

Algorithmique et Programmation – Examen (durée 1h30)

Deuxième session du 27 juin 2024

NOTES : Aucun document autorisé. Sont interdits les calculatrices, les téléphones, ainsi que tout autre ustensile de calcul ou de communication.

Exercice 1 : Questions de connaissances générales (3 points)

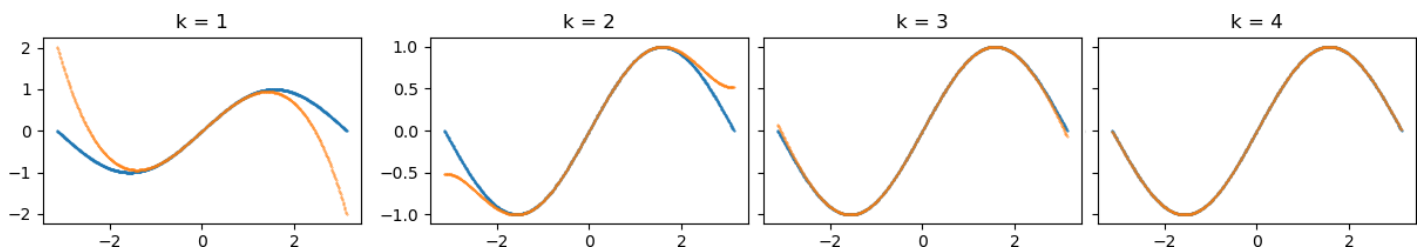
6 questions

Exercice 2 : Trois fonctions et un programme à compléter (8 points)

Nous souhaitons calculer la fonction sinus par son développement en série entière (ou plutôt développement en série de Taylor des fonctions entières), permettant ainsi son calcul par des additions et des multiplications (opérations dont nous disposons dans le langage C) :

$$\sin(x) = \sum_{n=0}^{+\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Ne pouvant nous contenter d'exécuter notre programme indéfiniment, la série calculée sera d'ordre k (càd pour n dans $[0, k]$). Le tracer des courbes pour quatre valeurs de k sont illustrées ci-dessous (en orange), en comparaison de la courbe obtenue avec la fonction `sin(x)` de la bibliothèque standard du langage C (en bleu) :



Ainsi, la précision du calcul augmente avec la valeur de k . Dans la suite de cet exercice, nous écrivons un programme permettant de mesurer l'écart entre la fonction `sin(x)` standard (qui est elle-même d'une précision limitée) et notre propre fonction `sinus(x, k)`, selon des valeurs de k données par l'utilisateur du programme. L'exemple d'exécution suivant illustre ce programme :

```

C:\ExemplesC> exercice2.exe
Donnez k : 3
x=0.000000 : sinus(x) = 0.000000, sin(x) = 0.000000, diff = 0.000000
x=0.392699 : sinus(x) = 0.382683, sin(x) = 0.382683, diff = 0.000000
x=0.785397 : sinus(x) = 0.707106, sin(x) = 0.707106, diff = 0.000000
x=1.178096 : sinus(x) = 0.923867, sin(x) = 0.923879, diff = 0.000012
x=1.570795 : sinus(x) = 0.999843, sin(x) = 1.000000, diff = 0.000157
x=1.963494 : sinus(x) = 0.922726, sin(x) = 0.923880, diff = 0.001154
x=2.356192 : sinus(x) = 0.701241, sin(x) = 0.707108, diff = 0.005867
x=2.748891 : sinus(x) = 0.359605, sin(x) = 0.382686, diff = 0.023080
x=3.141590 : sinus(x) = -0.075217, sin(x) = 0.000003, diff = 0.075220
Difference cumulee en 9 points sur [0, Pi] : 0.105491332
C:\ExemplesC>

```

Pour les besoins de ce programme, nous allons nous munir de trois fonctions : `factorielle(n)`, `puissance(x, n)` et enfin `sinus(x, k)`. Les quatre parties du code du programme sont à compléter séparément et indépendamment ci-après (c'est-à-dire, chaque partie sera vérifiée indépendamment des autres).

Rappels :

- La factorielle d'un nombre entier positif n est le produit de lui-même et de tous les entiers inférieurs jusqu'à 1 : $n! = n \times (n-1) \times \dots \times 3 \times 2 \times 1$, sachant que $0! = 1$. Par exemple, $1! = 1$, $2! = 2$ et $3! = 6$.
- La puissance n d'un nombre réel x (aussi notée x^n) est sa multiplication n fois par lui-même, c'est-à-dire : $x \times x \times \dots \times x$ avec n facteurs, sachant que la puissance 0 vaut 1. Par exemple, $x^3 = x \times x \times x$, $x^2 = x \times x$, $x^1 = x$ et $x^0 = 1$.

Notation : Des points sont attribués même si échec des tests. Toutes les cases d'une question doivent être justes pour obtenir tous les points de la question.

1) Complétez ci-dessous le code de la fonction qui calcule la factorielle :

```

----- <----->
----- <----->

unsigned long ___ factorielle (unsigned long ___ n) _
    unsigned long ___ i, res = 1 _
    ___ (i = _ _ i __ n _ i++) _
        ___ __ i _
    -
    ----- res _
-

```

2) Complétez ci-dessous le code de la fonction qui calcule la puissance :

```

long double puissance (long double x, unsigned long ___ n) _
    long double res = 1 _
    long ___ i _
    ___ (i = 1 _ i __ _ _ i++) _
        ___ *= _ _
    -
    ----- res _
-

```

3) Complétez ci-dessous le code de la fonction qui calcule le sinus :

```

long double sinus (long double x, unsigned long ___ k) _
    long double res = 0 _
    unsigned long ___ n _
    ___ (n = _ _ n __ _ _ n++) _
        ___ += (n % 2 ? __ : _) * _____ (_, 2 _ _ _ _) / _____ (2 _ _ _ _ _) _
    -
    ----- res _
-

```

4) Complétez ci-dessous le code de la fonction principale du programme mesurant l'écart entre les deux fonctions :

```

--- ---- ( ) _
    unsigned long ___ k _
    long double x, y1, difference, cumulee = 0 _
    double y2 _

    ----- ("Donnez k : ") _
    ----- ("%ld", __) _

    ___ (x = 0 _ x <= 3.14159 _ x += 3.14159/8) _
        y1 = ----- (x, k) _
        y2 = sin (x) _
        difference = (y1 _ y2) _ y1 - y2 _ y2 - y1 _
        ----- ("x=%Lf : sinus(x) = %9.6Lf, sin(x) = %9.6f, diff = %9.6Lf\n", _, __, __, -----) _
        cumulee __ ----- _
    -

    ----- ("Difference cumulee en 9 points sur [0, Pi] : %11.9Lf\n", cumulee) _

    ----- 0 _
-

```

Exercice 3 : Une fonction et deux programmes à écrire (9 points)

L'objectif de cet exercice est de pouvoir déterminer la position de points par rapport à une droite affine.

Notation :

- Pour valider un test, l'exécution doit correctement répondre à la question en permettant d'obtenir le calcul demandé. Les points de la question sont obtenus seulement si réussite de tous les tests.
- L'exécution peut être vérifiée en appuyant sur le bouton "Vérifier" situé en-dessous de la zone de saisie.
- Vous pouvez exécuter autant de fois que nécessaire sans perte de points.

a) Nous écrivons une fonction en langage C qui prend en paramètres les coordonnées d'un point (Px, Py) et d'une droite affine définie par l'équation $y = m.x + p$ selon coefficient directeur m et l'ordonnée à l'origine p , qui retourne :

- la valeur nulle si le point est sur la droite (c.-à-d. zéro),
- une valeur positive si le point est au-dessus de la droite,
- et une valeur négative si le point est en-dessous de la droite.

Le prototype de la fonction sera le suivant :

```
char position_sur_la_droite (float Px, float Py, float m, float p) ;
```

Écrivez la déclaration de la fonction.

b) Pour tester la fonction du (a), nous écrivons maintenant un programme en langage C dans lequel l'utilisateur saisira les coordonnées d'un point, puis le programme indiquera la position du point par rapport à la droite d'équation $y = -x - 2$ (c.-à-d., en-dessous, au-dessus ou sur la droite).

Ci-dessous, trois exemples d'exécution de ce programme, accompagnés d'une représentation graphique de la droite et des points :

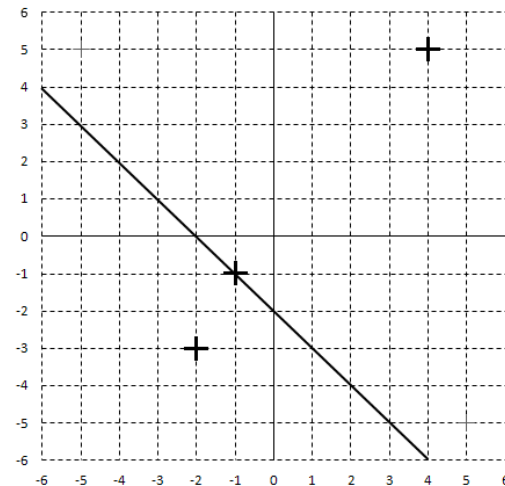
```

C:\ExemplesC> exercicelb.exe
Donnez l'abscisse du point : -1
Donnez l'ordonnee du point : -1
Le point est sur la droite.
C:\ExemplesC>

C:\ExemplesC> exercicelb.exe
Donnez l'abscisse du point : -2
Donnez l'ordonnee du point : -3
Le point est en-dessous de la droite.
C:\ExemplesC>

C:\ExemplesC> exercicelb.exe
Donnez l'abscisse du point : 4
Donnez l'ordonnee du point : 5
Le point est au-dessus de la droite.
C:\ExemplesC>

```



Écrivez le code du programme.

c) Nous écrivons maintenant un programme en langage C dans lequel l'utilisateur saisira le coefficient directeur puis l'ordonnée à l'origine d'une droite. Ensuite, le programme comptera combien des 10 points suivants sont sur la droite, et calculera le barycentre des points qui ne sont pas sur la droite, les abscisses (*point_x*) et ordonnées (*point_y*) des 10 points à examiner étant données par les deux tableaux suivants (que vous déclarerez au début du programme) :

```

float point_x[10] = { 1.8, 2.9, 0.0, -3.2, -5.0, -1.9, 3.4, 0.7, -5.5, 3.8 };
float point_y[10] = { -4.5, 4.2, -1.0, -2.6, 2.8, 1.2, -1.9, -4.7, 0.0, 3.9 };

```

Par exemple, les coordonnées du 2^e point sont (2.9, 4.2).

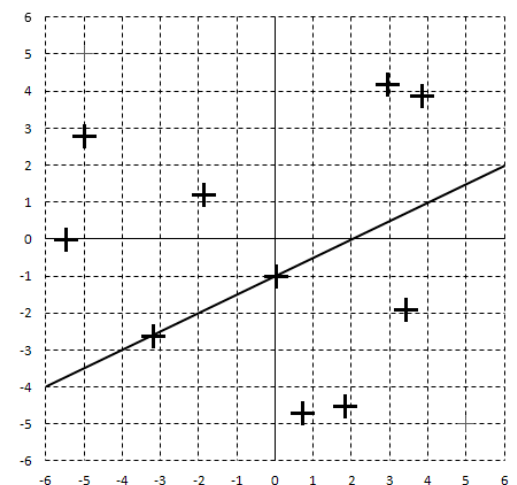
Enfin, le programme indiquera par rapport à la droite quelle est la position du barycentre des points qui ne sont pas sur la droite. Ce programme doit lui aussi utiliser la fonction du (a).

Ci-dessous, un exemple d'exécution de ce programme et une représentation graphique des points :

```

C:\ExemplesC> exercicelc.exe
Donnez le coefficient directeur (m) : 0.5
Donnez l'ordonnee a l'origine (p) : -1.0
Le point (0.0;-1.0) appartient a la droite.
Le point (-3.2;-2.6) appartient a la droite.
La droite passe par 2 des points.
Le barycentre des points extérieurs est (-0.0;-0.0).
Ce barycentre est au-dessus de la droite.
C:\ExemplesC>

```



Rappel : le barycentre d'un ensemble de n points de coordonnées (x_i, y_i) est le point de coordonnées $(\frac{\sum_i x_i}{n}, \frac{\sum_i y_i}{n})$.

Écrivez le code du programme.