

NOM	DATA	8 / 6 / 2016	QUALIFICACIÓ
ÀREA/MATÈRIA	ASIX - M03 - UF2	CURS	
		2015 - 2016	

Feu captures de pantalla per a documentar l'examen.

RA1-1)(2 punts) Emprant **gdb** poseu un punt de ruptura a la línia de **bCondicio = (fY>fX*fX);** dins del bucle **for** de la funció **main** de **m4e6_07.c** (codi adjunt a aquest examen) per a saber el valor de **fX** i **fY** a les dues primeres iteracions. Captureu el tros de pantalla que presenta el valor de **fX** i **fY** a les dues primeres iteracions.

RA1-2)(2 punts) Dividiu **m4e6_07.c** en 3 arxius : **m4e6Main.c** (a on hi va la funció principal), **m4e6Func.c** (implementació de les funcions que no són la funció principal) i **m4e6Func.h** (declaració de prototipus de les funcions que no són la funció principal). Documenteu com feu la compilació.

RA1-3)(2 punts) Desenvolpeu el programa **nomCat.c** que presenta un nombre de Catalan (del matemàtic belga Eugène Charles Catalan). Ha de contenir la funció recursiva **nNombreDeCatalan(int nNum)** que calcula el nombre de Catalan d'un número. Per a desenvolupar la funció recursiva heu de tenir en compte :

$$C_n = \begin{cases} \text{si } n = 0 & \Rightarrow 1 \\ \text{si } n > 0 & \Rightarrow \frac{2(2n-1)}{n+1} C_{n-1} \end{cases}$$

* Quan nNum és igual a 0:
`nNombreDeCatalan(nNum) = 1`

* Quan nNum és més gran que 0:
`nNombreDeCatalan(nNum) = (2*(2*nNum-1)/(nNum+1))*nNombreDeCatalan(nNum-1)`

Verifiqueu el seu funcionament :

```
ecat@debian8:~/Documents/dawl/m3/20160429ex/uf2$ ./nomCat
nNombreDeCatalan(0) = 1
nNombreDeCatalan(1) = 1
nNombreDeCatalan(2) = 2
nNombreDeCatalan(3) = 5
nNombreDeCatalan(4) = 14
nNombreDeCatalan(5) = 42
nNombreDeCatalan(6) = 132
nNombreDeCatalan(7) = 429
nNombreDeCatalan(8) = 1430
nNombreDeCatalan(9) = 4862
nNombreDeCatalan(10) = 16796
nNombreDeCatalan(11) = 58786
nNombreDeCatalan(12) = 208012
```

```
int main(){
    int nN = 0;

    do{
        printf("nNombreDeCatalan(%d) \t= %d\n", nN, nNombreDeCatalan(nN));
    }while(nN++ < 12);

    return 0;
}
```

RA1-4)(1 punt) Desenvolpeu la funció **vMajAsterisc** que quan troba una majúscula la converteix a un asterisc (*). Es comporta d'aquesta manera:

```
ecat@debian8:~/Documents/dawl/m3/20160429ex/uf2$ ./majAsterisc00
szA: L'escola de Jesuïtes El Clot
szA: *'escola de *esuïtes *l *lot
szB: #JEClot
szB: #***lot
ecat@debian8:~/Documents/dawl/m3/20160429ex/uf2$ █

int main(){
    char szA[N] = "L'escola de Jesuïtes El Clot", szB[N] = "#JEClot";

    printf("szA: %s\n", szA);
    vMajAsterisc(szA);
    printf("szA: %s\n", szA);
    printf("szB: %s\n", szB);
    vMajAsterisc(szB);
    printf("szB: %s\n", szB);

    return 0;
}
```

RA1-5)(1 punt) Com a evolució de la pregunta anterior, desenvolupau la funció **szMajAsterisc** que es comporta d'aquesta manera:

```
ecat@debian8:~/Documents/daw1/m3/20160429ex/uf2$ ./majAsterisc01
szA: L'escola de Jesuïtes El Clot
szA: *'escola de *esuïtes *l *lot
szA: *'escola de *esuïtes *l *lot
szB: #JEClot
szB: #***lot
szB: #***lot
ecat@debian8:~/Documents/daw1/m3/20160429ex/uf2$ █
```

```
int main(){
    char szA[N] = "L'escola de Jesuïtes El Clot",szB[N] = "#JEClot";

    printf("szA: %s\n",szA);
    printf("szA: %s\n",szMajAsterisc(szA));
    printf("szA: %s\n",szA);
    printf("szB: %s\n",szB);
    printf("szB: %s\n",szMajAsterisc(szB));
    printf("szB: %s\n",szB);

    return 0;
}
```

Targetes de crèdit

L'algoritme per comprovar la validesa del nombre d'una targeta de crèdit és una barreja del dels codis de barres i de l'ISBN.



Una targeta de crèdit acostuma a tenir 16 xifres que designen l'entitat bancària, el compte i una darrera xifra de control.

Número de targeta
1234 5678 9876 5437

Per calcular aquest dígit es procedeix així:

- les xifres que ocupen un lloc senar es multipliquen per 2
- si el producte té dues xifres es redueixen a una sumant les del resultat obtingut ($8 \cdot 2 = 16 \Rightarrow 1+6=7$)
- les xifres que ocupen un lloc parell es multipliquen per 1 (no es fa res)
- es sumen tots els resultats
- es mira si la suma és divisible per 10
- si no ho és el número és incorrecte.

Observa les operacions que es fan per comprovar el número de targeta.

Número de targeta: 1234567898765437															
1	2	3	4	5	6	7	8	9	8	7	6	5	4	3	7
x2	x1	x2	x1	x2	x1	x2	x1	x2	x1	x2	x1	x2	x1	x2	x1
2	2	6	4	10	6	14	8	18	8	14	7	10	4	6	7
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
2	2	6	4	1	6	5	8	9	8	5	7	1	4	6	7
Divisió per 10															80
															80/10=8 (exacta)

// Funció principal de tc00.c, adjunt a aquest examen.

```
int main(){
    char szTargeta1[N]="1234567898765437",szPesos[N]="21212121212121";
    char szTargeta2[N]="4780258155837810",szTargeta3[N]="1600587239741473";

    vPresentaTargetaPerPesos(szTargeta1,szPesos);

    return 0;
}
```

RA1-6)(1 punt) Evolucioneu **tc00.c**, adjunt a aquest examen, com a **tc01.c** modificant-hi la funció **vPresentaTargetaPerPesos** per a poder veure aquesta execució de programa:

```
ecat@debian8:~/Documents/daw1/m3/20160429ex/uf2$ ./tc01
Targeta:
1      2      3      4      5      6      7      8      9      8      7      6      5      4      3      7
Pesos:
2      1      2      1      2      1      2      1      2      1      2      1      2      1      2      1
Multiplicat i reduït:
2      2      6      4      1      6      5      8      9      8      5      6      1      4      6      7
ecat@debian8:~/Documents/daw1/m3/20160429ex/uf2$ █
```

Feu servir la funció **int nSuma2xifres(int nNum)**:

```
int nSuma2xifres(int nNum){
    return nNum % 10 + nNum/10;
}
```

RA1-7)(1 punt) Evolucioneu **tc01.c** com a **tc02.c** afegint-hi la funció **nSumaTargetaPerPesos** per a poder veure aquesta execució de programa:

```
ecat@debian8:~/Documents/daw1/m3/20160429ex/uf2$ ./tc02
La targeta 1234567898765437 és vàlida
La targeta 4780258155837810 és vàlida
La targeta 1600587239741473 no és vàlida
```

```
int main(){
    char szTargeta1[N]="1234567898765437",szPesos[N]="21212121212121";
    char szTargeta2[N]="4780258155837810",szTargeta3[N]="1600587239741473";

    //vPresentaTarjetaPerPesos(szTargeta1,szPesos);
    if(nSumaTarjetaPerPesos(szTargeta1,szPesos) % 10)
        printf("La targeta %s no és vàlida\n",szTargeta1);
    else
        printf("La targeta %s és vàlida\n",szTargeta1);
    if(nSumaTarjetaPerPesos(szTargeta2,szPesos) % 10)
        printf("La targeta %s no és vàlida\n",szTargeta2);
    else
        printf("La targeta %s és vàlida\n",szTargeta2);

    printf("La targeta %s %s vàlida\n",
           szTargeta3,
           (nSumaTarjetaPerPesos(szTargeta3,szPesos) % 10)?"no és":"és");

    return 0;
}
```

Molta sort a totes i tots !!!!