

Nom/prénom :

no camipro :

# ME-213, Programmation pour ingénieur

## Test intermédiaire 2024 **+corr**

---

Durée 45min.

0

Veillez signer cette page.

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 en 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.

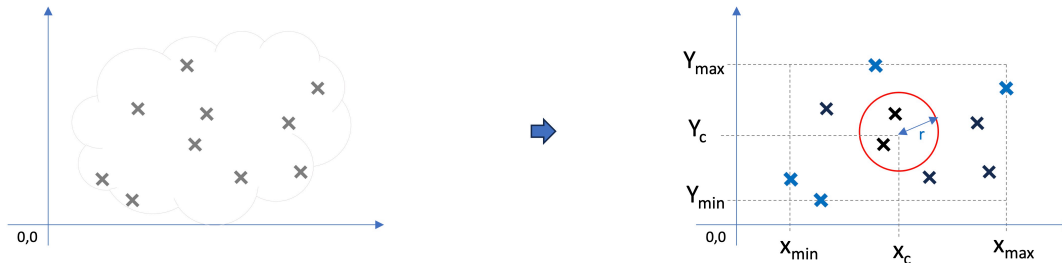
place:

*Il y a plusieurs variations de cette série*

Signature :

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

1. (20 pts) Les vecteurs  $\mathbf{Xs}$  et  $\mathbf{Ys}$  contiennent les coordonnées d'un nuage de points. Les coordonnées  $\mathbf{Xmin}$ ,  $\mathbf{Ymin}$ ,  $\mathbf{Xmax}$ ,  $\mathbf{Ymax}$  définissent le rectangle de surface minimale contenant tous les points. La fonction C `myF()` retourne le nombre de points se trouvant à l'intérieur d'un disque de rayon  $\mathbf{r}$  situé au centre  $\mathbf{Xc}, \mathbf{Yc}$  du rectangle.



Ecrire en C la fonction `myF()` qui calcul le nombre de point à l'intérieur du cercle au centre du nuage de points. S'il y a moins de 4 points, `myF()` retourne -1.

```
int myF (int nb, double Xs[], double Ys[], double r)
```

Ex.

```
double Xs[] = {0, 10, 10, 0, 1, 2, 3, 4, 5.5, 6 };
double Ys[] = {0, 0, 10, 10, 1, 2, 3, 4, 5, 6};
```

```
int nb1 = myF (10, Xs, Ys, 1.5); -> 3
int nb2 = myF (10, Xs, Ys, 1.0); -> 1
int nb3 = myF (10, Xs, Ys, 0.1); -> 0
int nb4 = myF ( 3, Xs, Ys, 1.0); -> -1
```

$\mathbf{Xs}$ ,  $\mathbf{Ys}$  contiennent les coordonnées X et Y des points;  $\mathbf{nb}$  contient le nombre d'éléments de  $\mathbf{Xs}$  et  $\mathbf{Ys}$ ;  $\mathbf{r}$  est le rayon du cercle

Les vecteurs  $\mathbf{Xs}$  et  $\mathbf{Ys}$  sont valides (pas besoin de tester) et contiennent chacun le même nombre de valeurs  $\mathbf{nb}$ . Les valeurs des vecteurs  $\mathbf{Xs}$  et  $\mathbf{Ys}$  sont entre 0 et 1000.

```
int PtsInCenter(int nb, double Xs[], double Ys[], double r){
    double minX=1000, minY=1000, maxX=0, maxY=0;

    if (nb<4) return -1;
    for (int i =0; i<nb;i++){
        if (Xs[i]>maxX) maxX=Xs[i];
        if (Xs[i]<minX) minX=Xs[i];
        if (Ys[i]>maxY) maxY=Xs[i];
        if (Ys[i]<minY) minY=Xs[i];
    }
    double centerX = (maxX+minX)/2.0, centerY = (maxY+minY)/2.0;

    int nbpts=0;
    for (int i =0; i<nb;i++){
        if (((Xs[i]-centerX)*(Xs[i]-centerX)+(Ys[i]-centerY)*(Ys[i]-centerY)) <= r*r )
            nbpts++;
    }

    return nbpts;
}
```

2. (4 pts) En utilisant les opérateurs bit à bit (<<, >>, &, |, !) écrire en C le corps de la fonction (sur 1 ligne) **int Mul4(int x)** qui retourne '1' si x est un multiple de 4 et '0' autrement. Vous ne pouvez **pas** employer modulo '%' ou division '/'.

```
int Mul4(int a){
    return !(a & 3);
}
```

3. (2 pts) En C, Si `int E[]={4,5,6}`, que vaut **n** après l'appel à `myF(E)` justifiez ?

```
int myF(int E[]) {
    return sizeof(E)/sizeof(E[0]);}
```

```
int n=myF(E);
```

n= 2, pour pointeur 64bit et int 32bit

4. (6 pts) En matlab, quelles sont les valeurs successives de **F**. Les valeurs de **F** se réfèrent aux valeurs précédentes de **F**.

```
F=zeros(2)
```

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

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

```
F= [0 0; 0 1; 0 1]
```

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

```
F= [2 0; 2 1; 2 1]
```

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

```
A=10:-2:0
```

```
A= [10 8 6 4 2 0]
```

```
A(find(A>3)) = - find(A>3)
```

```
A= [ -1 -2 -3 -4 2 0]
```

6. (4 pts) Ecrire en matlab, **sans boucle**, la fonction anonyme **RemoveSup** (X,s) qui retourne les éléments du vecteur **X** dont la valeur est strictement inférieure au seuil **s**.

`RemoveSup = @(X,s) X(find(X<s))`

7. (10pts) La fonction `fcnC()` ne compile pas ! Trouvez **10** 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 fcnC(const int A[]; char *b, unsigned int h ) {
    int f2(void) {return i;}
    unsigned short d=10065535;
    char* c ="hello";
    if (A[0] > *A) {
        b = c[0];
        d=d++;
        c[2]=c[1];
        int e = 5;
        if (h = 88);
            h += A[h];
    }
    else {
        d=c[2]-'l';
        if (d+1 >d) {
            return *b/2.0;
        }
        int* f=&c;
        e = *f* *b;
    }
    return;
};
```

Pas le bon séparateur

Pas de déclaration de fcn dans une autre fcn

i inconnu

i pas initialisé

10065535 > max unsigned short

A[0] est \*A -> donc égal et variations

b -> \*b

résultat non défini, modifications aléatoire de 'd'

c[] non modifiable

assignation de h

if () tjs vrai

A[h] n'existe peut être pas

si vrai -> ;

d+1 > d toujours vrai

donc toujours return et 2 lignes

suyvantes jamais executée

retourne un double, int attendu

e inconnu ici

retourne rien, int attendu

manque }

Variations de ces réponses acceptées