R101 – TD4

Exercice 1 : saisir un entier dont la valeur doit être supérieure à un paramètre

Ecrire la fonction `saisirEntierSupX ()` qui renvoie un entier naturel saisi au clavier, supérieur à une valeur entière `X` passée en troisième paramètre - les deux premiers paramètres sont les chaînes de caractères représentant l'invite de la saisie et le message d'erreur si la valeur saisie n'est pas supérieure à `X`.

Exercice 2 : trouver une valeur présente dans un tableau

Ecrire la fonction `find ()` de profil :

```
fonction find (val : in entier , tabInt : in tableau_de entier)
renvoie entier_naturel;
```

Cette fonction renvoie l'indice de la première occurrence de `val` dans `tabInt`. Pour cet exercice, on suppose que `Val` est présent dans `tabInt`.

En utilisant les sous-programmes que vous avez déjà vus, écrire un algorithme permettant de tester la fonction `find ()`. Faire la trace de votre algorithme.

Exercice 3 : trouver une valeur qui peut-être présente dans un tableau

Ecrire la fonction `find ()` de profil identique à celui de l'exercice 2. Pour cet exercice, on ne sait pas si `Val` est présente dans `tabInt`. Si `val` est présente, on renvoie l'indice de la première occurrence. Dans le cas contraire, on renvoie la taille de `tabInt`.

En utilisant les sous-programmes que vous avez déjà vus, écrire un algorithme permettant de tester la fonction `find ()`. Faire la trace de votre algorithme.

Exercice 4 : trouver une valeur qui peut-être présente dans un tableau (sentinelle)

Reprendre l'exercice 3 en utilisant la technique des sentinelles.

Exercice 5 : générer un tableau d'entiers tous différents (V1)

On souhaite écrire le sous-programme `genereTabInt ()`. Celui-ci doit générer un tableau de `N` entiers aléatoires distincts dans [`1..M`] (`M > N`).Dans cet exercice, à chaque itération, on va remplir un élément du tableau. L'algorithme de haut niveau est le suivant :

```
...
pour (nb variant_de 0 a N - 1)
faire
    genererEntierAleatoireJusquACeQuIlNeSoitPasDejaStocke;
    stockerLEntierAleatoireDansTabInt;
ffaire
...
```

Ecrire l'algorithme qui teste `genereTabInt ()`.

Exercice 6 : générer un tableau d'entiers tous différents (V2)

On souhaite écrire le sous-programme `genereTabInt ()`. Celui-ci doit générer un tableau de `N` entiers aléatoires distincts dans [`1..M`] (`M > N`).Dans cet exercice, à chaque itération, on va générer un nombre aléatoire, que l'on va ranger ou non dans le tableau. L'algorithme de haut niveau est le suivant :

```
...
tant_que (Nb < N)
faire
    genererEntierAleatoire;
    si (lEntierAleatoireNAppartientPasAuTableau)
        stockerLEntierAleatoireDansTabInt;
    fsi
ffaire
...
```

Ecrire l'algorithme qui teste `genereTabInt ()`.

Exercice 7 : générer un tableau d'entiers tous différents (V3)

On souhaite écrire le sous-programme `GenereTabInt ()`. Celui-ci doit générer un tableau de `N` entiers aléatoires distincts dans [`1..M`] (`M vaut N`).