

Nom/prénom :

no camipro :

ME-213, Programmation pour ingénieur

Test intermédiaire 2022 + Corr

Durée 45min.

Signez cette page et mettez votre nom/prénom/no camipro sur les autres pages.

Aucun document, ni appareil électronique (calculatrice, smartphone, etc.) n'est autorisé. Vous pouvez utiliser un crayon et/ou un stylo et répondre en français ou anglais. Vos réponses doivent être lisibles !

Aucune feuille volante n'est reprise, seules les feuilles distribuées seront reprises. Vous pouvez utiliser le verso des pages comme brouillon.

Si vous devez faire des hypothèses, notez-les.

Signature :

--

--	--	--	--	--

Nom/prénom :

no camipro :

1. (20 pts) Ecrire en C la fonction **PairImpair ()** qui **sépare** les valeurs paires des valeurs impaires du vecteur d'entiers (int) **V** de taille **sz**.

PairImpair() retourne le nombre de valeurs paires trouvées ou -1 si **sz** est invalide. Après l'appel de la fonction, **V** contiendra en premier les valeurs *paires*, puis les valeurs *impaires*. Les valeurs retournées paires sont dans un ordre quelconque, idem pour les valeurs impaires.

Un vecteur (int) temporaire **T** (valeurs non initialisées) de taille **sz** vous est fourni, vous pouvez l'employer à votre convenance.

Exemples

avec $V = [1,2,3,4]$ **PairImpair (V,T,4)** retourne **0** et $V = [2,4,1,3]$ ou $[4,2,3,1]$

avec $V = [1,2]$ **PairImpair (V,T,-2)** retourne **-1** et $V = [1,2]$

Ecrire la fonction **PairImpair ()** suivante:

```
int PairImpair(int V[],int T[], int sz)
```

qui retourne :-1 si **sz** est invalide

le nombre de valeurs paires autrement

```
int PairImpair(int v[],int t[], int sz){
    if (sz<0) return -1;

    int ip=0;
    int ii=0;

    for (int i =0; i<sz;i++)
        if ((v[i]%2)==0) { // pair
            t[ip]=v[i];
            ip++;
        }
        else {
            t[sz-ii]=v[i];
            ii++;
        }

    for (int i =0; i<ip;i++)    v[i]    = t[i];
    for (int i =0; i<ii;i++)    v[i+ip] = t[sz-i];

    return ip;
}
```

Nom/prénom :

no camipro :

2. (4pts) a) Que calcule la fonction matlab suivante :

```
function [a, c] = maFunctionA(b)
    a = b(1);
    for index = 2:length(b)
        if b(index) > a
            a = b(index);
        end
    end
    c=find(b==a);
    c=c(1) ;
end
```

b) Que retourne `[a, c]=maFunctionA()` pour `b = [8, 5, 9, 7, 9, 3, 4]` ?

a) `a=max(b)` et `c=index de a`

b) `9, 3`

3. (6pts) En Matlab, écrire la fonction anonyme **Seuil** qui converti (et retourne) l'image en niveaux de gris **G** (i.e. un tableau 2D de valeurs 0..255) en une image "noir/blanc" avec '0' pour les valeurs en dessous du seuil **S** (scalaire) et **S** pour les autres valeurs.

```
Seuil = @(G,S) S*(R>=S)
```

.....

4. (10pts) En matlab, quelles sont les valeurs successives de **F**. **F** est initialement vide. Les valeurs de **F** se réfèrent aux valeurs précédentes de **F**.

`F(2,3)=7`

`F= [0 0 0; 0 0 7]`

`F(3 :-1 :1)= 2`

`F= [2 2 0; 2 0 7]`

`F=6-find(find(find(F>1)))`

`F= [5; 4; 3; 2]`

`F(:, :) = 3`

`F= [3; 3; 3; 3]`

`F=prod(F)`

`F= 81`

5. (10pts) La fonction C1 () ne compile pas !

Trouvez 5 warnings et/ou erreurs, (il y en a plus !)

Entourez les problèmes dans le code et **justifiez** brièvement ci-dessous la raison de ce problème.

```

int C1(const int A[], int *b, unsigned int c ) {
    double d;
    if (c < 0) {
        b = A[0];
        d++;
        int e = 5;
        if (d == 1);
            d += A[d];
    }
    else {
        return *b/2.0;
        A[1] = e* *b;
    }
}

```

1)

2)

3)

4)

5)