Exercice 1 — Un

Exercice 2 — Deux

Exercice 3 — Date A Raccoon

On considère une application *Date A Raccoon !*, qui permet de faire des rencontres entre ratons laveurs, et qui possède des enregistrements de membres, au format suivant :

Types
- raccoon_t : enregistrement
  - id : entier
  - taille : réel `# en cm`
  - pays : chaîne de caractères
  - albinos : booléen
  - rage : booléen

Q 1) *(X pt)* En utilisant le type `raccoon_t`, écrire un algorithme qui (1) demande la saisie de **maximum** 42 ratons laveurs (sans possibilité de dépasser) et les stocke tous dans un tableau, puis (2) affiche le nombre de ratons laveurs qui ont effectivement été saisis.

Vous devez renseigner le champ `id` avec un numéro permettant d'identifier le raton laveur de façon unique. On considère que la personne qui fait la saisie répondra obligatoirement soit *"oui"*, soit *"non"*, aux questions fermées.

Correction :

Variables
- *continuer, ouiounon : chaînes de caractères*
- *ratons : tableau de raccoon_t [42]*
- *i : entier*

Début
- $i \leftarrow 0$
- *continuer* $\leftarrow$ "*oui*"
- `# On pourrait aussi demander de saisir au préalable le nombre de ratons laveurs qui vont être saisis, le vérifier (0<n<43), et ensuite utiliser un Pour.`
- **Tant que** $i < 42$ **et** *continuer* = "*oui*" **Faire**
  - *ratons*[*i*].*id* $\leftarrow$ *id*
  - *ratons*[*i*].*taille* $\leftarrow$ *saisir("Entrer une taille (en cm) :")*
  - *ratons*[*i*].*pays* $\leftarrow$ *saisir("Entrer un pays :")*
  - *ratons*[*i*].*albinos* $\leftarrow$**Faux**
  - *ouiounon* $\leftarrow$ *saisir("Albinos (oui/non) ?")*
  - **Si** *ouiounon* = "*oui*" **Alors**
    - *ratons*[*i*].*albinos* $\leftarrow$**Vrai**
  - **Finsi**
  - *ratons*[*i*].*rage* $\leftarrow$**Faux**
  - *ouiounon* $\leftarrow$ *saisir("Rageux (oui/non) ?")*
  - **Si** *ouiounon* = "*oui*" **Alors**
    - *ratons*[*i*].*rage* $\leftarrow$**Vrai**
  - **Finsi**
  - *continuer* $\leftarrow$ *saisir("Continuer (oui/non) ?")*
  - $i \leftarrow i + 1$
- **Fintantque**
- *afficher("Ratons laveurs saisis :" + i)*

Fin

Q 2) *(X pt)* Écrire une fonction `get_score` qui prend en paramètre un raton laveur `raccoon_t`, et qui retourne son score d'attractivité. En partant d'un score initial de zéro, on ajoute 1 point si le raton laveur est au-dessus de la taille moyenne des ratons laveurs (80 cm), on ajoute 1 point par critère de rareté (les albinos et les japonais sont rares), et on retire 3 points s'il a la rage.

Correction :

Fonction `get_score`(*raton : raccoon_t*) *: entier*
- **Variables**
  - *score : entier*
- **Début**
  - *score* $\leftarrow$ *0*
  - **Si** *raton.taille* > 80 **Alors**
    - *score* $\leftarrow$ *score* $+ 1$
  - **Finsi**
  - **Si** *raton.albinos* **Alors**
    - *score* $\leftarrow$ *score* $+ 1$
  - **Finsi**
  - **Si** *raton.pays* $=$ "*japon*" **Alors**
    - *score* $\leftarrow$ *score* $+ 1$
  - **Finsi**
  - **Si non** *raton.rage* **Alors**
    - *score* $\leftarrow$ *score* $- 3$
  - **Finsi**
  - *retourner score*
- **Fin**

Finfonction

Q 3) *(X pt)* Écrire une fonction qui prend en paramètre le tableau de ratons laveurs ainsi que son nombre de valeurs effectives, et qui retourne le nombre de couples possibles. On considère que deux ratons laveurs pourraient former un couple dès lors que leurs scores d'attractivité respectifs (on compte sur l'existence de la fonction précédente) ne sont pas différents de plus de 2 points.

Attention à ne pas compter deux fois les mêmes couples. La fonction `abs(entier) : entier` retourne la valeur absolue d'un entier donné.

Correction :

Fonction `get_nbmatches`(*ratons : tableau de raccoon_t [42], n_ratons : entier*) *: entier*
- **Variables**
  - *i, j, n : entiers*
- **Début**
  - *n* $\leftarrow$ *0*
  - **Pour** *i allant de* 0 *à* $n\_ratons - 1$ **Faire**
    - **Pour** *j allant de* 0 *à* $i - 1$ **Faire**
      - **Si** $abs(get\_score(ratons[i]) - get\_score(ratons[j])) \leqslant 2$ **Alors**
        - *n* $\leftarrow$ *n + 1*
      - **Finsi**
    - **Finpour**
  - **Finpour**
  - *retourner n*
  - `# Si la seconde boucle parcourais tout le tableau : soustraction du nombre total de ratons laveurs pour éliminer ceux qui ont matché avec eux-mêmes, puis division par 2 pour éliminer les doublons.`
  - `# retourner` $(n - n\_ratons)/2$
- **Fin**

Finfonction

Q 4) *(X pt) Outre-Atlantique, le raton laveur est parfois apprécié au barbecue, accompagné de patates douces. La base de données de l'application* Date A Raccoon! *(le tableau) ayant fuité sur Internet, des humains s'en sont emparés pour savoir où chasser le raton laveur (une autre base de données leur permet d'associer les identifiants uniques à des coordonnées géographiques).*

Écrire une fonction qui prend en paramètre le tableau de ratons laveurs ainsi que son nombre de valeurs effectives, et retourne un tableau uniquement constitué **des identifiants** (`id`) des ratons laveurs qui sont comestibles (i.e. qui n'ont pas la rage). Un tableau étant toujours accompagné du nombre effectif de valeurs qu'il contient, ce nombre doit également être retourné par la fonction. Pour pouvoir retourner 2 variables à la fois, vous devrez au préalable créer un nouveau type d'enregistrement (écrivez simplement sa définition sur votre copie, avant d'écrire la fonction).

Correction :

```
Types
    ids_t : enregistrement
        list : tableau d'entiers [42]
        n : entier
Fonction get_safe(ratons : tableau de raccoon_t [42], n_ratons : entier) : ids_t
    Variables
        i : entier
        ids : ids_t
    Début
        ids.n ← 0
        Pour i allant de 0 à n_ratons − 1 Faire
            Si non ratons[i].rage Alors
                ids.list[n] ← ratons[i].id
                ids.n ← ids.n + 1
            Finsi
        Finpour
        retourner ids
    Fin
Finfonction
```