

NOM	DATA	25 / 04 / 2018	QUALIFICACIÓ
ÀREA/MATÈRIA	ASIX - 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 25 al codi **bubble04.c** adjunt a aquest examen. Mireu el valor d'on apunta **p** en les dues primeres iteracions emprant **gdb**.

RA1-2)(2 punts) Dividiu **bubble04.c** en tres arxius: **bubbleMain.c** (a on hi va la funció principal), **bubbleFunc.c** (implementació de les funcions que no són **main**) i **bubbleFunc.h** (declaració de prototipus de les funcions que no són **main**). Documenteu com feu la compilació.

RA1-3)(1 punt) Desenvolpeu la funció **nSuma** per a que retorni la suma del primer argument i el segon argument. En cas de que el número sumat sigui més gran que 9 heu de restar 10 a la suma. Exemple de visualització :

```
$ ./ex02
1234567890
4567890123

int main(){
    int nVector[N] = {1,2,3,4,5,6,7,8,9,0},i;

    vVisualitzaVector(nVector,N);
    for(i = 0 ; i < N ; i++)
        printf("%d",nSuma(nVector[i],3));
    printf("\n");

    return 0;
}
```

RA1-4)(1 punt) Desenvolpeu la funció **vSumaVector**, fent servir **nSuma**, per a que sumi a cada element del vector, passat per referència al primer argument, el valor del segon argument, essent el tercer argument la mida del vector. Exemple de visualització :

```
$ ./ex03
1234567890
4567890123

int main(){
    int nVector[N] = {1,2,3,4,5,6,7,8,9,0},i;

    vVisualitzaVector(nVector,N);
    vSumaVector(nVector,3,N);
    vVisualitzaVector(nVector,N);

    return 0;
}
```

RA1-5)(1 punt) Desenvolpeu la funció **pnSumaVector**, fent servir **vSumaVector**, per a que retorni l'adreça del vector. Exemple de visualització :

```
$ ./ex04
1234567890
4567890123

int main(){
    int nVector[N] = {1,2,3,4,5,6,7,8,9,0},i;

    vVisualitzaVector(nVector , N);
    vVisualitzaVector(pnSumaVector(nVector,3,N) , N);

    return 0;
}
```

... la prova continua a la següent pàgina →

RA1-6)(1 punt) Desenvolueu la funció **bool bSumaSuperiorA9(int *nV,int nQuantSumo)** per a sumar el valor del segon argument al valor apuntat per la referència del primer argument. En cas de que el número sumat sigui més gran que 9 heu de restar 10 a la suma. Si ha fet la resta retornarà **true** i en cas contrari **false**. Exemple de visualització :

```
$ ./ex05
1234567890
#define bool unsigned char
nVector[0] = 4 <-- La suma donava inferior a 9
nVector[1] = 5 <-- La suma donava inferior a 9
nVector[2] = 6 <-- La suma donava inferior a 9
nVector[3] = 7 <-- La suma donava inferior a 9
nVector[4] = 8 <-- La suma donava inferior a 9
nVector[5] = 9 <-- La suma donava inferior a 9
nVector[6] = 0 <-- La suma donava superior a 9
nVector[7] = 1 <-- La suma donava superior a 9
nVector[8] = 2 <-- La suma donava superior a 9
nVector[9] = 3 <-- La suma donava inferior a 9
4567890123

int main(){
    int nVector[N] = {1,2,3,4,5,6,7,8,9,0},i;

    vVisualitzaVector(nVector , N);
    for(i = 0 ; i < N ; i++)
        if(bSumaSuperiorA9(&nVector[i],3))
            printf("nVector[%d] = %d <-- La suma donava superior a 9\n",i,nVector[i]);
        else
            printf("nVector[%d] = %d <-- La suma donava inferior a 9\n",i,nVector[i]);
    vVisualitzaVector(nVector , N);

    return 0;
}
```

RA1-7)(2 punts) Desenvolueu la funció **nQuantesSumesSuperiorsA9** per a que compti el número de cops que la suma feta és superior a 9, fent servir la funció **bMajusculaAminuscula**. Exemple de visualització :

```
$ ./ex06
1234567890
Hi ha hagut 3 sumes superiors a 9.
4567890123

int main(){
    int nVector[N] = {1,2,3,4,5,6,7,8,9,0},i;

    vVisualitzaVector(nVector , N);
    printf("Hi ha hagut %d sumes superiors a 9.\n",
        nQuantesSumesSuperiorsA9(nVector, 3 , N));
    vVisualitzaVector(nVector , N);

    return 0;
}
```

Molta sort a totes i tots !!!!