

CS110 – Introduction à l’algorithmique – Contrôle n° 1

Nom :*Correction*.....
Prénom :

Document autorisé : fiche de référence
Durée : 30 minutes
Barème indicatif sur 10 points

N’oubliez pas de donner une **spécification** claire et concise pour chaque algorithme ou procédure que vous écrivez. Prenez également soin d’utiliser correctement l’**indentation**.

3 points

Pour chacune des deux instructions suivantes, quels peuvent être les types des variables x y et i pour que l’instruction soit correcte ?

$x \leftarrow 3 \times i + 5$ $y \leftarrow i + 3 \times 5.2$ x et i réels ou x et i entiers ou x réel et i entier y réel, i réel ou entier
 $x \leftarrow 3 \times i + 5.0$ $y \leftarrow 5.2 \times i + 3$ x réel, i réel ou entier y réel, i réel ou entier

Quel est le type et la valeur de ces expressions ?

$4 + 2 \times 12 / 2$ réel, 16.0..... $x < 4$ et $x > 4$ incorrect, pb de priorité!.....
 $2.0 \bmod 3 \times 5$... incorrect, mod entre entiers! ... non faux ou vrai ... booléen, vrai ...

4 points

Rédigez un algorithme pour le jeu du “C+/C-”, où l’utilisateur doit découvrir un nombre secret. L’algorithme demande à l’utilisateur d’entrer un nombre et lui indique s’il est inférieur (“C+”) ou supérieur (“C-”) au nombre cherché, jusqu’à ce qu’il l’ait trouvé. Quand le nombre est trouvé, l’algorithme indique le nombre d’essais et se termine. La valeur du nombre secret sera affectée à une variable comme première instruction de l’algorithme (par exemple $\text{secret} \leftarrow 123$).

Comme souvent, il y a plusieurs formulations possibles. Voici une possibilité avec une boucle répéter.

```
algorithme c_plus_c_moins
(* joue au C+/C- *)
var secret:entier (* le nombre recherché *)
    prop:entier (* la proposition *)
    cpt:entier (* nbre d'essais *)
début
    secret ← 123
    cpt ← 0
    répéter
        écrire("Entrez proposition :")
        lire(prop)
        si prop>secret alors écrire("C-") fin si
        si prop<secret alors écrire("C+") fin si
        cpt ← cpt + 1
    jusqu'à prop=secret
    écrire("Trouvé après ", cpt, " essais")
fin
```

On souhaite apporter deux améliorations à notre algorithme :

1. on aimerait que le nombre secret ne soit pas codé "en dur" dans l'algorithme,
2. le nombre d'essais doit être limité à une valeur maximale. Si ce nombre est atteint sans que le secret ait été trouvé, l'algorithme se termine en indiquant à l'utilisateur qu'il a perdu.

Rédigez une procédure qui met en œuvre cette version du jeu du C+/C-, avec le secret et le nombre maximal d'essais passés en paramètre. Justifiez le type des paramètres et leur mode de passage, ainsi que les conditions dans lesquelles la boucle se termine.

Il s'agit de deux paramètres d'entrée (fournissent une donnée à la procédure), ils sont donc passés par valeur (absence du mot-clé var). La boucle doit se terminer si l'une des deux conditions suivantes se produit :

1. le secret a été trouvé,
2. le nombre maximal d'essais a été atteint.

La condition de terminaison est donc le "ou logique" de ces deux conditions (attention aux parenthèses !)

Une fois la boucle terminée, il faut tester pour quelle raison cela s'est produit. A-t-on trouvé le secret ou effectué le nombre maximal d'essais ? Le message à afficher en dépend. Attention au cas où l'on trouve le secret lors du dernier essai autorisé... ici les deux parties de la condition de terminaison peuvent devenir vraies en même temps.

```

procédure c_plus_c_moins(secret:entier; essais_max:entier)
(* jeu du C+/C- avec en paramètres secret et nbre max d'essais *)
var prop:entier
    cpt:entier
début
    cpt ← 0
    répéter
        écrire("Entrez proposition :")
        lire(prop)
        si prop>secret alors écrire("C-") fin si
        si prop<secret alors écrire("C+") fin si
        cpt ← cpt + 1
    jusqu'à (prop=secret) ou (cpt=essais_max)

    si prop=secret alors
        écrire("Trouvé après ", cpt, " essais")
    sinon
        écrire("Désolé, nombre maximal d'essais atteint")
    fin si
fin

```