets ordenats. Per exemple,

si el fitxer conté els nombres:

-11 -12 -10 -9 -8  
-5 -1 2 4 55  
2 6 6 8 12  
-34 -8 -1 2 1  
1 2 34 46 51  
-20 -18 -5 -1 10

el programa ha de retornar: 46.6667  
(60+50+30)/3.

### Problema G

- a) Escriviu un subprograma (acció o funció) que determini si un codi simplificat ISBN de llibres, de nou dígit, és correcte. Per validar un codi ISBN, cal sumar nou cops el primer dígit, més vuit cops el segon,..., més un cop el darrer dígit. Si la suma és divisible per onze, es considera que el codi ISBN és vàlid. Per exemple, el codi ISBN 201103311 és vàlid, doncs

$$9*2 + 8*0 + 7*1 + 6*1 + 5*0 + 4*3 + 3*3 + 2*1 + 1*1 = 55 = 5*11.$$

- b) Escriure un programa que donat n codis ISBN que poden ser vàlids o invàlids, on n es un natural que es dona també abans per l'entrada, compti quants codis ISBN vàlids hi ha. El subprograma ha d'utilitzar el subprograma definit en l'apartat a).

## Problema H

- a) Escriure un subprograma que **donats** tres nombres enters de tres xifres **retorni** un premi de 50 euros si els nombres són iguals i el dígit del mig és el més gran dels tres dígit; un premi de 20 euros si els tres nombres són iguals però el dígit del mig no és el més gran; i en qualsevol altre cas un premi de 0 euros.
- b) Escriure un programa que llegeixi una seqüència d'elements formats per tres enters de tres xifres del fitxer **Mod5.txt** i calculi els percentatges de premis de 20 i 50 euros. El programa ha d'utilitzar el subprograma del apartat 1). Per exemple,

si el fitxer conté els nombres:

```
123 456 789
111 111 222
451 451 451
345 234 345
123 123 123
444 555 666
234 567 890
123 123 122
345 543 345
121 121 121
```

el programa ha de retornar:

```
Premis de 20 euros: 10.0%
Premis de 50 euros: 20.0%
```

## Problema I

- a) Escriure un subprograma que **donats** dos nombres enters positius  $n1$  (de màxim nou xifres) i  $n2$  (de 5 xifres) concateni **la suma** de los dígit de  $n1$  a la esquerra de  $n2$  i **retorni** el nombre resultant. Per exemple, si  $n1$  es 999999999 i  $n2$  es 54321 el subprograma ha de retornar el nombre 8154321.
- b) Escriure un programa que llegeixi una seqüència d'elements formats per parelles de nombres enters positius  $n1$  (de màxim nou xifres) i  $n2$  (de tres xifres) del fitxer **Mod4.txt** i calculi el percentatge de nombres obtinguts amb el subprograma l'apartat a) que tenen *set* xifres. Per exemple,

si el fitxer conté els nombres:

→ el programa ha de retornar:

```
12 34567
14121 11111
88 34561
652 56798
121111 19111
923 12345
0 12345
```

50%

## Problema J

Se us demana que desenvolueu un programa per realitzar certes operacions sobre una llista d'enters d'un fitxer. Concretament, haurà de llegir un enter  $n$  i realitzar les 3 operacions següents sobre l'enter :

- Comptar les xifres de  $n \rightarrow$  **nxifres**.
- Factorial de l'enter **nxifres** calculat en a)
- Determinar si l'enter  $n$  introduït és positiu.

Heu de fer el programa seguint els següents passos:

- definir tres funcions, una per a cada operació sol·licitada.
- Un programa `main()` que llegeixi una llista d'enters d'un fitxer **entrada3.txt** i mostri per pantalla el resultat de les 3 operacions descrites per cada enter de la llista usant les funcions descrites en l'apartat anterior.

## Problema K

Una empresa de transportes que actúa en Catalunya, guarda la información del destino de los paquetes en un fichero. La información corresponde al código postal (CP) del destino de los paquetes.

Hacer un programa para leer el fichero y procesar sus datos teniendo en cuenta:

- Una función que reciba el CP y devuelva el nombre de la provincia al que pertenece. [3p]
- Leer el fichero y visualizar el CP y su correspondiente nombre de la provincia. [3p]
- Visualizar la provincia con mayor número de paquetes [1p]

Datos: un código postal está formado por 5 dígitos: ABCDE. Los primeros dos dígitos corresponden a la provincia y los últimos 3 dígitos corresponden a su ubicación dentro de la provincia. Para Catalunya hay:

08 : Provincia de Barcelona (por ejemplo 08020)

17 : Provincia de Girona (por ejemplo 17007)

25 : Provincia de Lleida (por ejemplo 25795)

43 : Provincia de Tarragona (por ejemplo 43870)

Fichero: CP.txt: 08020 25795 08007 17300 17255 25080 25798 17007 25199 25798 43870  
08030 43820 17003 08001 1700 08004 43850 08010 43006

Salida del programa:

08020:Barcelona	43870:Tarragona
25795:Lleida	08030:Barcelona
08007:Barcelona	43820:Tarragona
17300:Girona	17003:Girona
17255:Girona	08001:Barcelona
25080:Lleida	1700: Girona
25798:Lleida	08004:Barcelona
17007:Girona	43850:Tarragona
25199:Lleida	08010:Barcelona
25798:Lleida	43006:Tarragona

Barcelona tiene el mayor numero de paquetes: 6

## Problema L

- Escribiu un subprograma (acció o funció) que donat una minúscula c la codifiqui i retorni el seu codi ASCII rotat k posicions, on k és un paràmetre del subprograma que se sap que ha de ser menor o igual que 26. Per exemple si c = 'b' i k = 5 es retorna el caràcter 'g', i si c = 'y' i k = 3 es retorna el caràcter 'b'.
- Escribiu un programa que donat un natural k menor o igual que 26 i una seqüència de caràcters acabada en un punt, escrigui la seqüència codificada amb el subprograma de l'apartat a) utilitzant el natural k donat.

## Problema M

- Escribiu un subprograma (acció o funció) que donat un natural ens doni el seu dígit central. En el cas en que no existeixi el subprograma ha de retornar el dígit 0.  
**Exemples:** N = 45678 dígit central = 6  
N = 1234 dígit central = 0
- Escribiu un programa que donat un natural n i una seqüència de n naturals, introduïts per teclat, doni la suma de tots els seus dígitos centrals. El programa ha d'utilitzar el subprograma definit en l'apartat a).

## Problema N

- Dissenyau una funció que donat tres nombres (prec article, IVA i descompte) calculi el cost aplicant el descompte al prec i després el IVA.
- Escribiu un programa que calculi la compra total de N articles (prec , IVA i descompte). N es un enter introduït abans de la sèrie de tripletes. El programa ha d'utilitzar el subprograma definit en l'apartat a).
- Escribiu un programa que calculi el nombre d'articles (preu, descompte i IVA) comprats just abans de superar 100€. El programa ha d'utilitzar el subprograma definit en l'apartat a).

### III. **Autoevaluaciones.**

III.1 Aprobar (nota > 8.5) las 6 autoevaluaciones:

- Autoavaluació 1 Primers Programes
- Autoavaluació 2 Primers Programes
- Autoavaluació d'Expressions
- Autoavaluació Composició alternativa
- Autoavaluació Composició iterativa
- Autoavaluació Accions/Funcions

### IV. **Preparar el examen parcial.**

IV.1 En el tema “Examen parcial” del curso en Atenea, estudiar los 3 ejemplos de exámenes parciales. Resolver un ejercicio de cada tipo (estos estarán a partir del día 22).

IV.2 Resolver el siguiente examen parcial pasado (se recomienda hacerlo papel y comprobar en ordenador)

- S'ha de posar el nom, cognoms, DNI i especialitat a cada full que s'entregui.
- Qualsevol hipòtesi que es faci sobre un problema s'haurà d'argumentar.
- És indispensable **fer cada problema en un full separat**. Si algun problema no es fa, s'ha d'entregar igualment el corresponent full en blanc amb les dades personals.

**Problema 1** (4 punts). Imitarem una calculadora usando la información del fichero "Operacions.txt", en cada línea tenemos dos **operandos** (los números que se operan) y un carácter indicando un **operador** (la operación aritmética que se realiza). Se solicita:

- a) (1 punto) desarrollar un programa que obtenga los resultado de cada una de las operaciones del fichero. Cada operación consta de tres entradas: **un operando** seguido por **un operador** de los seis permitidos (+, -, \*, /, %, ^), seguido de **otro operando**, siendo los operandos números enteros. Algunos ejemplos de operaciones son: 3 - 5, -9 % 2 i -2 ^ 8.

La dificultad es que para hacer el programa, o cualquier subprograma, **se exige que únicamente se usen las tres operaciones permitidas: la suma +, la resta - i el cambio de signo**. Solamente para la suma y la resta se pueden usar directamente los operadores de C++ conseguidos, mientras que para el resto de las operaciones será necesario crear los subprogramas siguientes (donde no está permitido ni utilizar los operadores \*, / o %, ni tampoco ninguna biblioteca matemática);

- b) **PRODucte** (0,75 puntos) – recibe dos enteros y regresa el producto de estos.
- c) **POTencia** (0,75 puntos) – recibe dos enteros: *base* y *exp* y regresa el resultado de la potencia  $base^{exp}$ . La única restricción es que *exp* sea mayor o igual a cero. Se puede usar el subprograma PRODucte.
- d) **DIVisio** (1,25 puntos) – recibe dos enteros *D* (dividendo) y *d* (divisor) i regresa dos enteros como resultados: el cociente y su residuo. En cas que *d* sea igual a cero, el subprograma regresará las constantes (ya conocidas por el compilador de C++) **INT\_MAX** o **INT\_MIN** dependiendo del signo de *D*, los cuales son los valores posibles máximo y mínimo del tipo **int** en C++. Por definición, el signo del residuo es el mismo signo del dividendo. Ejemplo se entradas del fichero "Operacions.txt" y sys salidas correspondientes son:

Entrada	Sortida	Entrada	Sortida	Entrada	Sortida
8 + 6	Suma = 14	2 / 3	Quocient = 0	9 % -2	Residu = 1
-2 - 6	Resta = -8	2 % 3	Residu = 2	-9 / 0	Quocient = -2147483648
-2 - -6	Resta = 4	8 / 2	Quocient = 4	-9 % 0	Residu = 0
2 * 0	Producte = 0	8 % 2	Residu = 0	9 / 0	Quocient = 2147483647
0 * 3	Producte = 0	9 / 4	Quocient = 2	9 % 0	Residu = 0
2 * 0	Producte = 0	9 % 4	Residu = 1	2 @ 3	Operacio Invalida
0 * 0	Producte = 0	-9 / 2	Quocient = -4	2 ^ 0	Potencia = 1
2 * 3	Producte = 6	-9 % 2	Residu = -1	0 ^ 3	Potencia = 0
-2 * 3	Producte = -6	-9 / -2	Quocient = 4	2 ^ 3	Potencia = 8
2 * -3	Producte = -6	-9 % -2	Residu = -1	-2 ^ 3	Potencia = -8
-2 * -3	Producte = 6	9 / -2	Quocient = -4	0 ^ 0	Potencia = 1
0 / 3	Quocient = 0	3 / 3	Quocient = 0		
0 % 3	Residu = 0	3 % 3	Residu = 0		

**Problema 2** (3 punts)

Un número entero  $n > 0$  es **abundante** si es menor que la suma de sus divisores propis (i.e. todos sus divisores excepto el propio  $n$ ); se dice en cambio que  $n$  es **deficiente** si es mayor que la suma sus divisores propios. Si  $n$  no es ni abundante ni deficiente, entonces es **perfecto**. Per exemple:

- 12 es abundante, porque 1,2,3,4,6 son sus divisores propios y  $1+2+3+4+6=16 > 12$ .
- 15 es deficiente porque 1,3,5 son sus divisores propios y  $1+3+5=9 < 15$ .
- 6 es perfecto porque 1,2,3 son sus divisores propios y  $1+2+3=6 == 6$ .

Escribir un subprograma (acción o función) que, dados dos valores enteros positivos  $a$  i  $b$  con  $a < b$ , regrese tres valores: la cantidad de números abundantes, la cantidad de números deficientes y la cantidad de números perfectos que hay en el intervalo  $[a,b]$ .

Nota histórica: Los números naturales van ser clasificados en deficientes, abundantes i perfectos per *Nicomachus* en el su *Introductio Arithmetica* (circa 100 AD).



- S'ha de posar el nom, cognoms, DNI i especialitat a cada full que s'entregui.
- Qualsevol hipòtesi que es faci sobre un problema s'haurà d'argumentar.
- És indispensable **fer cada problema en un full separat**. Si algun problema no es fa, s'ha d'entregar igualment el corresponent full en blanc amb les dades personals.

**Problema 3** (3 punts)

El siguiente programa carga desde un fichero una secuencia de caracteres y muestra por pantalla si hay más letras mayúsculas que minúsculas, así como el porcentaje de vocales entre aquellos caracteres que sean una letra. Para ello utiliza un subprograma que, dada una letra, regresa si es mayúscula, minúscula y si es vocal o no. Lamentablemente el código está incompleto. Hay una marca **## n ##** que indican la falta de código i que necesitan ser remplazadas para que funcione correctamente. Puede ser un símbolo (por ejemplo, >= ), una línea de código (por ejemplo, **float** xx, pi=3.1416;) o partes de una instrucción (por ejemplo: **else if** (a==0) ). Nunca representan más de una instrucción. **Escriba la lista del código que falta.**

```
#include<iostream>
## 1 ##
using namespace std;

## 2 ##
int main()
{
    ifstream fin("caracters.txt");
    char ca;
    ## 3 ##

    int conta_let=0, conta_ma=0, conta_vo=0;
    while(## 4 ##)
    {
        if(ca>='A' && ca<='Z' ## 5 ##){
            ## 6 ##
            deletras(ca,esmima,esmivo);
            if(esmima==true) conta_ma++;
            if(esmivo==true) conta_vo++;
        }

        cout << conta_let << endl << conta_vo << endl;
        if (conta_let==0) cout << "NO HI HA CAP LLETRA EN EL FITXER" << endl;
        else
        {
            cout << "EL PERC. DE VOCALS ES: " << ## 7 ## << endl;
            if (conta_ma >= ## 8 ##)
                cout << "HI HA MES (o MATEIXES) MAJUSCULES QUE MINUSCULES" << endl;
            else cout << "HI HA MENYS MAJUSCULES QUE MINUSCULES" << endl;
        }
    }

    void deletras(char le ## 9 ##)
    {
        es_MAY=false;
        es_voc=false;
        if (## 10 ##)
        {
            es_MAY=true;
            if(le=='A' || le=='E' || le=='I' || le=='O' || le=='U') es_voc=true;
        }
        else
        {

```