Exercicis de programació en C

12 de febrer de 2014

1.- Escriu un programa que pregunti dos nombres de tipus `int`. El, programa ha de dir per pantalla si el primer número és divisible pel segon.

2.- Escriu un programa que pregunti per pantalla els paràmetres d'un cilindre en format `double`: el radi r de la base i l'alçada h. El programa ha de treure per pantalla el volum del cilindre $\pi \cdot r^2 \cdot h$.

3.- Fes un programa que calculi les arrels d'una equació de segon grau *tant si són reals com complexes*.

4.- Escriu un programa que pregunti dos nombres de tipus `int` `n1` i `n2` (no necessàriament ordenats). El programa ha de treure per pantalla la suma de tots els nombres enters que hi ha entre `n1` i `n2` (incloent `n1` i `n2`).

5.- Escriu un programa que accepti com a únic paràmetre de línia de comandes un nombre real r que el suposem `double`. El programa ha d'escriure per pantalla en tres línies separades:

- el signe s d'r ($+1$ o -1).
- la part entera n del valor absolut d'r.
- la part decimal d del valor absolut d'r.

Per a fixar idees noteu que $r = s \cdot (n + d)$.

6.- Fes un programa que escrigui per pantalla les taules de multiplicar. Es recomana escriure una funció

```
void taula(int n)
```

que escrigui per pantalla la taula de multiplicar del número n.

7.- ***El programa càstig.*** Escriu un programa que accepti dos paràmetres de línia de comandes. El primer ha de ser un enter positiu `r`. El segon ha de ser una cadena de caràcters. El programa ha d'escriure r vegades a la pantalla el text entrat com a segon paràmetre.

8.- ***El programa de la conversió de divises.*** Volem escriure un programa perquè un banc pugui fer la conversió de les divises. El programa treballa en Euros (moneda 1); Dòlars USA (moneda 2); Lliures Esterlines (moneda 3); Yens (moneda 4) i Dòlars Australians (moneda 5). El programa ha de preguntar un import (`float`); en quina moneda està (1, 2, 3, 4 ò 5) i, finalment, a quina moneda es vol convertir (1, 2, 3, 4 ò 5 — diferent de l'anterior). Al acabar el programa escriu per pantalla l'import en la segona moneda (arrodonit a 2 decimals). Per exemple:

```
2.37 unitats de la moneda 2 valen 7.34 unitats de la moneda 4.
```

Idea: useu una matriu `float C[5][5]` que heu d'inicialitzar de manera que el lloc i, j de la matriu C (`C[i][j]`) conté el nombre d'unitats de la moneda j que ens paguen per una unitat de la moneda i. Òbviament: $C[j][i] = 1/C[i][j]$.

9.- ***Modificació de l'anterior (practiquem strings).*** Afegeix al programa anterior un vector de cadenes de caràcters (és a dir una matriu de caràcters) que contingui *els noms* de les divises. El programa, en lloc d'escriure el text anterior ha d'escriure alguna cosa com:

```
2.37 Dòlars USA valen 7.34 Yens.
```

10.- Escriu un programa que tregui per pantalla tots els caràcters imprimibles de la taula ASCII estàndard (es a dir del ' ' — codi ASCII 32 al ' ' — codi ASCII 126) junt amb el seu codi decimal equivalent segons la taula ASCII.

11.- El programa `factorial.c` que es llista a continuació calcula $n!$ de manera recursiva (el nombre n entra com a paràmetre del programa).

```
#include <stdio.h>
#include <stdlib.h>

unsigned int factpas (unsigned int);

int main (int argc, char *argv[])
{

  if (argc != 2)
    {
      printf ("\nÚs: fact <num>\n\n");
      return 1;
    }

  printf ("%u\n", factpas (atoi (argv[1])));
  return 0;
}

unsigned int factpas (unsigned int n)
{
  if (n < 2) return 1;
  return n * factpas (n - 1);
}
```

Modifica la funció `factpas` del programa anterior de manera que el calcul de $n!$ no sigui recursiu (cal usar un `for`).

12.- Fes un programa que escrigui per pantalla la taula de la funció logaritme neperià per valors d'x entre 1 i 10; en intervals de 10^{-3}. El resultat ha d'estar organitzat en dues columnes. La primera conté els valors d'x descrits anteriorment i la segona el valor del $\log(x)$ corresponent.

13.- Fes un programa que pregunti dos números en format `double` a i b. El programa ha d'escriure per pantalla la taula de la funció

$$\frac{a \cdot x + 1}{b \cdot x + a \cdot x^2 + 1}$$

per valors d'x entre -1 i 1; en intervals de 10^{-3}. Com en l'exercici anterior, el resultat ha d'estar organitzat en dues columnes. La primera conté els valors d'x descrits anteriorment i la segona el valor de la funció en el punt x. Es obligatori escriure i usar una funció que, donat x, calculi i retorni el valor de la funció en el punt.

14.- Es sap que la temperatura C en graus Celsius s'obté a partir de la temperatura F en graus Fahrenheit de la manera següent:

$$C = \frac{5}{9}(F - 32).$$

Fes una funció que converteixi temperatures en graus Fahrenheit a temperatures en graus Celsius i usa-la en un programa que escrigui una taula de conversió de temperatures en graus Fahrenheit a temperatures en graus Celsius. La taula ha d'anar des de 0°`F` fins a 300°`F`; en intervals de 20°`F`.

15.- Escriu un programa que accepti una data en forma de tres paràmetres de línia de comandes (dia, mes i any). Es suposa que la data és posterior a l'1/1/1970. El programa ha d'escriure per pantalla el dia de la setmana que correspon a la data entrada.
Nota: Els anys de traspàs son els divisibles per 4 (Exemple: 1970, 1974, ..., 1994, 1998, ...)

però que no siguin centenes (Exemple: l'any 2000 *no* és de traspàs encara que sigui divisible per 4). L'1/1/1970 era dijous.

16.- Escriu un programa que accepti com a únic paràmetre el nom d'un fitxer. Es suposa que el fitxer conté una única columna de dades. El programa ha d'obrir el fitxer amb totes les comprovacions necessàries, n'ha de llegir totes les dades i, seguidament, ha d'escriure per pantalla la seva mitjana i la seva variança.

17.- Escriu un programa que llegeixi caràcter a caràcter del "standard input" i escrigui al "standard output" fins que trobi la marca de final de fitxer. L'objectiu del programa és el d'escriure *exactament* el que rep però fent visibles els tabuladors i els salts de línia. Per això, cada vegada que el programa trobi un tabulador, en lloc seu, haurà d'escriure la cadena `"\t"` i cada vegada que el programa trobi un salt de línia, en lloc seu, haurà d'escriure la cadena `"\n"`.

18.- Escriu un programa que llegeixi caràcter a caràcter del "standard input" i escrigui al "standard output" fins que trobi la marca de final de fitxer. El programa ha de convertir totes les majúscules a la minúscula corresponent i, inversament, totes les minúscules a la majúscula corresponent.

19.- Escriu un programa que accepti un únic paràmetre que serà una cadena de caràcters. El programa ha d'escriure per l'"standard output" la cadena invertida. **Exemple:** Si s'entra "Bondia" el programa ha de treure "aidnoB".

20.- Escriu un programa que llegeixi caràcter a caràcter del "standard input" fins que trobi la marca de final de fitxer. Al final, el programa ha d'escriure el nombre de caràcters i línies que ha llegit (de manera similar a com ho fa el programa `wc` del Linux). **Nota:** Recorda que la marca de final (i, per tant, de separació) de línia és el caràcter `'\n'`.

21.- Modifica el programa anterior per tal que, a més, escrigui el nombre de paraules llegit. **Avís:** El separador de paraules no té perquè ser necessàriament un únic espai en blanc. Pot ser qualsevol combinació arbitrària d'espais en blanc i caràcters `'\t'` i `'\n'`.

22.- El Linux te la instrucció `seq` que funciona com segueix:

- `seq n1 n2` escriu per l'"standard output" tots els nombres enters que van de `n1` a `n2` (inclosos) d'un en un. **Exemple:** `seq 1 10` escriu `1 2 3 4 5 6 7 8 9 10` (cada nombre en una línia diferent).
- `seq n1 interv n2` escriu tots els nombres enters que van de `n1` a `n2` (inclòs `n1`) fent salts de mida `interv`. Nota que en aquest cas pot ser que `n2` no s'escrigui. **Exemple:** `seq 1 4 10` escriu `1 5 9` (cada nombre en una línia diferent).

Fes el programa que implementi la comanda `seq`. Nota que el programa ha d'acceptar *solament* 2 o 3 paràmetres. Si té dos paràmetres la comanda funciona com en el primer cas; si en té 3 com en segon.

23.- En aquest exercici farem un programa que comença de manera molt similar al programa final del document *Programació en C — transparències de les classes*; que pots usar de model.
Inici del programa: A la part de lectura de paràmetres i manipulació de fitxers, el programa ha de:

(a) Acceptar dos paràmetres de línia de comandes. El primer ha de ser un enter positiu que cal guardar a la variable `ncols`. El segon ha de ser el nom d'un fitxer de dades, que suposarem que té exactament `ncols` columnes.

(b) Obrir el fitxer de dades (especificat en el segon paràmetre) per a lectura.

(c) Obrir un fitxer per escriptura. El nom d'aquest segon fitxer ha de ser el mateix que el del fitxer de lectura amb `.est` afegit al final (**exemple:** si el fitxer de lectura es diu `dades.dat`, el de sortida s'ha de dir `dades.dat.est`).

El programa ha de llegir el fitxer de dades d'entrada; calcular la suma de totes les columnes (sabem que n'hi ha `ncols`) de cada fila d'aquest fitxer i escriure el resultat al fitxer de sortida (cada suma en una línia diferent). Així el fitxer de sortida tindrà el mateix nombre de línies que el d'entrada però solament una columna.

Exemple: Si el fitxer d'entrada és:

```
1 2 3 4
5 6 7 8
9 10 11 12
```

el fitxer de sortida ha de ser:

```
10
26
42
```

Millores a realitzar:

(1) Al final del fitxer de sortida, en línies diferents, escriure el nombre de línies processades i la mitjana i la variança de les sumes de les files.

(2) Calcular i escriure al fitxer de resultats el màxim i el mínim de les sumes de les files.

(3) Eliminar com a paràmetre de la línia de comandes l'especificació del nombre de columnes `ncols` del fitxer d'entrada. Així el programa accepta un únic paràmetre que és el nom del fitxer de lectura. El nombre de columnes del fitxer d'entrada (imprescindible per a l'execució del programa) cal calcular-lo al començar el programa, analitzant el nombre de camps (paraules) que hi ha a la primera línia (és a dir fins el primer caràcter `'\n'`). Per a obtenir aquest número podeu usar les tècniques de l'Exercici 21. Seguidament cal fer un `rewind` per a tornar a principi de fitxer i (com en els apartats anteriors) iniciar la lectura normal del fitxer de dades.

24.- (a) Escriure una funció que tingui per capçalera:

```
double evalpoli(int g, double * coef, double x)
```

On: `g` és el grau del polinomi, que és més petit o igual que 25; `coef` és un vector que conté els $g+1$ coeficients del polinomi amb el conveni que `coef[i]` és el coeficient de x^i. És a dir, el polinomi el pensem com

```
coef[0] + coef[1] * x + coef[2]*x^2 + ... + coef[g] * x^g
```

`x` és un nombre real. La funció ha de tornar el valor del polinomi avaluat al punt x. Si és possible això s'hauria de fer usar la Regla de Horner.

(b) Escriure un programa que pregunti el grau $g \leq 25$ (`int`) d'un polinomi $p(x)$. Seguidament el programa ha de preguntar els coeficients del polinomi $p(x)$ i els ha de guardar a un vector (`double`). Llavors, el programa ha de calcular el nombre R que és el màxim del valor absolut de tots els coeficients. Finalment el programa ha d'usar la funció feta a l'apartat anterior per a tabular el polinomi $p(x)$ per a tots els valors d'x entre $-R$ i R; en intervals de 10^{-2}. Com en exercicis anteriors, el resultat ha d'estar organitzat en dues columnes. La primera conté els valors d'x descrits anteriorment i la segona el valor del polinomi $p(x)$ en el punt x.

25.- Reescriu el programa de l'Exercici 5 usant una funció que faci la descomposició d'un nombre real x. Aquesta funció té per capçalera:

```
void descomp(double x, int *s, int *n, double *d)
```

El paràmetre `x` és d'entrada i és el nombre a descomposar. Els paràmetres `s` (signe), `n` (part entera del valor absolut d'`x`) i `d` (part decimal del valor absolut d'`x`) són de sortida. Ara el programa ha de cridar la funció `descomp` passant-li `x` i recollint `s`, `n` i `d`; i escriure el resultat.

26.- Usa la funció `descomp` de l'exercici anterior per a modificar el programa de l'Exercici 24 de manera que, ara, el programa per a cada x solament llisti el *signe* de $p(x)$ en lloc del valor real de $p(x)$.

27.- ***Operacions de matrius.*** Escriu un programa que pregunti una matriu `float` A, de dimensió 3×3, i escrigui per pantalla les matrius $A, A*A, A*A*A, \ldots, A^{10}$. Per això cal:

(a) Declarar la matriu i preguntar la seva dimensió i els seus valors per pantalla. Si es vol que el programa sigui el mes general possible es recomana fer la declaració:

```
#define DIM 10

int main ()
```

```
{
  float mat[DIM][DIM];
```

Respecte de l'entrada de valors per pantalla pots trobar un model al final de la pàgina 29 del manual *Programació en C per a matemàtics*, de Lluís Ribas Xirgo).

(b) Escriure les següents funcions auxiliars:

(i) Una funció que escrigui una matriu per pantalla. La seva capçalera podria ser

```
void EscriuMatriu (float matriu[DIM][DIM], int dimensio)
```

i el seu prototipus:

```
void EscriuMatriu (float [DIM][DIM], int);
```

(ii) Una que dupliqui una matriu `input` a una `output`. Capçalera:

```
void CopiaMatriu(float input[DIM][DIM], float output[DIM][DIM],
                 int dimensio)
```

Prototipus:

```
void CopiaMatriu (float[DIM][DIM], float[DIM][DIM], int);
```

(iii) Una funció que faci el producte de la matriu `A` per la matriu `B` i guardi el resultat a la matriu `Res`. Capçalera:

```
void MultiplicaMatrius (float A[DIM][DIM], float B[DIM][DIM],
                        float Res[DIM][DIM], int dimensio)
```

Prototipus:

```
void MultiplicaMatrius (float[DIM][DIM], float[DIM][DIM],
                        float[DIM][DIM], int);
```

(c) Programa un "bucle" `for` que vagi calculant les potències de la matriu i les escrigui.

28.- ***Difícil:*** Escriu un programa que pregunti una matriu `double` A, de dimensió $n \times n$ amb $n \leq 10$ i la triangularitzi pel mètode de Gauss. Si en algun moment es troba un pivot zero es para el programa i es dona un missatge d'error. Finalment el programa ha d'escriure la matriu triangular final i el seu determinant per pantalla.