

Examen final de Programació 1

Grau en Enginyeria Informàtica

12 de gener de 2018

Nom i cognoms:

NIF:

Problema 1 (1,5 punts) Responen al full de l'enunciat.

Donat el programa en C++ següent:

```
#include<stdio.h>
// Prototipus accions auxiliars
void A(int b, int &a);
void B(int &i, int &j);

int main(){
    int a = 1, b = 0;
    A(a, b);
    printf("Main: %i    %i \n",a , b);
}
// Implementació accions auxiliars
void A(int b, int &a){
    int c;
    b = 5;
    c = a;
    a = b;
    B(b, a);
    printf("Acció A: %i    %i    %i \n", a, b, c);
}
void B(int &i, int &j){
    int c = 4;
    i = j + 2;
    j = c + 2;
    printf("Acció B: %i    %i    %i \n", i, j, c);
}
```

Determineu quins valors mostra per pantalla l'execució del programa.

Problema 2 (1,5 punts) Responen al full de l'enunciat.

El programa en C++ següent conté errors a les capçaleres de les accions.

```
#define N 3
#include<stdio.h>
// Prototipus accions auxiliars
void copiar_valor(float &f, float x[N]);
void intercanvi(float t1, float t2);
void intercanviar_valor(float f1, float f2);
void mostrar(float t);

int main(){
    float t1[N], t2[N];
    int i;
    for (i=0; i<N; i++){
        copiar_valor(3.2 , t1[i]);
        copiar_valor(5.7 , t2[i]);
    }
    intercanvi(t1, t2);
    mostrar(t1);
    mostrar(t2);
} // final de main

// Implementació accions auxiliars
void copiar_valor(float &f, float x[N]){
    x = f;
}
void intercanvi(float t1, float t2){
    int i;
    for (i=0; i<N; i++){
        intercanviar_valor(t1[i], t2[i]);
    }
}
void intercanviar_valor(float f1, float f2){
    float v;
    v = f1;
    f1 = f2;
    f2 = v;
}
void mostrar(float t){
    int i;
    for (i=0; i<N; i++) {
        printf("%G\t", t[i]);
    }
    putchar('\n');
}
```

Considerant que totes les crides a les accions són correctes elimineu tots els errors de “casament” de paràmetres. **Com a solució al problema únicament heu d’indicar clarament la nova capçalera de cada acció.**

L’execució del programa corregit mostra per pantalla els valors següents:

```
5.7  5.7  5.7
3.2  3.2  3.2
```

Problema 3 (3 punts) **Implementeu** en el llenguatge de programació ANSI C/C++ una funció que anomenareu **diagonal** que determini si una matriu quadrada de valors enters és simètrica respecte de la diagonal secundària.

La funció **diagonal** tindrà el prototipus següent:

```
#define N 4
bool diagonal(int m[N][N]);
```

La funció **diagonal** rebrà com a entrada la matriu m de $N \times N$ valors enters i retornarà **true**, si m és simètrica respecte de la diagonal secundària, i **false**, en un altre cas.

Per exemple, si la matriu d'entrada m és la següent:

0	1	2	3
4	5	6	2
7	8	5	1
9	7	4	0

la funció **diagonal** retornarà **true** ja que la matriu és simètrica respecte de la diagonal secundària (posicions (0,3), (1,2), (2,1) i (3,0) a la matriu de 4×4). **Feu un disseny que no depengui de la mida de la matriu quadrada**, és a dir, que no depengui de si N val 4 o 100.

Problema 4 (4 punts) **Implementeu** en el llenguatge de programació ANSI C/C++, **utilitzant la tècnica del disseny descendent**, una acció que anomenareu **digit** que esbrini quin és el dígit que més es repeteix en una seqüència de caràcters.

L'acció **digit** tindrà el prototipus següent:

```
#define M 30
void digit(char s[M+1], char &d, int &r);
```

L'acció **digit** tindrà un paràmetre d'entrada:

- s : string de mida màxima M marcat amb el caràcter NUL '\0', és a dir, rebrà com a entrada una seqüència de caràcters marcada amb el caràcter NUL '\0'.

i dos paràmetres de sortida:

- d : el dígit (símbol '0'...'9') més freqüent a s (dígit que més es repeteix en s).
- r : el nombre de repeticions en s del dígit més freqüent d .

Per exemple, si la seqüència d'entrada s és la següent:

'a'	'7'	'3'	'0'	'7'	'B'	'7'	'2'	'1'	'3'	'\0'	...
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-----

$d = '7'$ i $r = 3$ ja que el dígit amb més repeticions és el '7' i es repeteix 3 cops.

Consideracions:

- En cas que la seqüència d'entrada no contingui cap dígit, fixarem $d = '0'$ i $r = 0$.
- En cas que el dígit més freqüent no sigui únic, agafarem el que denota el valor enter més baix. Per exemple, si la seqüència d'entrada s és la següent:

'a'	'3'	'3'	'4'	'4'	'B'	'c'	'+'	'x'	'1'	'2'	'\0'	...
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	-----

veiem que els díigits més freqüents són el '3' i el '4', ambdós amb 2 repeticions. Per tant, agafarem el més baix: $d = '3'$ i $r = 2$

- Per implementar la solució podeu fer ús de les funcions definides a la llibreria estàndard `<string.h>` i `<ctype.h>`.
- Intenteu fer un codi clar, entenedor i ben estructurat. Si creieu que pot ajudar, fiquen algun comentari.