

NOM	DATA	23 / 04 / 2018	QUALIFICACIÓ
ÀREA/MATÈRIA	DAW - M03 - UF2	CURS	
		2017 - 2018	

Feu captures de pantalla per a documentar l'examen.

RA1-1)(2 punts) Emprant **gdb** poseu un punt de ruptura a la línia 33 al codi **rps01.c** (joc del pedra -**R** de rock-, paper -**P**- i tisora -**S** de scissor-) adjunt a aquest examen. Demostreu com sempre podeu guanyar a l'ordinador emprant **gdb**.

RA1-2)(2 punts) Dividiu **rps01.c** en tres arxius: **rpsMain.c** (a on hi va la funció principal), **rpsLogica.c** (implementació de les funcions que no són **main**) i **rpsLogica.h** (declaració de prototipus de les funcions que no són **main**). Documenteu com feu la compilació.

RA1-3)(1 punt) Desenvolpeu la funció **cMajusculaAminuscula** per a que tota majúscula passada per valor la retorni com a minúscula, i en cas contrari retorni el mateix caràcter. Exemple de visualització :

```
$ ./ex02
abcdef 1234 xyz
```

```
int main(){
    char szCadena[N] = "aBcDeF 1234 xYz";
    int i;

    for(i = 0 ; szCadena[i] ; i++)
        printf("%c",cMajusculaAminuscula(szCadena[i]));
    printf("\n");

    return 0;
}
```

RA1-4)(1 punt) Desenvolpeu la funció **vMajusculesAminuscules**, fent servir **cMajusculaAminuscula**, per a que tota majúscula passada per una cadena per referència la converteixi a minúscula, i en cas contrari deixi el mateix caràcter. Exemple de visualització :

```
$ ./ex03
aBcDeF 1234 xYz
abcdef 1234 xyz
```

```
int main(){
    char szCadena[N] = "aBcDeF 1234 xYz";

    printf("%s\n",szCadena);
    vMajusculesAminuscules(szCadena);
    printf("%s\n",szCadena);

    return 0;
}
```

... la prova continua a la següent pàgina →

RA1-5)(1 punt) Desenvolpeu la funció **szMajusclesAminuscules**, fent servir **vMajusclesAminuscules**, per a que tota majúscula passada per una cadena per referència la converteixi a minúscula, i en cas contrari deixi el mateix caràcter. Exemple de visualització :

```
$ ./ex04
aBcDeF 1234 xYz
abcdef 1234 xyz

int main(){
    char szCadena[N] = "aBcDeF 1234 xYz";

    printf("%s\n",szCadena);
    printf("%s\n",szMajusclesAminuscules(szCadena));

    return 0;
}
```

RA1-6)(1 punt) Desenvolpeu la funció **bool bMajusculaAminuscula(char *cMin,char c)** per a que tota majúscula passada per valor la retorni com a minúscula per referència, i en cas contrari retorni el mateix caràcter per referència. Si ha fet la conversió retornarà **true** i en cas contrari **false**. Exemple de visualització :

```
$ ./ex05
Covertit B a b
Covertit D a d
Covertit F a f
Covertit Y a y

int main(){
    char szCadena[N] = "aBcDeF 1234 xYz",cConvertit;
    int i;

    for(i = 0 ; szCadena[i] ; i++)
        if(bMajusculaAminuscula(&cConvertit,szCadena[i]))
            printf("Covertit %c a %c\n",szCadena[i],cConvertit);

    return 0;
}
```

RA1-7)(2 punts) Desenvolpeu la funció **nQuantesMajusclesHiHa** per a que compti el número de majúscules d'una cadena passada per referència fent servir la funció **bMajusculaAminuscula**. Exemple de visualització :

```
$ ./ex06
A "aBcDeF 1234 xYz" hi ha 4 majúscules

int main(){
    char szCadena[N] = "aBcDeF 1234 xYz";

    printf("A \"%s\" hi ha %d majúscules\n",
        szCadena,nQuantesMajusclesHiHa(szCadena));

    return 0;
}
```

Molta sort a totes i tots !!!!