



Université
de Limoges

FACULTÉ
DES SCIENCES
ET TECHNIQUES

Tutorat d'Informatique

Devoir Blanc Mars 2013

Durée : 1 heure 30 Cours, documents et calculatrice non autorisés

NB : une rédaction soignée des algorithmes est exigée.

Exo 1 (7/20) L'ONU (Organisation des Nations Unies) a décidé d'organiser son système d'informations de la façon suivante :

A) Pour chaque pays, on souhaite avoir les informations suivantes :

- ⌚ nom du pays
- ⌚ nombre d'habitants
- ⌚ superficie (en km²)

B) On souhaite stocker aussi des informations sur les pays de la planète que l'ONU reconnaît :

- ⌚ tableau (vecteur) des 193 pays membres
- ⌚ tableau des 3 pays non membres

1.1 Définir les structures nécessaires pour stocker les informations décrites en A et B.

1.2 Écrire deux fonctions pour saisir les informations sur un pays participants (`saisirPays()`) ainsi que les informations sur les pays membres de l'organisation (`saisirPaysReconnus()`).

Remarque : pensez à faire les contrôles nécessaires lors de la saisie

Exo 2 (13/20)

2.1 Définir un enregistrement permettant de représenter des « Permis de conduire à point ». Les rubriques seront le numéro de permis qui est un identifiant unique (entier), la date d'obtention (un enregistrement de type Date), le nom du possesseur du permis (chaîne de 20 caractères), le prénom du possesseur du permis, la date de Naissance (un enregistrement de type Date) et le nombre de points restants (entier).

La date sera un enregistrement de type Date contenant les rubriques jour (entier), mois (entier) et année (entier).

2.2 Écrire une procédure *AjouterPermis* qui permet aux forces de l'ordre d'ajouter dans le fichier les informations sur un nouveau détenteur du permis. Cette procédure prend en entrée le nom du fichier (paramètre de type chaîne). Le numéro de permis sera calculé par la procédure comme étant le numéro de

la position à laquelle le permis sera enregistré (si le fichier est vide, le premier permis qui sera saisi aura le numéro 1 – on suppose que même si le fichier est vide, il existe déjà). Les autres informations seront saisies par les forces de l'ordre. Il serait pertinent de faire un sous-programme pour la saisie d'une date. Le nombre de points assignés à un conducteur dépendra de son âge à la date d'obtention. S'il a moins de 20 ans alors il aura 6 points, sinon il en aura 12.

2.3 Écrire une procédure *RetirerPoints* de mise à jour du nombre de points restants pour une personne. Elle prend en entrée le nom du fichier (paramètre de type chaîne) ainsi que les nom et prénom (deux paramètres de type chaîne) de la personne à laquelle on veut retirer des points et le nombre de points à lui retirer (paramètre de type entier). On considérera que le fichier ne contient pas deux personnes avec les mêmes noms et prénoms. S'il reste moins de points à la personne que le nombre de points à lui retirer, il lui en restera 0 au final.

2.4 Écrire l'algorithme principal qui appelle cette dernière procédure pour ouvrir le fichier «FichierNationalDesPermis.dat » et modifie le nombre de points restants pour un individu sur la base des informations saisies par un membre des forces de l'ordre.

Correction

Exo 1

```
structure Pays
{
    nom : chaine de 20 caractères
    nombre_habitants : entier
    superficie : entier
}

structure PaysReconnus
{
    pays_membres : vecteur de 193 Pays
    pays_non_membres : vecteur de 3 Pays
}

fonction saisirPays() : Pays
{
    variable p : Pays

    Afficher("Saisir le nom du pays")
    Saisir(p.nom)

    Répéter
    {
        Afficher("Saisir le nombre d'habitants : ")
        Saisir(p.nombre_habitants)
    } jusqu'à (p.nombre_habitants >=0)

    Répéter
    {
        Afficher("Saisir la superficie en km^2 : ")
        Saisir(p.superficie)
    } jusqu'à (p.superficie>=0)

    retourne p
}

fonction saisirPaysReconnus() : PaysReconnus
{
    constante entier NB_membres <-- 193
    constante entier NB_non_membre <-- 3
    variable PR : PaysReconnus
        i : entier

    Pour i de 1 à NB_membres
    {
        PR.membres[i-1] <-- saisirPays()
    }

    Pour i de 1 à NB_non_membres
    {
```

```

        PR.non_membres[i-1] <-- saisirPays()
    }

    retourne PR
}

```

Exo 2

```

structure Date
{
    j : entier
    m : entier
    a : entier
}

structure Permis
{
    identifiant : entier
    date_obtention : Date
    nom : chaîne de 20 caractères
    prenom : chaîne de 20 caractères
    date_naissance : Date
    nbPointsRestants : entier
}

fonction saisirDate(E m : chaîne de caractères):Date
{
    variable d: Date
    Afficher(m)
    Saisir(d.j, d.m, d.a)
    retourne d
}

procedure AjouterPermis(E nom_f:chaîne de caractères)
{
    variable f:Fichier
        p : Permis
        numero_permis : entier

    numero_permis <-- 1

    Désigner(nom_f, f)
    Ouvrir(f,lecture/écriture)
    Decaler(f,Début_fichier, 0)

    Tant que (Non Fin(f))
    {
        Decaler(f,Position_courante, 1)
        numero_permis <-- numero_permis + 1
    }

    p.identifiant <-- numero_permis
}

```

```

    p.date_obtention <-- saisirDate("Veuillez entrer le jour, le mois et l'année
d'obtention du permis :")
    Afficher("Veuillez entrer le nom et le prénom de la personne :")
    Saisir(p.nom, p.prenom)
    p.date_naissance <-- saisirDate("Veuillez entrer le jour, le mois et l'année de
naissance de la personne :")

    Si (p.date_naissance.a+20<p.date_obtention.a) alors
        p.nbPointsRestants <-- 6
    sinon
        p.nbPointsRestants <-- 12
    Decaler(f,Position_courante, -1)
    Ecrire(f,p)
    Fermer(f)
}

```

```

procedure RetirerPoint(E nom_f:chaîne de caractères,
                        E nom:chaîne de caractères,
                        E prenom:chaîne de caractères)
{
    variable f:Fichier
        TROUVE : logique
        p : Permis
        nbpointsAretirer : entier

    TROUVE <-- FAUX

    Désigner(nom_f, f)
    Ouvrir(f,lecture/écriture)
    Decaler(f,Début_fichier, 0)

    Tant que (Non Fin(f) ET non (TROUVE))
    {
        Lire(f,p)
        Si ((p.nom = nom) ET (p.prenom = prenom)) alors
        {
            TROUVE <-- VRAI
        }
    }

    Si (TROUVE) alors
    {
        Afficher("Entrez le nombre de points à retirer à ", p.nom, " ",
p.prenom, " : ")
        Saisir(nbpointsAretirer)
        Si (p.nbPointsRestants <= nbpointsAretirer) alors
            p.nbPointsRestants <-- 0
        sinon
            p.nbPointsRestants <-- p.nbPointsRestants - nbpointsAretirer
        Decaler(f,Position_courante, -1)
        Ecrire(f,p)
    }
}

```

```
        Fermer(f)
    }
```

Algorithme Principal

```
{
    constante chaîne de caractères fichier <-- "FichierNationalDesPermis.dat"
    variable nom, prenom : chaîne de caractères

    Afficher("Saisir le nom et le prénom de la personne à qui vous voulez retirer
des points")
    Saisir(nom, prenom)

    ModificationProgrammation(fichier, nom, prenom)
}
```