

Programmation impérative et langage C

feuille de **TD n° 4** : Spécifications

L'analyse du problème à résoudre conduit à une formalisation appelée **spécification**. Il s'agit d'un triplet :

$$/* Q */ , prog_sol([E], [es], [s]) , /* R */$$

où :

- $/* Q */$: est un prédicat d'entrée contenant les informations disponibles pour résoudre le problème, hormis le type des données
- $prog_sol([E], [es], [s])$: est l'abstraction du programme exprimée par une action paramétrée par 3 listes : d'entrée E, d'entrée/sortie es ou de sortie s.
 - . E liste des valeurs en entrée
 - . s, es listes des variables contenant le résultat identifiées avant le développement et à partir du problème (Etape 1)
 - s : zones mémoires contenant le résultat
 - es : zones mémoires contenant des valeurs en entrée et modifiées dans le programme
- $/* R */$: est un prédicat de sortie qui spécifie les résultats attendus.

Conventions :

- On sous-entend le type des données
- On note en majuscule : les constantes et les variables mathématiques (qui sont utilisées dans la description de la propriétés : prédicats Q et R)
- On note en minuscule : les variables du programme

On pourra utiliser :

- le quantificateur universel : $\forall I : P_1 \rightarrow P_2$
- le quantificateur existentiel : $\exists I : P_1 \wedge P_2$
- le quantificateur numérique noté ν : $(\nu I : I \in E \wedge P(I))$ qui donne le nombre des I dans E vérifiant la propriété P .

Pour simplifier la spécification des vecteurs, on notera :

- $(T(I..J), <)$ ou $(T(I..J), \leq)$ pour exprimer la croissance entre les indices I et J
- $t = A$ pour exprimer que les 2 vecteurs contiennent des valeurs identiques
- $t = \text{permut}(A)$ pour exprimer que le vecteur t contient toutes les valeurs de A mais peut-être dans le désordre

◇ Exercice 1 : Ecrire la spécification d'un programme qui, en entrée, reçoit un entier N et calcule, dans une variable a , la valeur absolue de N .

◇ Exercice 2 : Ecrire la spécification d'un programme qui calcule, dans une variable a , la partie entière de la racine carrée d'un entier positif ou nul N .

◇ Exercice 3 : Ecrire la spécification d'un programme **echanger** qui effectue l'échange des valeurs entre deux variables x et y .

◇ Exercice 4 : A et t étant deux vecteurs indexés de 0 à 10, écrire des formules de la logique des prédicats exprimant les propriétés suivantes :

- $(A(I..J), <)$
- $t = A$
- $t = \text{permut}(A)$

◇ Exercice 5 :

1. Ecrire la spécification d'un programme qui "trie" les 3 réels x, y et z dans l'ordre croissant (Après exécution, x contiendra la plus petite des 3 valeurs initiales, z la plus grande, et y celle qui reste.)
2. Nous supposons maintenant que la seule opération autorisée pour modifier les variables est **echanger** définie à l'exercice précédent.

◇ Exercice 6 : Soit $N \geq 0$ et un tableau $B(0..N)$ d'entiers. Ecrire les spécifications de programmes qui déterminent :

1. la première position de la valeur maximale du tableau B .
2. la dernière position de la valeur maximale du tableau B .

◇ Exercice 7 : Soit $B(0..N)$ un tableau d'entiers croissants (non strictement) et X une valeur entière telle que $B(0) \leq X < B(N)$. Ecrire la spécification d'un programme qui détermine la position de la dernière occurrence du plus grand nombre du tableau inférieur ou égal à X .

◇ Exercice 8 : Ecrire la spécification d'un programme qui cherche la position de la dernière occurrence d'un tableau $Y(0..1)$ dans un tableau $X(0..N)$ en sachant que Y est présent dans X . (La position d'une occurrence de Y dans X est l'indice de la case de X de valeur $Y(0)$.)

◇ Exercice 9 : Ecrire la spécification d'un programme qui cherche un entier X dans un tableau d'entiers $B(0..N)$ dont les cases contiennent toutes des valeurs différentes. Si X est présente dans B , le programme retourne sa position sinon il retourne la valeur $N + 1$.

◇ Exercice 10 : Pour un tableau $B(0..N)$, on appelle sous-tableau tout tableau extrait $B(d, f)$ avec $0 \leq d \leq f \leq N$. On appelle plateau de B tout sous-tableau dont les cases contiennent toutes la même valeur.

Ecrire la spécification d'un programme qui détermine la position du premier plateau contenant la valeur maximale d'un tableau de $N + 1$ entiers.

◇ Exercice 11 : Ecrire la spécification d'un programme qui reçoit en entrée un tableau $B(0..N)$ de caractères et qui retourne le nombre d'occurrences de "L", "L" (doubles occurrences contiguës du caractère "L") qui se trouvent dans B , sachant que B contient au moins une telle occurrence.

1. Donner la plus petite valeur possible de N et le nombre maximum possible de couples " L ", " L " dans B .
2. Donner un exemple significatif de tableau B en précisant le nombre de couples " L ", " L ".
3. Ecrire la spécification de ce programme.

◇ Exercice 12 : Écrire la spécification d'un programme qui reçoit un tableau $T(0..N)$ de $N + 1 > 1$ entiers tous différents et renvoie la position des 2 plus grands éléments de ce tableau.

◇ Exercice 13 : Ecrire la spécification d'un programme qui inverse l'ordre des éléments dans un tableau de 50 entiers lus au clavier.

◇ Exercice 14 : Le problème du drapeau : A partir d'un tableau $T(0..N)$ contenant 3 couleurs mélangées : bleu, blanc et rouge, il s'agit de trier le drapeau de façon à obtenir d'abord tous les bleus, puis tous les blancs et enfin tous les rouges.

On suppose que les 3 couleurs sont présentes dans le tableau. L'algorithme implémenté ne doit pas faire appel à un autre tableau : toutes les opérations doivent se faire sur le tableau d'origine.

En sortie, on attend les résultats suivants :

- p_1 indice du premier des bleus
- p_2 indice du premier des blancs
- p_3 indice du premier des rouges

