TD 11
Recherche séquentielle

Algorithmique
Semestre 1

1 Sous-programme incideMinimum

1.1 En-tête du sous-programme

```
1  -- Recherche de facon sequentielle le rang du plus petit element d'un
       tableau tab (premiere occurence)
2  fonction indiceMin (entree tab <TabEntier>,
3            entree nbElem <Entier>)
4            retourne <Entier>;
```

Listing 1 – En-tête de incideMinimum

1.2 Corps du sous-programme

```
1  -- Recherche de facon sequentielle le rang du plus petit element d'un
       tableau tab (premiere occurence)
2  fonction indiceMin (entree tab <TabEntier>,
3            entree nbElem <Entier>)
4            retourne <Entier>
5  glossaire
6    i <Entier>; -- indice de parcours du tableau
7    iMin <Entier>; --rang du minimum
8
9  debut
10   iMin <- 1; --le plus petit est 1
11   i <- 2;
12
13   tantque i <= nbElem faire
14     si tab[i] < tab[iMin] alors
15       iMin <- i;
16     fin si;
17     i <- i + 1;
18   fin tantque;
19   retourner(iMin);
20 fin
```

Listing 2 – corps de indiceMinimum

2 Sous-programme rechercherOccurence

2.1 En-tête du sous-programme

```
1 -- Recherche s'il existe le rang de la premiere occurence d'un element x
     dans un tableau tab de n elements
2 procedure rechercherOccurence (entree tab <TabEntier>,
3                 entree n <Entier>,
4                 entree x <Entier>,
5                 sortie existe <Booleen>,
6                 sortie rang <Entier>);
```

Listing 3 – Entête de la procédure rechercherOccurence

2.2 Préconditions

```
1 -- Recherche s'il existe le rang de la premiere occurence d'un element x
     dans un tableau tab de n elements
2 -- necessite i <= n <= N
3 -- declanche rangInvalide
4 procedure rechercherOccurence (entree tab <TabEntier>,
5                 entree n <Entier>,
6                 entree x <Entier>,
7                 sortie existe <Booleen>,
8                 sortie rang <Entier>);
```

Listing 4 – Entête de la procédure rechercherOccurence avec les préconditions

2.3 Postconditions

a) Ne retourne pas faux si x non trouvé.
Ne garantit pas que ça soit la première occurence

b) Ne retourne pas faux mais garanti que ça soit la première occurence

c) Implique l'unciité de l'élément recherché

d) Post-condition

e) $tab[i] = x$ implique que tous les les éléménts soient égales à x

2.4 Corps du sous-programme

```
1 -- Recherche s'il existe le rang de la premiere occurence d'un element x
     dans un tableau tab de n elements
2 -- necessite i <= n <= N
3 -- declanche rangInvalide
4 -- entraine d)
5 procedure rechercherOccurence (entree tab <TabEntier>,
6                 entree n <Entier>,
7                 entree x <Entier>,
8                 sortie existe <Booleen>,
9                 sortie rang <Entier>)
10 --VERSION 1
11 glossaire
12   i <Entier>; --indice
13 debut
14   si n < 1 ou n > N alors
15     declancher trancheInvalide;
16   fin si;
```

```
17    i <- i;
18
19    tantque i <= n et tab[i] /= x faire
20      i <- i + 1;
21    fin tantque;
22    si i > n alors
23      existe <- FAUX;
24    sinon
25      existe <- VRAI;
26      rang <- i;
27    fin si;
28  fin
```

Listing 5 – Corps de la procédure rechercherOccurence (version 1)

```
1   -- Recherche s'il existe le rang de la premiere occurence d'un element x
        dans un tableau tab de n elements
2   -- necessite i <= n <= N
3   -- declanche rangInvalide
4   -- entraine d)
5   procedure rechercherOccurence (entree tab <TabEntier>,
6                 entree n <Entier>,
7                 entree x <Entier>,
8                 sortie existe <Booleen>,
9                 sortie rang <Entier>)
10  --VERSION 2
11  glossaire
12    i <Entier>; --indice
13    fini <Booleen>;
14  debut
15    si n < 1 ou n > N alors
16      declancher trancheInvalide;
17    fin si;
18    i <- i;
19    fini <- FAUX;
20
21    tantque non fini faire
22      si i > n alors
23        fini <- VRAI;
24        existe <- FAUX;
25      sinon
26        si tab[i] = x alors
27          fini <- VRAI;
28          existe <- VRAI;
29          rang <- i;
30        sinon
31          i <- i + 1:
32        fin si;
33      fin si;
34    fin tantque;
35  fin
```

Listing 6 – Corps de la procédure rechercherOccurence (version 2)

2.5 Exemple d'appel du sous-programme

2.5.1

C'est un tableau

2.5.2

```
importer entreeSortie;

type TabEntier : tableau [1 a 50] de <Entier>;

procedure lireTableauValeur(tabValeurs, nbValeurs); --cf TD9

procedure rechercherOccurence (entree tab <TabEntier>,
               entree n <Entier>,
               entree x <Entier>,
               sortie existe <Booleen>,
               sortie rang <Entier>)
glossaire
  i <Entier>; --indice
debut
  si n < 1 ou n > N alors
    declancher(trancheInvalide);
  fin si;
  i <- i;

  tantque i <= n et tab[i] /= x faire
    i <- i + 1;
  fin tantque;
  si i > n alors
    existe <- FAUX;
  sinon
    existe <- VRAI;
    rang <- i;
  fin si;
fin

programme rechercher

glossaire
  v <Entier>;
  tdeValeurs <TabEntier>;
  nbValeurs <Entier>;
  trouve <Booleen>;
  rang <Entier>;

debut
  lireTableauValeur(tabValeurs, nbValeurs);
  lire(v);
  rechercherOccurence(tabValeur, nbValeur, v, trouve, rang);

  si trouve alors
    ecrire(rang);
  sinon
    ecrire("La valeur na pas ete trouve");
  fin si;

  traite-exception
    lorsque debordement faire
      ecrire("Trop de valeurs a lire");
    fin lorsque;

    lorsque trancheInvalide faire
```

```
57          ecrire("Tranche invalide de l array");
58        fin lorsque;
59    fin
```

Listing 7 – Programme