Test sur les algorithmes (45 minutes)

$\boxed{1}$ On considère l'algorithme suivant :

Entrées :
Saisir x
Saisir y

Traitement :
z prend la valeur $2x - y$
y prend la valeur $2y - 3z$
t prend la valeur $5z + y - 4x$

Sortie :
Afficher t

1°) Exécuter l'algorithme « à la main » pour $x = 7$ et $y = 2$.

2°) Recommencer pour d'autres valeurs initiales de x et y.

3°) Expliquer* le résultat obtenu.

$\boxed{2}$ On considère la fonction f définie sur $\mathbb{R}$ par $f(x) = x^2 - 2$ si $x < 3$ et $f(x) = 2x - 1$ si $x \geqslant 3$.
Écrire en langage naturel un algorithme qui demande la valeur de x et affiche son image par f.

$\boxed{3}$ 1°) Écrire en langage naturel un algorithme qui calcule la somme des carrés des 50 premiers entiers non nuls : $1^2 + 2^2 + ... + 50^2$.

2°) Le programmer et donner la somme affichée.

$\boxed{4}$ 1°) Expliquer ce que produit l'algorithme suivant (on pourra le faire tourner à la main).

Initialisations :
k prend la valeur 0
u prend la valeur 1

Traitement :
Tantque $u > 0,1$ **Faire**
  k prend la valeur $k + 1$
  u prend la valeur $0,5u$
FinTantque

Sortie :
Afficher u et k

2°) Modifier l'algorithme (le réécrire) pour qu'il donne le plus petit entier naturel k tel que $0,9^k < 0,001$.

3°) Le programmer et donner la valeur de k obtenue.

* On attend une démonstration.

Solution

Il faut bien faire la différence entre algorithme (écrit en langage naturel) et programme.

1

1°) **Exécuter l'algorithme « à la main » pour $x = 7$ et $y = 2$.**

On trouve 0 en sortie.

On notera bien que pour la variable t, on reprend la valeur de y précédente et non celle du début.

La calculatrice est ici inutile (les calculs sont très simples et peuvent être faits à la main).

Attention, on demande d'exécuter l'algorithme à la main, pas de le programmer.
Cela dit, on peut effectivement le programmer pour vérifier les résultats obtenus.

2°) **Recommencer pour d'autres valeurs initiales de x et y.**

On trouve encore 0 en sortie.

3°) **Expliquer le résultat obtenu.**

On doit faire une démonstration.

En effet, y prend la valeur $2y - 3(2x - y) = 5y - 6x$ et t prend la valeur $5(2x - y) + (5y - 6x) - 4x = 0$.
Pour toutes les valeurs de x et y en entrée, l'affichage donne toujours 0 en sortie.

2 $f(x) = x^2 - 2$ si $x < 3$ et $f(x) = 2x - 1$ si $x \geqslant 3$

Écrire en langage naturel un algorithme qui demande la valeur de x et affiche son image par f.

Entrée :
Saisir x

Traitement :
Si $x < 3$ Alors y prend la valeur $x^2 - 2$
| **Sinon** y prend la valeur $2x - 1$
FinSi

Sortie :
Afficher y

3

1°) **Écrire en langage naturel un algorithme qui calcule la somme $1^2 + 2^2 + ... + 50^2$.**

On connaît le nombre d'itérations donc on utilise une boucle « Pour » (on peut aussi utiliser une boucle « Tantque » mais c'est plus maladroit).

Initialisation :
S prend la valeur 0

Traitement :
Pour *k* allant de 1 à 50 **Faire**
    S prend la valeur $S + k^2$
FinPour

Sortie :
Afficher *S*

2°) **Le programmer et donner la somme affichée.**

Calculatrice TI
: 0 → S : For (K, 1, 50) : S + K^2 → S : End : Disp S

La somme est égale à 42 925.

4

1°) **Expliquer ce que produit l'algorithme suivant.**

Initialisations :
k prend la valeur 0
u prend la valeur 1

Traitement :
Tantque *u* > 0,1 **Faire**
    k prend la valeur *k* + 1
    u prend la valeur 0,5*u*
FinTantque

Sortie :
Afficher *u* et *k*

On reconnaître aisément que l'algorithme fait intervenir la suite (u_n) définie par son premier terme $u_0 = 1$ et la relation de récurrence $u_{n+1} = 0,5u_n$ (pour tout entier naturel *n*).

La suite (u_n) est la suite géométrique de premier terme 1 et de raison 0,5.

Donc $\forall\, n \in \mathbb{N} \quad u_n = 0,5^n$.

L'algorithme donne le plus petit entier k pour lequel $u_k \leqslant 0,1$ et la valeur de u_k correspondante.

2°) **Modifier l'algorithme pour qu'il donne le plus petit entier naturel k tel que $0,9^k < 0,001$.**

On remplace :

- l'instruction « Tantque $u > 0,1$ Faire » par l'instruction « Tantque $u \geqslant 0,001$ Faire »

et

- l'instruction « u prend la valeur 0,5u » par « u prend la valeur 0,9u ».

3°) **Le programmer et donner la valeur de k obtenue.**

Calculatrice TI
: 0 → K : 1 → U : While U ≥ 0,001 : K + 1 → K : 0,9∗U → U : End : Disp K, U

On trouve $k = 66$ et $u_k = 0,00955...$.