

NOM	DATA	27 / 4 / 2021	QUALIFICACIÓ
ÀREA/MATÈRIA	DAW - M03 - UF2	CURS	
		2020 - 2021	

Feu captures de pantalla per a documentar l'examen. A l'acabar-ho, trameteu-ho a carles.olive@fje.edu i jordibinefa@fje.edu, adjuntant un arxiu en **pdf** demostrant els punts claus de les respostes i un arxiu **.zip** (o **.tar.gz**, **No s'accepten .7z o .rar**) amb els codis i **sense** els arxius **executables**. Aquest examen té més de 10 punts, escolliu les preguntes que considereu més adients.

RA1-1)(2 punts) Depuració

RA1-1-1)(0,3 punts) Compileu **ex01.c**, adjunt a aquest examen, per a poder ser depurat pel **gdb**. El nom de l'executable ha de ser **ex01**.

RA1-1-2)(0,5 punts) Emprant el **gdb**, visualitzeu el codi des de la línia **39** fins a la **46**.

RA1-1-3)(0,2 punts) Poseu un punt de ruptura a la línia **44** per a saber el valor de ***If** a cada iteració.

RA1-1-4)(1 punt) Visualitzeu el valor de ***If** a les **tres** primeres iteracions.

RA1-2)(1,5 punts) Dividiu **ex01.c** en cinc arxius: **ex02main.c** (a on hi va la funció principal), **ex02senseStdio.c** (implementació de les funcions que **no fan ús** de l'arxiu de capçalera **stdio.h**), **ex02senseStdio.h** (declaració de prototipus de les funcions que **no fan ús** de l'arxiu de capçalera **stdio.h**), **ex02ambStdio.c** (implementació de les funcions **que fan ús** de l'arxiu de capçalera **stdio.h**), **ex02ambStdio.h** (declaració de prototipus de les funcions **que fan ús** de l'arxiu de capçalera **stdio.h** i **els #define**). **Documenteu** com feu la compilació.

RA1-3-1)(1,5 punts) Desenvolpeu la funció **float fSumaRecursiva(float* pF,int nQuants)** que troba recursivament la suma dels elements del vector passat per referència i un enter que indica la quantitat d'elements a sumar.

Aproveiteu el codi **ex03previ.c** i deseu-lo com a **ex03.c**. Aquesta és la lògica de funcionament:

```
Si nQuants == 0 → fSumaRecursiva(float* pF,int nQuants) = 0.f
Si nQuants == 1 → fSumaRecursiva(float* pF,int nQuants) = pF[0]
Si nQuants > 1 → fSumaRecursiva(float* pF,int nQuants) = pF[nQuants-1] + fSumaRecursiva(pF,nQuants-1)
```

```
$ ./ex03
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma recursiva dels valors a fPesos és 1.00
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma recursiva dels valors a fValors és 31.35
La suma no recursiva dels valors a fValors és 31.35
```

RA1-3-2)(1,5 punts) Desenvolpeu la funció **fSumaRecursiva(IF,nQuants)** que troba recursivament la suma dels elements del vector passat per referència i un enter que indica la quantitat d'elements a sumar.

Aproveiteu el codi **ex04previ.py** i deseu-lo com a **ex04.py**. Aquesta és la lògica de funcionament:

```
Si nQuants == 0 → fSumaRecursiva(IF,nQuants) = 0
Si nQuants == 1 → fSumaRecursiva(IF,nQuants) = IF[0]
Si nQuants > 1 → fSumaRecursiva(IF,nQuants) = IF[nQuants-1] + fSumaRecursiva(IF,nQuants-1)
```

```
$ python3 ex04.py
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma recursiva dels valors a fPesos és 1.00
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma recursiva dels valors a fValors és 31.35
La suma no recursiva dels valors a fValors és 31.35
```

RA1-4-1)(1 punts) Desenvolpeu la funció **float fQuinValor(float* pF,int nQuin)** que retorna el valor de l'element **nQuin** del vector passat per referència. Aproveiteu el codi **ex05previ.c** i deseu-lo com a **ex05.c**

```
int main(){
    float fPesos[N] = {0.1f,0.15f,0.2f,0.25f,0.3f}, fValors[N] = {5.1f,4.2f,6.15f,7.4f,8.5f};
    float fSumaNoRecursiva = 0.f;
    int i;

    printf("Pesos : [ ");
    for(i = 0; i < N ;i++){
        printf("%.2f ",fQuinValor(fPesos,i));
        fSumaNoRecursiva += fQuinValor(fPesos,i);
    }
    printf("]\n");
    printf("La suma no recursiva dels valors a fPesos és %.2f\n",fSumaNoRecursiva);
    fSumaNoRecursiva = 0.f;
    printf("Valors: [ ");
    for(i = 0; i < N ;i++){
        printf("%.2f ",fQuinValor(fValors,i));
        fSumaNoRecursiva += fQuinValor(fValors,i);
    }
    printf("]\n");
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaNoRecursiva);

    return 0;
}
```

```
$ ./ex05
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
```

RA1-4-2)(1 punts) Desenvolpeu la funció **fQuinValor(IF,nQuin)** que retorna el valor de l'element **nQuin** de la llista **IF**. Aproveiteu el codi **ex06previ.py** i deseu-lo com a **ex06.py**

```
fPesos = [0.1,0.15,0.2,0.25,0.3]
fValors = [5.1,4.2,6.15,7.4,8.5]
fSumaNoRecursiva = 0.

print("Pesos : [ ",end='')
for i in range(len(fPesos)):
    print("%.2f "%fQuinValor(fPesos,i),end='')
    fSumaNoRecursiva += fQuinValor(fPesos,i)
print("]")
print("La suma no recursiva dels valors a fPesos és %.2f"%fSumaNoRecursiva)
fSumaNoRecursiva = 0.
print("Valors: [ ",end='')
for i in range(len(fValors)):
    print("%.2f "%fQuinValor(fValors,i),end='')
    fSumaNoRecursiva += fQuinValor(fValors,i)
print("]")
print("La suma no recursiva dels valors a fValors és %.2f"%fSumaNoRecursiva)
```

```
$ python3 ex06.py
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
```

RA1-5-1)(1 punts) Desenvolpeu la funció **float fSumaIterativa(float* pF,int nQuants)** que retorna la suma dels **nQuants** primer elements del vector apuntat per **pF**. Aprofiteu el codi **ex07previ.c** i deseu-lo com a **ex07.c**

```
int main(){
    float fPesos[N] = {0.1f,0.15f,0.2f,0.25f,0.3f},fValors[N] = {5.1f,4.2f,6.15f,7.4f,8.5f};

    vVisualitza("Pesos ",fPesos,N);
    printf("La suma no recursiva dels valors a fPesos és %.2f\n",fSumaIterativa(fPesos,N));
    vVisualitza("Valors",fValors,N);
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaIterativa(fValors,N));

    return 0;
}
```

```
$ ./ex07
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
```

RA1-5-2)(1 punts) Desenvolpeu la funció **fSumaIterativa(IF,nQuants)** que retorna la suma dels **nQuants** primer elements de la llista **IF**. Aprofiteu el codi **ex08previ.py** i deseu-lo com a **ex08.py**

```
fPesos = [0.1,0.15,0.2,0.25,0.3]
fValors = [5.1,4.2,6.15,7.4,8.5]

vVisualitza("Pesos ",fPesos,len(fPesos))
print("La suma no recursiva dels valors a fPesos és %.2f"%fSumaIterativa(fPesos,len(fPesos)))
vVisualitza("Valors",fValors,len(fValors))
print("La suma no recursiva dels valors a fValors és %.2f"%fSumaIterativa(fValors,len(fValors)))

$ python3 ex08.py
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
```

RA1-6-1)(1 punts) Desenvolpeu la funció **void vMultiplicaVectors(float* fV,float* fP,int nQuants)** que multiplica cadascun dels elements dels vectors apuntats per **fV** i **fP** i ho desa al mateix vector apuntat per **fV** d'**nQuants** elements. Aprofiteu el codi **ex09previ.c** i deseu-lo com a **ex09.c**

```
int main(){
    float fPesos[N] = {0.1f,0.15f,0.2f,0.25f,0.3f},fValors[N] = {5.1f,4.2f,6.15f,7.4f,8.5f};

    vVisualitza("Pesos ",fPesos,N);
    printf("La suma no recursiva dels valors a fPesos és %.2f\n",fSumaIterativa(fPesos,N));
    vVisualitza("Valors",fValors,N);
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaIterativa(fValors,N));
    vMultiplicaVectors(fValors,fPesos,N);
    vVisualitza("v x p ",fValors,N);
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaIterativa(fValors,N));

    return 0;
}
```

```
$ ./ex09
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
v x p : [ 0.51 0.63 1.23 1.85 2.55 ]
La suma no recursiva dels valors a fValors és 6.77
```

RA1-6-2)(1 punts) Desenvolpeu la funció **IMultiplicaVectors(IV,IP,nQuants)** que multiplica cadascun dels elements de les llistes **IV** i **IP** i ho retorna com a llista d'**nQuants** elements. Aproveiteu el codi **ex10previ.py** i deseu-lo com a **ex10.py**

```
fPesos = [0.1,0.15,0.2,0.25,0.3]
fValors = [5.1,4.2,6.15,7.4,8.5]
fProducte = []

vVisualitza("Pesos ",fPesos,len(fPesos))
print("La suma no recursiva dels valors a fPesos és %.2f"%fSumaIterativa(fPesos,len(fPesos)))
vVisualitza("Valors",fValors,len(fValors))
print("La suma no recursiva dels valors a fValors és %.2f"%fSumaIterativa(fValors,len(fValors)))
fProducte = lMultiplicaVectors(fValors,fPesos,len(fValors))
vVisualitza("v x p ",fProducte,len(fProducte))
print("La suma no recursiva dels valors a fProducte és %.2f"%fSumaIterativa(fProducte,len(fProducte)))
```

```
$ python3 ex10.py
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
v x p : [ 0.51 0.63 1.23 1.85 2.55 ]
La suma no recursiva dels valors a fProducte és 6.77
```

RA1-7)(1 punts) Desenvolpeu la funció **float* pfMultiplicaVectors(float* fV,float* fP,int nQuants)** que multiplica cadascun dels elements dels vectors apuntats per **fV** i **fP**, ho desa al mateix vector apuntat per **fV** d'**nQuants** elements i retorna l'adreça del primer element del vector **fV**. Aproveiteu el codi **ex09previ.c** i deseu-lo com a **ex09.c**

```
int main(){
    float fPesos[N] = {0.1f,0.15f,0.2f,0.25f,0.3f},fValors[N] = {5.1f,4.2f,6.15f,7.4f,8.5f};

    vVisualitza("Pesos ",fPesos,N);
    printf("La suma no recursiva dels valors a fPesos és %.2f\n",fSumaIterativa(fPesos,N));
    vVisualitza("Valors",fValors,N);
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaIterativa(fValors,N));
    vVisualitza("v x p ",pfMultiplicaVectors(fValors,fPesos,N),N);
    printf("La suma no recursiva dels valors a fValors és %.2f\n",fSumaIterativa(fValors,N));

    return 0;
}
```

```
$ ./ex11
Pesos : [ 0.10 0.15 0.20 0.25 0.30 ]
La suma no recursiva dels valors a fPesos és 1.00
Valors: [ 5.10 4.20 6.15 7.40 8.50 ]
La suma no recursiva dels valors a fValors és 31.35
v x p : [ 0.51 0.63 1.23 1.85 2.55 ]
La suma no recursiva dels valors a fValors és 6.77
```

RA1-8)(2 punts) Desenvolpeu **desxifrador.c** per a desxifrar el missatges xifrats per **xifrador.c**

```
$ ./xifrador
Escriuiu una frase sense accents ni caràcters no anglesos (ç, · o ñ) per a xifrar-la: Examen o
rdinari a DAW de la segona UF
Frase xifrada: "wzlJWmrEtsomltorlrSLPrsWr lraWfEmlrID"
```

```
$ ./desxifrador
Escriuiu la frase xifrada per a desxifrar-la: wzlJWmrEtsomltorlrSLPrsWr lraWfEmlrID
Frase desxifrada: "Examen ordinari a DAW de la segona UF"
```

```
int main(){  
    char szCadenaOriginal[N_MIDA_MAX],szCadenaXifrada[N_MIDA_MAX];  
  
    vDemaneuFraseXifrada(szCadenaXifrada);  
    vDesxifra(szCadenaXifrada,szCadenaOriginal);  
    printf ("Frase desxifrada: \"%s\\n\",szCadenaOriginal);  
  
    return 0;  
}
```

Molta sort a totes i tots !!!!