

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 (3 punts)

Desenvolupar un programa que llegeixi una seqüència de caràcters acabada en * i imprimeixi per pantalla de quin dels següents tipus hi ha més caràcters: lletres minúscules, lletres majúscules o que no siguin lletres. Per a l'elaboració d'aquest programa, s'ha d'escriure i utilitzar el subprograma següent:

Un subprograma que rep per paràmetres un caràcter i indica si el seu tipus és (1) lletra majúscula; (2) lletra minúscula; (3) no és una lletra.

Per exemple, si l'entrada és: A Q U E S T A a s s i g n a t u r a e m M O L A M O L T !!! *
 La sortida serà: Hi ha mes lletres majuscles.

Problema 2 (2.5 punts)

El següent codi resol el següent problema: Un Banc Central ha de determinar quants i quins d'un grup de 120 bancs fallen un "test d'estrès". Si C és el capital d'un banc i A és el valor dels seus actius, llavors el percentatge de capital de solvència del banc o "core capital" és $(C/A) \cdot 100$. Un banc falla el test d'estrès si el seu "core capital" és menor del 5.5%. El programa llegeix les dades dels 120 bancs (nom, valor del seu capital, i valor dels seus actius) i imprimeix per pantalla el nom i el core capital d'aquells bancs que no aproven el test d'estrès així com el nombre total de bancs que no aproven.

Què hi farem! el codi està incomplet... Hi ha unes marques #N# que indiquen que hi falta codi i que cal que ompliu per tal de que funcioni correctament. Pot ser un símbol (per exemple >=), una línia de codi (per exemple float xx, pi=3.1416;) o parts d'una instrucció (per exemple: else if (a==0)). Mai representa més d'una instrucció. Escribiu la llista del codi que falta.

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

#2#

int main(){
    int i=0, #3#;
    float capi, val, coca;
    string nombanc;

    while (i<10)
    {
        cin >> nombanc >> capi >> val;
        #4# = corecap(capi, val);
        if (coca < #5# )
        {
            cout << "El banc " << nombanc << " te un core capital de: "
                << coca << "%" << endl;
            nocumplen++;
        }
        #6#
    }
    cout << "Hi ha " << #7# << " bancs que no arriben al core capital minim"
        << endl;
}

#8# corecap( #9# , float v_actius)
{ return (cap/v_actius)*#10# ; }
```

Problema 3 (2 punts)

Arriva la loteria de Nadal! El programa següent és un generador de nombres guanyadors. Dieu:

a) ¿Quin càlcul fa **xino** ?

b) ¿Quin càlcul fa **xano** ?

c) ¿Quina seria la sortida per pantalla si l'entrada fos el vostre telèfon (fixe o mòbil, que heu de escriure a la resposta)?

Feu la traça del **programa principal** (mostrant el canvi del valor de les variables a mesura que l'algorisme s'executa).

Nota bene: Si compressiu el nombre i us toquès un premi, el 10% per les vostres profes d'Informàtica

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

int xino (int tutele)
{
    int colapso=0, porjo=10;
    while (tutele!=0)
    {
        colapso += tutele%porjo;
        tutele = tutele/porjo;
    }
    return colapso;
}

void xano (int& xa, int& no)
{
    int xoni;
    if(xa>no)
    {
        xoni=xa;
        xa=no;
        no=xoni;
    }
}

int main()
{
    int t, sute, su, te, le=1, contale=0;
    cout << "Introduzca su numero de telefono (9 cifras)" << endl;
    cin >> t;
    sute=xino(t);
    su = sute/10;
    te = sute%10;
    xano(su,te);
    while(te <= te/2)
    {
        if(te%le==0)
        {
            cout << le;
            contale++;
        }
        le++;
    }
    while(contale<5)
    {
        cout << contale;
        contale++;
    }
}
```

Problema 4 (2.5 punts)

En un supermercat es volen apilar les llaunes de la següent manera:

- # la fila de dalt de tot només ha de tenir una llauna, i cadascuna de les files següents (en sentit descendent) ha de tenir una llauna més que la fila anterior.
- ## Feu un subprograma que, donat un nombre positiu que representi un nombre de
- ### llaunes, retorni algun valor que indiqui si és possible o no apilar les llaunes de la
- #### manera indicada i, en cas afirmatiu, el nombre de files que tindrà la pila. En el cas de
- #### que no sigui possible, el nombre de files serà zero.