EXERCICIS RESOLTS A CLASSE DE TEORIA

1. [Factorial.cpp] Realitzeu un programa que demane un nombre enter pel teclat i en torne el factorial. Caldrà crear una funció `factorial(x)`, la qual serà invocada des del programa principal.

2. [Potència.cpp] Escriviu un programa que continga una funció `potencia(x,y)`, implementada per vosaltres, que calcule el valor de x elevat a y mitjançant multiplicacions. Açò és, no podreu fer servir la funció `pow`. Suposeu que y és sempre un valor enter positiu.

3. [Nombree.cpp] El valor de e^x es pot aproximar mitjançant el sumatori següent:

$$e^x = \sum_{i=0}^{n} \frac{x^i}{i!}$$

Escriviu un programa que demane a l'usuari el valor de *x* i el valor de *n* i mostre per pantalla el valor de l'aproximació de e^x per a la *x* i la *n* introduïdes. La funció principal haurà de cridar a una funció `aproximacio(x,n)` que retorne el valor buscat. Compareu el resultat de l'aproximació calculada amb el resultat que s'obté mitjançant la funció `exp(x)` de la llibreria matemàtica i mostreu aquesta diferència per pantalla.

4. [Euclides.cpp] El màxim comú divisor (MCD) de dos nombres P i Q és el major enter D que divideix ambdós. Un algorisme molt conegut per a calcular-lo és el d'Euclides. Aquest algorisme utilitza dues variables, que a l'inici contenen cadascun dels nombres anteriors, i tracta de fer que el seu contingut siga el mateix. Per aconseguir-ho, restarà successivament el menor nombre al major fins que les variables allotgen el mateix valor. Aleshores, aquest valor és el màxim comú divisor dels dos nombres inicials.

Per exemple, si P = 18 i Q = 12, l'algorisme farà que P i Q prenguen successivament els valors següents:
A l'inici P = 18 i Q = 12 (P > Q => P ←P - Q)
Després P = 6 i Q = 12 (Q > P => Q ← Q -P)
Després P = 6 i Q = 6 (P = Q => El MCD és 6)

5. [MCM.cpp] Realitzeu un programa que calcule el mínim comú múltiple de dos nombres introduïts pel teclat. Useu la funció `mcd` implementada adés ja que cal recordar que:

$$MCM\ (a,b) = \frac{a * b}{mcd\ (a,b)}$$

[Obligatori 2]

6. [Combinatoris.cpp] Escriviu un programa que calcule el nombre combinatori de *m* sobre *n*. Dividiu el programa en funcions (empreu una funció que calcule el factorial d'un nombre que rep com a paràmetre).

$$C_m^n = C_{m,n} = C(m,n) = \binom{m}{n} = \frac{m!}{n!(m-n)!}$$

EXERCICIS PER RESOLDRE

7. [PotenciaV2.cpp] Escriviu un programa que continga la funció `potencia(x,y)`, implementada per vosaltres, que calcule el valor de **x** elevat a **y** mitjançant multiplicacions, on **x** i **y** són dos nombres enters. Feu-la servir en el programa principal per mostrar $\mathbf{x}^{\mathbf{y}}$ i $\mathbf{y}^{\mathbf{x}}$. No es pot emprar la funció `pow`.

8. [EsPrimer.cpp] Escriviu una funció que retorne si un nombre, que se li passa com a paràmetre, és primer o no. Empreu-la per fer un programa que, donat un nombre introduït pel teclat, mostre per pantalla tots els nombres primers inferiors a aquest.

9. [Divisió.cpp] Escriviu una funció que, donats dos enters positius x i y, calcule la divisió entera i el residu de la divisió fent servir només l'operador de subtracció.

10. [Racionals.cpp] Dissenyeu un programa que demane els quatre valors que componen el numerador i el denominador de dos nombres racionals i construïu una mòdul que reba aquests valors i torne la suma en forma de nombre en coma flotant. A continuació, feu un segon mòdul que faça el mateix càlcul, però que torne tant el numerador com el denominador del resultat.

11. [VolumEsfera.cpp] Dissenyeu un programa que demane el radi d'una esfera des del programa principal. Aquest valor serà passat a una funció `volum(r)` perquè en calcule el volum. El missatge que mostre el valor del volum de l'esfera per pantalla es farà des del programa principal.

[Optatiu 1]

12. [ÀreesVolums.cpp] Amplieu el programa "VolumEsfera.cpp" perquè calcule les àrees i els volums de diverses figures geomètriques. El programa mostrarà per pantalla el menú següent:

1- Àrea i volum d'una esfera.
2- Àrea i volum d'un cub.
3- Àrea i volum d'un cilindre.
4- Eixir.

Aquest menú es repetirà fins que l'usuari prema l'opció 4. Les àrees i els volums de cada opció es calcularan amb funcions diferents que seran cridades des de la funció principal, que serà on es trobarà el menú.

[Obligatori 4]

13. [NormalitzaVector.cpp] Escriviu un programa que, donades les components d'un vector (x1,y1), el normalitze. S'haurà de crear una funció que tornarà el valor de les dues components del vector normalitzades. Aquesta funció serà invocada des de la funció principal.

14. [Matemàtic.cpp] Escriviu un xicotet programa matemàtic que permeta realitzar els càlculs següents:

1- Calcular el perímetre d'una circumferència.
2- Calcular l'àrea d'un cercle.
3- Calcular el volum d'un cilindre

4- Calcular el volum d'una esfera.
5- Calcular el volum d'un con.

El programa mostrarà per pantalla un menú on l'usuari podrà triar cadascuna d'aquestes opcions a més a més d'una altra per eixir del programa. Primerament, feu el disseny descendent d'aquest problema.

[Obligatori 3]

15. [Distància.cpp] Escriviu un programa que calcule la distància de dos punts descrits per les seues coordenades (x_1, y_1) i (x_2, y_2). Haureu de crear una funció de l'estil `distancia(x1,y1,x2,y2)`, que serà cridada des de la funció principal i tornarà el valor de la distància. Empreu, a més a més, una funció per a llegir les coordenades d'un punt.

16. [Gravetat.cpp] Escriviu un programa que calcule la força d'atracció gravitatòria entre dues masses m1 i m2 situades en les coordenades (x1,y1) i (x2,y2). Caldrà crear una funció `fgrav(m1,x1,y1,m2,x2,y2)`, que serà invocada des del programa principal i tornarà el valor de la força. La funció `fgrav` cridarà a la funció `distància(x1,y1,x2,y2)` de l'exercici "Distància.cpp".

[Obligatori 5]

17. [Aleatoris.cpp] Si volem obtindre un nombre aleatori dins d'un rang determinat [R1…R2], podem cridar la funció `rand` i modificar el valor que torna de la manera següent:

$$y = \frac{x \cdot (R2 - R1)}{RAND\_MAX} + R1, \text{ on } \texttt{x= rand()},\ x \in [0 \ldots RAND\_MAX],\quad y \in [R1 \ldots R2]$$

Per iniciar el generador de nombres aleatoris sense haver de proporcionar explícitament una llavor diferent per a cada execució, podem cridar `srand(time(NULL))`, ja que `time(NULL)` torna l'hora actual en dècimes de segon. Feu una funció `aleat(n,m)` que genere enters aleatoris en el rang `[n…m]` i que torne el nombre aleatori generat. A continuació, feu un programa que empre aquesta funció per generar i mostrar per pantalla 10 nombres aleatoris en un rang triat per l'usuari.

[Obligatori 6]

18. [SeMultiplicar.cpp] Escriviu un programa que comprove si l'usuari sap multiplicar. El programa generarà dos enters aleatoris entre 2 i 9 i preguntarà: "`¿n x m?`". Llavors, l'usuari introduirà un valor i l'ordinador haurà de respondre: `Correcte o Incorrecte`. Aquest procés es repetirà indefinidament fins que l'usuari conteste amb el valor 0. Empreu la funció creada en l'exercici "Aleatoris.cpp".

[Optatiu 2]

19. [Endevina.cpp] Escriviu un programa que genere un nombre aleatori entre 0 i 99 i que demane a l'usuari que l'endevine tot i dient quants intents li han fet falta. L'ordinador donarà pistes al jugador sobre si el nombre que ha introduït és major o menor que el que ha d'endevinar. Empreu la funció creada en l'exercici "Aleatoris.cpp".

20. [ProdEscalar.cpp] Escriviu un programa que calcule el producte escalar de dos vectors descrits per les seues coordenades (x1,y1) i (x2,y2). S'haurà de crear una funció, amb el format `producte_escalar(x1,y1,x2,y2)`, que tornarà el valor del producte. Aquesta funció serà cridada des de la funció principal.

21. [Mastermind.cpp] Escriviu un procediment per a jugar al **mastermind**. En el començament, el jugador 1 ha d'introduir una combinació secreta de 4 enters. A continuació, després d'esborrar la pantalla, el jugador 2 anirà introduint combinacions fins encertar la combinació secreta. Cada cicle, el programa indicarà el nombre d'encerts aconseguit.

22. [CanviHorari.cpp] Escriviu un programa que convertisca valors horaris en notació de 24 hores a notació de 12 hores. Per exemple, el programa convertirà 14:25 a 2:25 PM. L'entrada del programa seran dos enters. Almenys hi haurà tres funcions: una per a l'entrada, que comprove que les dades són vàlides; una altra per a la conversió i una tercera per a l'eixida. Incloeu-hi un cicle que permeta a l'usuari repetir aquesta operació amb altres valors d'entrada fins que indique que desitja acabar el programa amb algun valor sentinella.

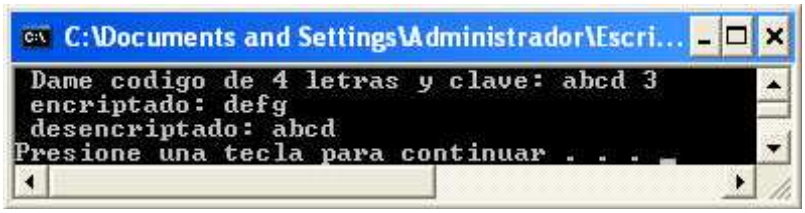
[Optatiu 3]

23. [Targetes.cpp] Escriviu un programa que imprimisca diverses targetes per pantalla. L'usuari podrà triar el tipus de targeta des del menú que oferirà el programa (Tipus de targeta: "Bon Nadal", "Feliç any nou", "Feliç aniversari", "Enhorabona" i "Ets un sol"). Empreu caràcters i símbols per a realitzar els dibuixos. Un exemple de targeta:

```
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&****    **********    ****&
&*  *    BON NADAL!    *  *&
&****    **********    ****&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

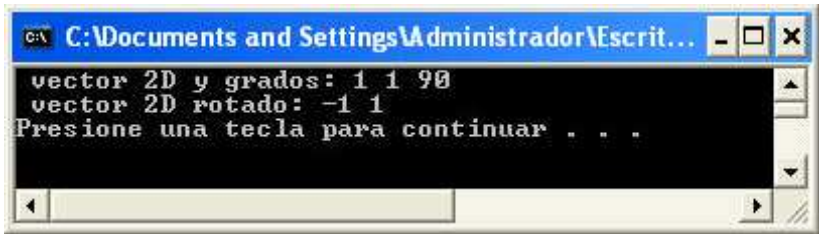
Nota: Feu servir una funció per a cada tipus de targeta.

24. [Xifrat.cpp] Escriviu una funció que xifre un codi de 4 caràcters. Per tal de fer-ho rebrà, a més a més dels 4 caràcters, un nombre enter que serà sumat al codi ASCII de cadascun dels caràcters que es volen xifrar. Així mateix, escriviu una funció complementària (`desxifrar`) i comproveu el funcionament d'ambdues (vegeu figura).



25. [Vectors.cpp] Escriviu una funció que permeta a l'usuari fer operacions sobre vectors 2D situats a l'origen de coordenades. Les funcions que cal implementar són:

- Convertir un vector 2D de coordenades polars (mòdul, angle) a cartesianes (x, y) i viceversa.
- `suma (int x1, int y1, int x2, int y2, int &xres, int &yres)` → suma dos vectors i torna el resultat sobre (xres, yres).
- Resta: resta dos vectors.
- Producte escalar ← x1*x2 + y1*y2.
- `girar(int x, int y, int angle, int &xres, &yres)` → torna sobre (xres, yres) el vector (x,y) girat <angle> graus en el sentit contrari a les agulles del rellotge (vegeu figura).



[Obligatori 1]

26. [Errors.cpp] Els programes següents són incorrectes o no donen el resultat esperat. Indiqueu per què i escriviu-los de manera correcta:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

void es_parell(void)
{
    if((valor_llegit % 2) == 0)
        cout<< "El valor "<<valor_llegit<<" és parell\n";
    else
        cout<< "El valor "<<valor_llegit<<" és imparell\n";
}

int main(void)
{
```

```
    int valor_llegit;

    cout << "Introdueix un valor";
    cin >> valor_llegit;
    es_parell();

    system("PAUSE");
    return 0;
}
-------------------------------
#include <iostream>
#include <stdlib.h>
using namespace std;

void demana_caracter(char lletra)
{
    do
    {
        cout <<"Introdueix una lletra minúscula\n";
        cin >> lletra;
    }
    while((int(lletra)<int('a'))|| ((int(lletra)> int('z'));
}

int main(void)
{
    char meua_lletra  = '?';

    demana_caracter(meua_lletra);
    cout << "La lletra llegida és" <<meua_lletra<< endl;

    system("PAUSE");
    return 0;
}
```