

Tema 4: Funcions i procediments

Problemes marcats en groc → Problemes avaluables

La resta de problemes són de dificultat intermèdia entre els avaluables. Si trobeu dificultats en passar d'un problema avaluable al següent, feu algun/s dels problemes no marcats com avaluables per a tenir un aprenentatge més progressiu.

Problemes marcats en blau → Problemes opcionals (no avaluables) però amb VPL disponible a Caronte per a auto-avaluació.

Problema 1

Volem saber quants equips diferents pot realitzar un entrenador de bàsquet si té 10 jugadors en la plantilla. Això ho podem calcular com:

$$\frac{n!}{r!(n-r)!} = \binom{n}{r}$$

on: n=10 número de jugadors en la plantilla
 r= 5 número de jugadors que pot alinear

Volem fer un programa que ens serveixi per a qualsevol esport. Per a fer-ho, feu una funció que calculi el factorial d'un nombre que se li passa com a paràmetre.

A continuació, fer el programa principal que demani a l'usuari els valors de n (número de jugadors que té a la plantilla de l'equip) i r (número de jugadors que pot alinear al seu esport). Després, ha de fer les crides necessàries a la funció que calcula el factorial per a calcular el nombre de combinacions dels n jugadors agafats de r en r que pot fer. Finalment, el programa mostrarà el missatge: "El nombre d'equips que es poden formar es XX" on XX serà el nombre de combinacions calculades pel programa.

Problema 2

Fer una funció que retorni un valor aleatori entre dos valors mínim i màxim passats com a paràmetres.

Fer també un programa principal que demani a l'usuari que introdueixi els valors mínim i màxim pel teclat i cridi a la funció per obtenir un valor aleatori entre aquests dos límits. El valor retornat per la funció, es mostrarà per pantalla amb un missatge: "Numero aleatori entre els valors minim i maxim: XX" on XX serà el valor aleatori retornat per la funció

Problema 3

Fer una funció que es digui Alerta i que rebi com a paràmetre un nombre enter i retorni un enter.

La funció ha d'escriure un missatge que digui "Alerta: Queden XX segons" on XX és l'enter passat com a paràmetre. En el cas que el paràmetre sigui igual 0, el missatge ha de ser "Alerta: S'ha acabat el temps". La funció retornarà un 0.

Fer el programa principal on es demana un enter (nombre de segons). El programa principal haurà de fer un compte enrere i cada vegada que es descompti un segon s'haurà de cridar la funció Alerta.

NOTA.- Utilitzeu:

- la funció Sleep(int MS) en VC fent un include de #include <windows.h>
- la funció sleep(int S) en Caronte fent un include de #include <unistd.h>

Problema 4

Volem millorar el programa fet anteriorment que demanava a l'usuari que introduís una hora en format HH:MM:SS i mostrés l'hora després d'incrementar un segon.

Per evitar repeticions de codi, implementeu un procediment anomenat DuesXifres que rebrà un enter com a paràmetre. La funció comprovarà si l'enter passat com a paràmetre és menor que 10 i en cas afirmatiu imprimirà un 0 per pantalla. En cas contrari, no farà res.

Utilitzeu aquesta funció al programa principal per a fer la impressió de l'hora per pantalla.

Problema 5

Fer una funció que retorni el resultat de fer una divisió entera. La funció tindrà 3 paràmetres de tipus int. Els dos primers seran el dividend i el divisor (en aquest ordre). El tercer serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 0 si s'ha pogut realitzar la divisió i un 1 en cas contrari (divisió per zero).

Aquesta funció s'utilitzarà en un programa principal que demanarà a l'usuari que introdueixi el dividend i el divisor per teclat i, després de cridar a la funció, mostri el resultat de la divisió amb un missatge "Resultat de la divisió: XX" on XX serà el valor calculat amb la funció, o bé mostri el missatge "Error: Divisió per zero" si la funció ha retornat un 0.

Problema 6

Volem tornar a millorar el programa fet anteriorment que demanava a l'usuari que introduís una hora en format HH:MM:SS i mostrés l'hora després d'incrementar un segon.

Per a seguir els principis de la programació modular, ara volem crear un procediment anomenat "IncrementaHora" que realitzarà l'increment d'un segon a l'hora. Aquesta funció rebrà com a paràmetres els valors de l'hora, els minuts i els segons (en aquest ordre).

Problema 7

Fer una funció que retorni les centenes d'un nombre. La funció tindrà un paràmetre de tipus int i retornarà el valor de les centenes del nombre que li ha estat passat com a paràmetre. (Ex.- Si el valor que es passa és el 21378 la funció retornarà un 3.)

A continuació fer el programa principal main() que ha de demanar a l'usuari que introdueixi un nombre enter i ha de realitzar la lectura per teclat abans de fer la crida a la funció anterior. Finalment ha d'imprimir el resultat amb el missatge "Les centenes del nombre XX són CC" on XX és el nombre introduït per l'usuari i CC són les centenes d'aquest nombre.

Problema 8

Escriure una funció que retorni un nombre enter llegit per teclat, que ha d'estar comprès entre dos límits (inferior i superior) que es passen com a paràmetres. La funció ha de demanar nombres per teclat fins que se n'introdueixi un que estigui dins dels límits.

El programa principal `main()` ha de demanar a l'usuari que introdueixi els valors dels dos límits i cridi a la funció implementada anteriorment. La funció retornarà un valor entre els dos límits que el programa principal mostrarà per pantalla amb un missatge: "Valor entre II i SS: XX" on II i SS són els límits inferior i superior entrats per l'usuari, i XX és el valor entre els límits retornat per la funció.

Problema 9

Fer una funció que retorni el resultat d'eleva un nombre real x a una determinada potència n que haurà de ser un nombre enter. La funció ha de rebre els valors de x i n com a paràmetres i ha de calcular x^n utilitzant només multiplicacions. Tingueu en compte que l'exponent n pot ser negatiu.

Aquesta funció s'ha de cridar des d'un programa principal, que primer demanarà a l'usuari que introdueixi els valors del nombre x i de l'exponent n , farà la crida a la funció, i mostrarà per pantalla el resultat amb el missatge "El resultat d'eleva XX a la potencia NN es YY", on XX i NN seran el nombre x i l'exponent n introduïts per l'usuari, i YY serà el valor de x^n .

Problema 10

Fer una funció que permeti introduir un màxim de N números per teclat i comprovi si són primers o no. El valor de N se li passarà a la funció com a paràmetre. L'entrada de números s'ha de parar o bé quan es detecti un número primer o bé quan s'hagin introduït els 5 números. La funció retornarà un 1 si s'ha introduït algun número primer o un 0 en cas contrari.

A continuació, fer un programa principal que demani a l'usuari que introdueixi per teclat el valor de N i a continuació faci la crida a la funció. El resultat de la funció s'ha de mostrar amb el missatge "La funció ha detectar un nombre primer" si la funció retorna un 1 o bé "La funció no ha detectat cap nombre primer" si la funció ha retornar un 0.

Problema 11

Definiu una funció per calcular la divisió (només el quocient) entre dos nombres enters, utilitzant només les funcions $+$ (suma) i $-$ (resta). Heu de tenir en compte el signe dels operands.

Feu també el programa principal que llegeixi dos nombres enters des de teclat (dividend i divisor). A continuació si el divisor és diferent de zero, el programa cridarà a la funció i mostrarà el quocient per pantalla amb un missatge "XX : YY = QQ" on XX és el dividend, YY és el divisor i QQ és el quocient calculat per la funció. Si el divisor és zero, el programa ha de mostrar un missatge d'error: "Error: Divisio per zero".

Nota: Per a fer aquest problema us podria ser d'utilitat la funció `abs(X)` que calcula el valor absolut del nombre X . Per a utilitzar aquesta funció heu d'incloure la llibreria `math.h` a Visual Studio i `stdlib.h` al VPL de Caronte.

Problema 12

Fer una funció que rebí un valor enter N com a paràmetre i comprovi si N és una potència exacta de 2 o no (per exemple, 32 sí que ho és, però 24 no ho és). Si el valor és una potència exacta de 2 la funció retornarà 1 i en cas contrari retornarà -1.

A continuació, fer un programa principal que demani a l'usuari que introdueixi el valor de N i cridi a la funció per a comprovar si és potència exacta de dos o no. Si la funció dóna com a resultat que és potència exacta de 2, s'ha de mostrar per pantalla el missatge

"NN es potencia exacta de 2", on NN és el nombre introduït per l'usuari. En cas contrari, s'ha de mostrar el missatge "NN no es potencia exacta de 2".

Problema 13

Fer una funció que permeti calcular el terme n de la sèrie de Fibonacci, definida per:

$$a_0 = 0$$

$$a_1 = 1$$

$$a_n = a_{n-1} + a_{n-2}$$

La funció rebrà el valor de n com a paràmetre i retornarà el valor del terme a_n .

A continuació, fer el programa principal que demani a l'usuari quin terme de la sèrie de Fibonacci vol calcular. Si introdueix un terme negatiu, s'ha de mostrar el missatge "Error: El nombre no pot ser negatiu" i s'ha de tornar a demanar a l'usuari que introdueixi el terme a calcular. Aquest procés s'ha de repetir mentre el nombre introduït sigui negatiu. Un cop introdueixi un valor correcte, el programa ha d'utilitzar la funció anterior per calcular el terme corresponent que es mostrarà per pantalla amb un missatge amb format: "El terme NN de la serie de Fibonacci es XX" on NN és el nombre n entrat per l'usuari i XX és el valor del terme a_n calculat per la funció.

Problema 14

Fer una funció que rebi com a paràmetre un valor N . La funció llegirà N valors positius per teclat i retornarà el recompte de quants valors superiors a 10 s'han introduït. Cada cop que s'introdueixi un valor es comprovarà que realment és un valor positiu i en cas de no ser-ho es mostrarà el missatge "Error: Valor negatiu" i es tornarà a demanar a l'usuari que entri el valor.

Aquesta funció s'utilitzarà en el programa principal on es demanarà a l'usuari que introdueixi el valor N i es cridarà a la funció. El nombre de valors superiors a 10 retornat per la funció es mostrarà per pantalla amb un missatge: "L'usuari ha introduït XX valors mes petits que 10" on XX serà el valor retornat per la funció.

Problema 15

Recupereu el codi de l'exercici de la calculadora multi-funció del tema anterior. Feu un procediment per a fer la impressió per pantalla del menú.

Des del programa principal, es cridarà aquesta funció per a imprimir el menú quan correspongui. La resta de funcionament de la calculadora serà exactament igual al que ja s'havia fet al tema anterior.

Problema 16

Fer un procediment que imprimeixi per pantalla tots els divisors d'un nombre donat que es passarà al procediment com a paràmetre. Els divisors del nombre es mostraran per pantalla en una mateixa línia, separats per un espai entre ells.

Fer també el programa principal que demani a l'usuari que introdueixi un valor enter i que cridi la funció implementada passant el valor introduït per l'usuari com a paràmetre.

Problema 17

Fer un procediment que rebi dos valors enters com a paràmetres i escrigui per pantalla totes les potències del primer número que són menors al segon número. Els valors mostrats per pantalla s'han de separar per un espai. Per exemple, si la funció rep 3 i 90 com a paràmetres, el procediment ha de mostrar per pantalla els valors: 3 9 27 81

El programa principal ha de demanar a l'usuari que introdueixi dos valors enters, N i L, i ha de cridar el procediment anterior per tal que es mostrin els números corresponents a les potències de N que són menors que el valor límit L. Cal comprovar que el valor de N sigui més gran que 1 i que el valor de L és més gran que N. Només si es compleixen aquestes dues condicions es farà la crida a la funció. En cas contrari es mostrarà un missatge d'error: "Error: Valors incorrectes".

Problema 18

Fer un procediment que donat un enter n diferent d'1, escrigui tots els enters compresos entre 1 i n que siguin múltiples de 3 o de 7 però no dels 2 a la vegada. S'ha de considerar el cas en què el nombre n és negatiu, i ha de funcionar igualment. Per exemple, si s'introdueix el número -10, el programa hauria de mostrar el següent resultat: -3, -6. -7 i -9.

Fer un programa que utilitzi aquesta funció. Primer s'ha de demanar a l'usuari que introdueixi un valor enter diferent de 1, 0 i -1. Cal comprovar que l'enter entrat per l'usuari és correcte i si no ho és, cal mostrar un missatge d'error i tornar a demanar el nombre. Aquest procés s'ha de repetir fins que l'usuari introdueixi un valor correcte. A continuació, el programa cridarà a la funció amb el valor proporcionat per l'usuari.

Problema 19

Fer una funció que s'anomeni `ObtenirNombreSecret` i que faci el mateix que la part en negreta del codi que hi ha a continuació. Modifiqueu el programa per tal que cridi a la funció que heu fet en lloc d'efectuar les instruccions que estan en negreta.

```
int main( )
{
    int x,y;

    y=5; /* nombre secret y que l'usuari desconeix */

    do
    {
        cout << "Introdueix un nombre entre 1 i 10: ";
        cin >> x;
    }while((x<1) || (x>10));

    if(x==y)
        cout << "Has encertat!!!" << endl << endl;

    return 0;
}
```

A continuació utilitzeu la funció feta al problema 2 per tal de generar un valor aleatori entre 1 i 10 i fer que el nombre secret y sigui aleatori i diferent a cada execució del programa.

Nota: Per tal que la funció rand no retorni sempre els mateixos valors, cal incloure la següent inicialització a la primera línia del programa principal:

```
srand( (unsigned)time(NULL) );
```

Aquesta instrucció utilitza l'hora actual de l'ordinador per crear una llavor per a la funció rand(). D'aquesta manera la seqüència de nombres aleatoris que retorna rand() no és sempre la mateixa. Observeu que en aquesta instrucció s'utilitza la funció time. Aquesta funció està definida a la llibreria time.h, i per tant caldrà incloure-la al vostre projecte.

Per últim, modifiqueu el joc de la següent forma:

- L'usuari podrà jugar tants cops com vulgui
- En cada nova partida podrà escollir el número d'oportunitats que pot tenir per endevinar el número secret.
- El nombre secret ja no serà sempre un nombre entre 1 i 10 sinó que pot estar entre 1 i k, on k és un nombre que es llegeix a cada inici de partida.

Problema 20

Escriure una funció que calculi el quadrat d'un valor. La funció tindrà dos paràmetres, els dos de tipus int. El primer serà el valor sobre el que volem calcular el quadrat i el segon serà el resultat. La funció comprovarà que el valor sigui més gran que -180 i més petit que 180 perquè el resultat no surti de rang. En el cas que es compleixi la condició calcularem el resultat i la funció retornarà un 0. En cas contrari no es calcularà l'operació i la funció retornarà 1.

Fer el programa principal que demani a l'usuari que introdueixi el valor sobre el que vol calcular el quadrat. A continuació el programa cridarà a la funció i mostrarà el resultat per pantalla. Si el quadrat s'ha pogut calcular, mostrarà el missatge "El resultat d'elevat NN al quadrat es RR" on NN serà el nombre introduït per l'usuari i RR el resultat calculat per la funció. Si el resultat no s'ha pogut calcular, es mostrarà el missatge "Error: Valor massa gran i no es pot calcular el quadrat".

Problema 21

Fer una funció que calculi la divisió (quocient i residu) entre dos nombres enters, utilitzant només les funcions + (suma) i - (resta). Heu de tenir en compte el signe dels operands. Si el divisor és zero, la funció ha de retornar 1 i, en cas contrari, retornar 0.

Utilitzeu la funció dins d'un programa que llegeixi dos nombres enters i doni el resultat (quocient i residu) de la divisió amb el missatge: "Resultat – Quocient: QQ i Residu: RR" on QQ i RR són els valors calculats per la funció. Si la funció ha retornat un valor d'error (1), el programa ha de donar un missatge d'error ("Error: Divisio per zero") indicant que no s'ha pogut calcular la divisió.

Nota: Per a fer aquest problema us podria ser d'utilitat la funció abs(X) que calcula el valor absolut del nombre X. Per a utilitzar aquesta funció heu d'incloure la llibreria math.h a Visual Studio i stdlib.h al VPL de Caronte.

Problema 22

Fer una funció que calculi l'àrea i perímetre d'una circumferència. Se li passa radi com a float i retorna l'àrea i perímetre també com a float. La funció haurà de comprovar que el valor del radi passat com a paràmetre sigui més gran que zero. Si és així, calcularà i retornarà l'àrea i el perímetre, i el valor de retorn de la funció serà 0. En cas contrari, no es farà cap càlcul i el valor de retorn de la funció serà 1.

Fer també el programa principal que demani a l'usuari el valor del radi de la circumferència i cridi a la funció anterior. A continuació, si la funció ha retornat l'àrea i el perímetre correctament, els mostrarà per pantalla. Si la funció no ha pogut fer el càlcul i ha retornat 1, es mostrarà el missatge d'error "Error: El radi ha de ser un valor positiu".

Problema 23

Fer una funció que llegeixi nombres enters positius i negatius introduïts pel teclat fins que s'introdueixi un zero o bé fins que s'hagin llegit 10 valors. La funció ha de calcular i retornar el mínim i el màxim (en aquest ordre) de tots els valors llegits. Si el primer nombre que s'introdueix és zero la funció no ha de fer res i retornarà un 1 com a valor de retorn. Si s'introdueixen un o més valors diferents de zero (es podrà calcular el màxim i el mínim) la funció retornarà un 0 com a valor de retorn.

A continuació fer el programa principal que utilitzi aquesta funció per a calcular el mínim i el màxim de la seqüència de nombres que introduirà l'usuari. Si la funció ha pogut calcular els valors mínims i màxims (l'usuari ha introduït alguns valors abans d'introduir el zero), aquests es mostraran per pantalla amb un missatge: "El mínim de la seqüència es XX i el màxim es YY" on XX i YY seran els valors calculats per la funció. Si la funció no ha pogut calcular el mínim i el màxim perquè l'usuari ha introduït només un zero, es mostrarà el missatge: "Error: seqüència buida"

Problema 24

Escriure un procediment anomenat "polars" al que li passem dos valors de tipus float que seran les coordenades x i y d'un punt i que retorni el mòdul i l'angle (en aquest ordre) del punt.

NOTA: mòdul = $\sqrt{x^2 + y^2}$ (utilitzar sqrt()) i angle = $\tan^{-1} \frac{y}{x}$ (utilitzar atan())

Fer el programa principal que demani les coordenades del punt (x,y), cridi a la funció, i mostri el resultat obtingut per pantalla.

Problema 25

Escriure un procediment que ens retorni la pendent (a) i el terme independent (b) d'una funció lineal $y = ax + b$ i on li passem quatre valors que representen dues coordenades de la recta (x1,x2,y1,y2).

NOTA: $a = \frac{y_2 - y_1}{x_2 - x_1}$ i $b = y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1$

Fer el programa principal que demani les coordenades de dos punts d'una recta (x1,x2,y1,y2), cridi a la funció, i mostri el resultat obtingut per pantalla.

Problema 26

Escriure una funció anomenada "equacio" per resoldre equacions quadràtiques:

$$ax^2 + bx + c = 0$$

La funció tindrà cinc paràmetres:

- Els tres primers per passar els coeficients a, b, c (seran enters)
- Els dos segons per retornar les dues possibles arrels (seran reals)

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Com a valor de retorn, la funció retornarà el nombre d'arrels:

- 0 en el cas que $b^2 - 4ac < 0$
- 1 en el cas que $b^2 - 4ac = 0$
- 2 en el cas que $b^2 - 4ac > 0$

NOTA

- Per calcular l'arrel quadrada utilitzeu: `float sqrt(float num)` de la llibreria `math.h`

Fer el programa principal que demani els valors de a, b i c, cridi a la funció, i mostri el resultat obtingut per pantalla.

Problema 27

Les funcions següents permeten calcular l'àrea d'un quadrat, d'un rectangle i d'un triangle:

```
int AreaQuadrat(float costat, float &area)
{
    if (costat>0)
    {
        area=costat*costat;
        return 0;
    }
    else
        return 1;
}

int AreaRectangle(float base, float altura, float &area)
{
    if ((base>0)&&(altura>0))
    {
        area=base*altura;
        return 0;
    }
    else
        return 1;
}

int AreaTriangle(float base, float altura, float &area)
{
    if ((base>0)&&(altura>0))
    {
        area=base*altura/2;
        return 0;
    }
    else
        return 1;
}
```


El procediment següent, mostra per pantalla un menú de selecció que permet triar entre diferents opcions per a calcular l'àrea de figures habituals:

```
void MenuSeleccio()
{
    cout << "----- MENU -----" << endl << endl;
    cout << "1.-Area quadrat" << endl;
    cout << "2.-Area rectangle" << endl;
    cout << "3.-Area triangle" << endl;
    cout << "4.-Sortir" << endl << endl;
    cout << "Tria una de les opcions: " << endl;
}
```

Utilitzant les funcions i el procediment anteriors, fer un programa principal que mostri a l'usuari el menú de selecció i li demani que triï una de les opcions (char). Depenent de l'opció triada, el programa ha de demanar a l'usuari les dades (base, altura, etc.) de la figura sobre la que es vol calcular l'àrea i es cridarà a la funció corresponent. Amb el valor que retorni la funció, s'ha de mostrar el missatge "Area: AA" on AA serà el valor de l'àrea de la figura seleccionada si s'ha pogut calcular, o bé "Error: Dimensions incorrectes" si no s'ha pogut calcular l'àrea. El procés s'haurà de repetir mentre l'usuari no triï l'opció de sortir.

Problema 28

Definir un procediment que permeti modificar la posició en la pantalla d'un punt (x,y) en una direcció donada, però dins d'un quadre limitat per les coordenades (x_min, x_max, y_min, y_max). Les coordenades del punt, la direcció de moviment i els límits del quadre s'han de passar com a paràmetres de tipus enter al procediment. La direcció de moviment la passarem com un enter: 1=moviment cap a la esquerra, 2=moviment cap a la dreta, 3=moviment cap a baix, 4=moviment cap a dalt. El procediment haurà de comprovar que el moviment a fer no situï el punt fora del quadre. En cas que això passi, no es modificaran les coordenades del punt.

Utilitzeu aquest procediment en un programa que inicialment demani a l'usuari els límits d'un quadre (x_min, x_max, y_min, y_max) i comprovi que els límits del quadre són correctes (x_min < x_max i y_min < y_max). Si no ho són, el programa anirà demanant els límits fins que siguin correctes.

A continuació, el programa demanarà a l'usuari que introdueixi les coordenades d'un punt (x,y) dins del quadre. Caldrà comprovar que el punt és dins dels límits del quadre i si no hi és, el programa ha d'anar demanant les coordenades del punt fins que siguin correctes.

Després, el programa llegirà la direcció del moviment (sense mostrar cap missatge) i, només si és un dels valors vàlids (1, 2, 3, 4) cridarà al procediment anterior per a calcular la nova posició del punt. En retornar del procediment, imprimirà els nous valors de les coordenades del punt en format "(X,Y)". El programa ha d'anar llegint direccions de moviment, fent crides al procediment i mostrant el valor del punt fins que l'usuari introdueixi un valor de 5 com a direcció de moviment.

Nota: Considereu que la pantalla té l'origen de coordenades (0,0) a la cantonada superior esquerra.

Problema 29. Donat aquest algorisme:

```
int Factorial(int n)
{
    int total;
    total=1;
    while (n>1)
    {
        total=total*n;
        n--;
    }
    return total;
}

int Suma(int n)
{
    int total;
    total=0;
    while (n>=1)
    {
        total=total+n;
        n--;
    }
    return total;
}

void Calcular (int n, int &fac, int &sum)
{
    fac=Factorial(n);
    sum=Suma(n);
}

int main ()
{
    int n,fac,sum;

    cin >> n;
    Calcular(n,fac,sum);
    cout << n << fac << sum;

    return 0;
}
```

Indicar el resultat que escriu per pantalla l'algorisme principal si l'usuari introdueix un valor de n=5.

Indicar el resultat que escriuria per pantalla l'algorisme principal si féssim els canvis següents (suposant n=5):

1. Canviar la definició del procediment *Calcular* per:
void Calcular (int n, int fac, int sum)
2. Canviar la definició de la funció *Factorial* per:
int Factorial (**int** &n)