



## Contrôle Continu

Aucun document n'est autorisé  
Les solutions doivent être rédigées en C  
Les appareils portables doivent être éteints

### 1 Affichage

8 pts. ⌚25'

Qu'affichent les deux programmes suivants :

```
1 #include <stdio.h>
2 void main () {
3     int n=0 , m=1;
4     n = m;
5     m = n;
6     printf ("n=%d, m=%d\n", n, m);
7     if(n==m)
8         n--;
9     else
10        m++;
11    printf ("n=%d, m=%d\n", n, m);
12    if(n)
13        n++;
14    if(m)
15        m--;
16    printf ("n=%d, m=%d\n", n, m);
17    (n==m)? n++ : m-- ;
18    printf ("n=%d, m=%d\n", n, m);
19 }
```

```
1 #include <stdio.h>
2
3 void main ()
4 {
5     int i, a;
6     a=3;
7     for (i=0; i<4; i ++ )
8     {
9         switch (a)
10        {
11            case 0 || 1 : a = a - 1 ;
12            case 2 : a = a +2; break ;
13            case 3 : a = a-3; break ;
14            case 4 : a = a+4;
15            default : a = 1 ; break ;
16        }
17    printf (" a = %d \n", a );
18 }
19 }
```

### 2 Remise de prix

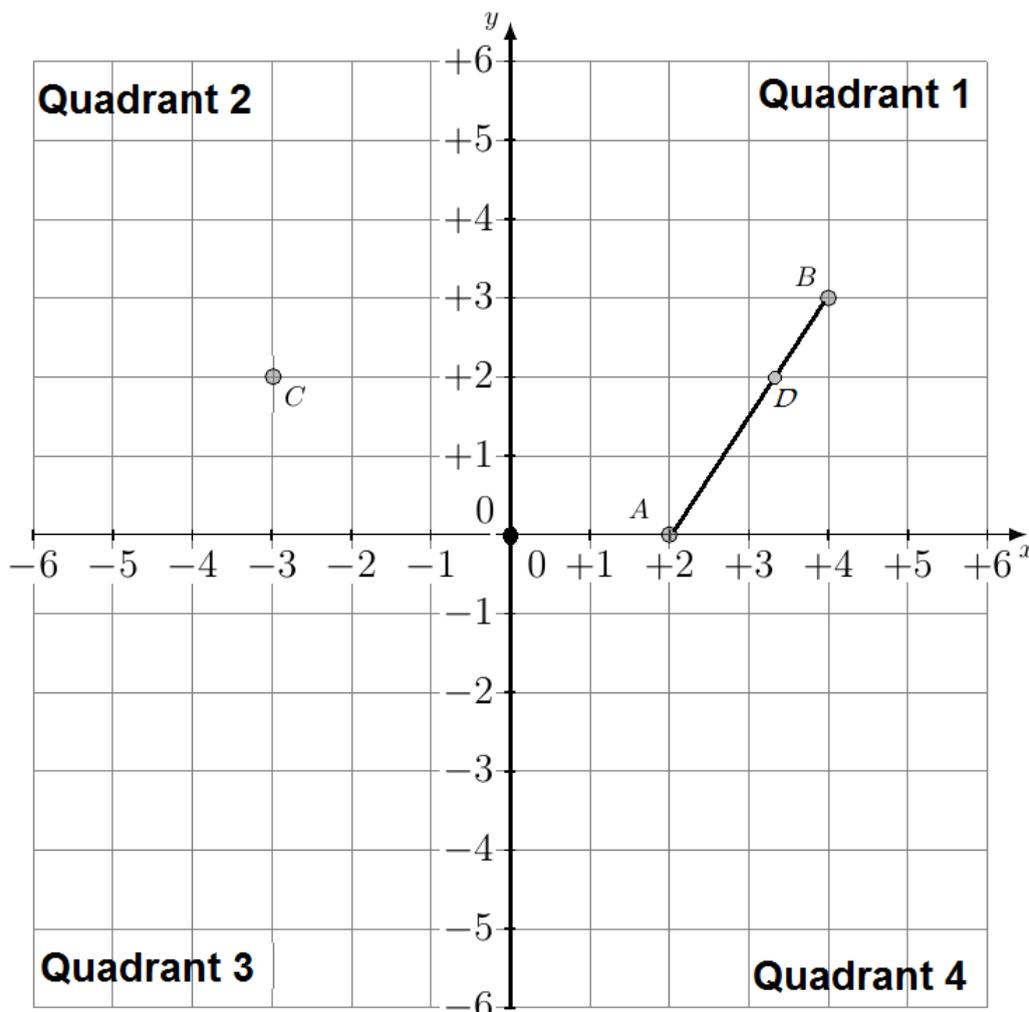
4 pts. ⌚25'

Écrire un programme qui demande un prix unitaire et la quantité d'un article commandé puis calcule et affiche :

1. Le prix hors taxe de la quantité commandée,
2. La TVA sur la quantité commandée ainsi que le prix TTC, sachant que le taux de TVA est égal à 19% et le prix TTC est égal au prix hors taxe plus la TVA calculée,
3. La remise ainsi que le prix total de la quantité commandée, sachant que la remise est calculée sur le prix TTC de la manière suivante :

Quantité	Remise
inférieur à 5	0%
entre 5 et 9	5%
entre 10 et 49	10%
entre 50 et 99	20%
à partir de 100	30%

Un plan cartésien est défini par 2 axes perpendiculaires : l'axe des abscisses (les  $x$ ) qui est horizontal et l'axe des ordonnées (les  $y$ ) qui est vertical. Les deux axes se croisent au point  $(0, 0)$  appelé repère. Tout point  $P$  peut donc être repéré par deux nombres réels (abscisse  $x_p$  et ordonnée  $y_p$ ) appelés coordonnées cartésiennes. Le plan cartésien est divisé en 4 régions que l'on appelle les quadrants numéroté de 1 à 4 comme indiqué dans la figure ci-dessous.



Écrire un programme qui :

1. demande à l'utilisateur de saisir les coordonnées cartésiennes de trois points  $A(x_A, y_A)$ ,  $B(x_B, y_B)$  et  $C(x_C, y_C)$ .
2. calcule et affiche la distance entre les deux point A et B. Sachant que la distance AB est calculée par la formule suivante :  $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$ .
3. affiche le numéro du quadrant dans lequel se trouve le point C.
4. affiche si le point C se trouve sur le segment formé par les deux points A et B ou non.

**Quelques illustrations :**

- Les points indiqués sur la figure ci-dessus ne sont que des exemples. Votre programme devra traiter n'importe quels points saisis par l'utilisateur.
- Par exemple, le point B a les coordonnées cartésiennes  $(4,3)$ .
- Le point C ne se trouve pas sur le segment  $[AB]$  alors que le point D s'y trouve.
- Le point C se trouve dans le quadrant numéro 2 alors que les autres points (A, B et D) se trouvent tous dans le quadrant 1.



## Correction du contrôle continu

Aucun document n'est autorisé  
Les solutions doivent être rédigées en C  
Les réponses aux questions de l'exercice 3 doivent être reportées sur le sujet.  
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

### 1 Affichage

8 pts. ☺25'

Qu'affichent les deux programmes suivants :

```
1 #include <stdio.h>
2 void main () {
3     int n=0 , m=1;
4     n = m;
5     m = n;
6     printf ("n=%d, m=%d\n", n, m);
7     if(n==m)
8         n--;
9     else
10        m++;
11    printf ("n=%d, m=%d\n", n, m);
12    if(n)
13        n++;
14    if(m)
15        m--;
16    printf ("n=%d, m=%d\n", n, m);
17    (n==m)? n++ : m-- ;
18    printf ("n=%d, m=%d\n", n, m);
19 }
```

```
1 #include <stdio.h>
2
3 void main ()
4 {
5     int i, a;
6     a=3;
7     for (i=0; i<4; i ++ )
8     {
9         switch (a)
10        {
11            case 0 || 1 : a = a - 1 ;
12            case 2 : a = a +2; break ;
13            case 3 : a = a-3; break ;
14            case 4 : a = a+4;
15            default : a = 1 ; break ;
16        }
17        printf (" a = %d \n", a );
18    }
19 }
```

### Solution

Affichage

```
n=1, m=1
n=0, m=1
n=0, m=0
n=1, m=0
```

Affichage

```
a=0
a=1
a=2
a=4
```

### 2 Remise de prix

4 pts. ☺30'

Écrire un programme qui demande un prix unitaire et la quantité d'un article commandé puis calcule et affiche :

1. Le prix hors taxe de la quantité commandée,
2. La TVA sur la quantité commandée ainsi que le prix TTC, sachant que le taux de TVA est égal à 19% et le prix TTC est égal au prix hors taxe plus la TVA calculée,
3. La remise ainsi que le prix total de la quantité commandée, sachant que la remise est calculée sur le prix TTC de la manière suivante :

Quantité	Remise
inférieur à 5	0%
entre 5 et 9	5%
entre 10 et 49	10%
entre 50 et 99	20%
à partir de 100	30%

### Solution

```

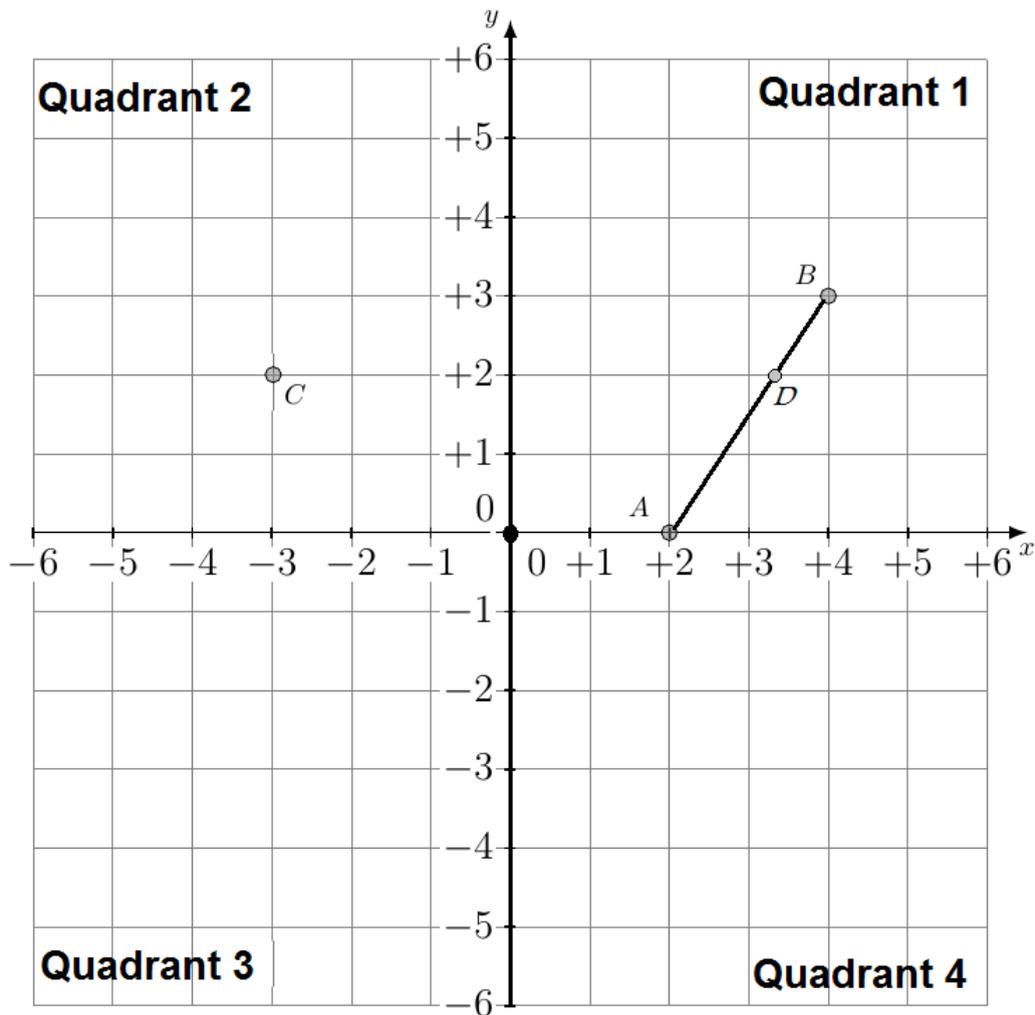
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int quantite=0;
7     double prix, prix_hors_TAX, TVA, prix_TTC, remise, prix_total;
8     printf("Donnez la quantite : \n");
9     scanf("%d",&quantite);
10    printf("Donnez le prix unitaire : \n");
11    scanf("%lf",&prix);
12
13    prix_hors_TAX = prix*quantite;
14    printf("prix hors taxe = %f\n", prix_hors_TAX);
15
16    TVA = prix_hors_TAX*0.19;
17    printf("TVA =%f\n", TVA);
18    prix_TTC = prix_hors_TAX + TVA;
19    printf("Le prix TTC est : %f\n", prix_TTC);
20
21    if (quantite<=0)
22        printf("cette quantite n est pas correcte");
23    else
24    {
25        if(quantite>=5 && quantite<10)
26            remise = 0.05*prix_TTC;
27        else if(quantite>=10 && quantite<50)
28            remise = 0.1*prix_TTC;
29        else if(quantite>=50 && quantite<100)
30            remise = 0.2*prix_TTC;
31        else if(quantite>=100)
32            remise = 0.3*prix_TTC;
33        prix_total = prix_TTC-remise;
34        printf("la remise = %f% \n", remise );
35        printf("prix total = %f", prix_total);
36    }
37 }

```

### 3 Coordonnées cartésiennes

8 pts. ☹️35'

Un plan cartésien est défini par 2 axes perpendiculaires : l'axe des abscisses (les  $x$ ) qui est horizontal et l'axe des ordonnées (les  $y$ ) qui est vertical. Les deux axes se croisent au point (0, 0) appelé repère. Tout point P peut donc être repéré par deux nombres réels (abscisse  $x_p$  et ordonnée  $y_p$ ) appelés coordonnées cartésiennes. Le plan cartésien est divisé en 4 régions que l'on appelle les quadrants numéroté de 1 à 4 comme indiqué dans la figure ci-dessous.



Écrire un programme qui :

1. demande à l'utilisateur de saisir les coordonnées cartésiennes de trois points  $A(x_A, y_A)$ ,  $B(x_B, y_B)$  et  $C(x_C, y_C)$ .
2. calcule et affiche la distance entre les deux point A et B. Sachant que la distance AB est calculée par la formule suivante :  $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$ .
3. affiche le numéro du quadrant dans lequel se trouve le point C.
4. affiche si le point C se trouve sur le segment formé par les deux points A et B ou non.

**Quelques illustrations :**

- Les points indiqués sur la figure ci-dessus ne sont que des exemples. Votre programme devra traiter n'importe quels points saisis par l'utilisateur.
- Par exemple, le point B a les coordonnées cartésiennes (4,3).
- Le point C ne se trouve pas sur le segment [AB] alors que le point D s'y trouve.
- Le point C se trouve dans le quadrant numéro 2 alors que les autres points (A, B et D) se trouvent tous dans le quadrant 1.

**Solution**

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<math.h>
4 int main()
5 {
6     double xA,yA,xB,yB,xC,yC;
7     double AB,AC,CB;
8     int quadrant = 0;
9
10    printf("Entrez les coordonnees cartesiennes de A \n");
11    scanf("%lf%lf",&xA,&yA);
12    printf("Entrez les coordonnees cartesiennes de B \n");
13    scanf("%lf%lf",&xB,&yB);
14    printf("Entrez les coordonnees cartesiennes de C \n");
15    scanf("%lf%lf",&xC,&yC);
16
17    AB = sqrt( (xB-xA)*(xB-xA)+(yB-yA)*(yB-yA) );
18    printf("La distance entre A et B est : %f.\n", AB);
19
20    if(yC >=0)
21        if (xC>=0)
22            quadrant = 1;
23        else
24            quadrant = 2;
25    else
26        if (xC<0)
27            quadrant = 3;
28        else
29            quadrant = 4;
30    printf("Le point C(%f,%f) se trouve dans le quadrant %d.\n",xC, yC, quadrant );
31
32    AC = sqrt( (xC-xA)*(xC-xA)+(yC-yA)*(yC-yA) );
33    CB = sqrt( (xB-xC)*(xB-xC)+(yB-yC)*(yB-yC) );
34    if( AB == (AC + CB) )
35        // pour être un peu plus correct : if( fabs(AB - (AC + CB)) <= 0.00000001
36        printf("Le point C est sur le segment [AB].\n");
37    else
38        printf("Le point C n'est pas sur le segment [AB].\n");
39 }

```