

Nom/prénom :

no camipro :

ME-213, Programmation pour ingénieur

Test intermédiaire 2023-A + 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.) est autorisé. Vous pouvez utiliser un crayon et/ou un stylo et répondre en français ou anglais. Vos réponses doivent être lisibles ! Note : Il n'y a pas deux copies identiques !

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

Si vous devez faire des hypothèses, notez-les.

Il y a souvent plusieurs solutions possibles, seule une est indiquée dans ce document.
De même il y avait plusieurs séries, uniquement la série A est fournie.

Signature :

--	--	--	--	--	--	--

Nom/prénom :

no camipro :

1a. (20 pts) Écrire en C la fonction **NoDoublons (U,V,sz)** qui sélectionne et retourne dans **U** les valeurs uniques du vecteur d'entiers **V** de taille **sz**.

Après l'appel à **NoDoublons ()**, **U** contiendra les valeurs de **V** sans ses doublons.

U a la même taille que **V** et est initialisé avec des 0 avant l'appel à **NoDoublons ()**.

Les valeurs de **V** sont toujours différentes de 0.

NoDoublons () retourne le nombre de valeurs uniques trouvées, ou -1 si **sz** est <0.

Exemples

avec $V = [1,4,2,2,1]$ **NoDoublons (V,U,5)** retourne **3** et $U = [1,4,2,0,0]$

avec $V = [1,4,2,2,1]$ **NoDoublons (V,U,-2)** retourne **-1** et $U = [0,0,0,0,0]$

Écrire la fonction **NoDoublons()** suivante:

```
int NoDoublons(const int V[],int U[], int sz)
```

qui retourne : -1 si **sz** est invalide (i.e. < 0)

le nombre de valeurs uniques autrement

```
int NoDoublons(const int A[], int B[], int szA){  
    if (szA<0) return -1;  
    int pos=0;  
    for (int i=0; i<szA; i++) {  
        int exist =0;  
        for (int j=0; j<szA; j++) {  
            if (A[i] == B[j]){  
                exist = 1;  
                break;  
            }  
        }  
        if (exist == 0){  
            B[pos]= A[i];  
            pos ++;  
        }  
    }  
    return pos;  
}
```

Nom/prénom :

no camipro :

2a. (4pts) En c, en utilisant les opérateurs bit à bit, (& | ! << >>) écrire le corps de la fonction **int Pair(int x)** qui retourne '1' si x est pair et '0' autrement (corps de la fonction sur 1 ligne)

```
int Pair(int x){  
    return (~x&1)-----  
}
```

3a. (4pts) En Matlab, que valent A et B après l'exécution des lignes ci-dessous

A = linspace(10,1,10)

A = [10 9 8 7 6 5 4 3 2 1]-----

B = find(A>5)

B = [1 2 3 4 5]-----

4a. (6pts) En Matlab, écrire la fonction anonyme AllMax(V) qui retourne un vecteur contenant la valeur maximale du vecteur **V** et les positions de ces maximums.

Ex. **AllMax**([1 2 3 2 1 3 1]) -> [3 3 6] et **AllMax**([]) -> []

```
AllMax = @(V) ([max(V), find(V==max(V))])
```

5a. (4pts) En Matlab, quelles sont les valeurs successives de A après l'exécution des lignes ci-dessous A est initialement vide.

A(2,3)=7

```
0 0 0  
0 0 7
```

A(2 :-1 :1)= 2

```
2 0 0  
2 0 7
```

Nom/prénom :

no camipro :

6a. (4pts) En Matlab, vectorisez le code ci-dessous.

```
k = 0;
for t = 0:.01:10
    k = k + 1;
    y(k) = sin(t) * sin(t) + 1;
end
```

```
-----
t = 0:.01:10
-----
y = sin(t) .* sin(t) + 1
-----
```

7a. (3pts) Qu'affiche la console après l'exécution de PrintI()

```
void PrintI(void){
    int i = 2;
    {
        if (i = 8)
            printf("%d ", i);
            {
                int i=4;
                i--;
                printf("%d ", i);
            }
        printf("%d ", i); }
    }
```

Console) 838-----