Travaux Dirigés 1

Actions paramétrées (Procédures et Fonctions)

Rappel de cours:

- **Déclaration et définition :** Pour les actions paramétrées, on suppose que déclaration et définition sont confondues. Ces déclarations sont placées avant l'algorithme principal (voir exercice 1).
- **Syntaxe :** (Exemples)
  - Une action paramétrée de type procédure avec ***par1*** entier transmis en entrée et ***par2*** entier transmis en sortie :
    Action *nomAction* (E/ ***par1*** :entier; S/ ***par2*** :entier) ;
    Debut … Fin ;
    (au lieu du mot clé **Action** on peut utiliser **Procedure**)
  - Une action paramétrée de type fonction retournant un entier avec ***par1*** entier transmis par valeur et ***par2*** entier transmis par variable :
    Fonction *nomFct* (E/ ***par1*** :entier; Ref/ ***par2*** :entier) : **entier *;***
    Debut … retourne(expression) ; Fin ;
- **Modes de transmission :** Quatre modes de transmission de paramètres : **Entrée** (ou par valeur), **Sortie**, **Mixte** et par **Référence** (ou par variable) spécifiés par E/, S/, ES/ et Ref/ respectivement. Pour expliquer le déroulement avec un schéma de RAM, on supposera qu'il y a création de variables locales de substitution dans l'espace pile pour les 3 premiers modes (E/, S/ et ES/).
  On suppose qu'un tableau peut être transmis par valeur.
- **Visibilité :**
  Les variables déclarées dans l'algorithme principal sont globales. Une action paramétrée peut appeler une autre si la définition de l'appelée est réalisée avant celle de l'appelante. Dans ce cas, les variables de cette dernière sont vues dans l'appelée.

Exercice 1 :

1. Ecrire une action paramétrée `MAXIMUM()` qui détermine le maximum de 2 entiers.
2. En utilisant `MAXIMUM()`, écrire un algorithme qui lit trois entiers et affiche leur maximum.

Exercice 2 :

1. Ecrire une action paramétrée `DIVISE()` qui vérifie si un nombre entier A divise un nombre entier B.

Soit tab un vecteur d'entiers de taille N<=10.

2. En utilisant `DIVISE()`, écrire un algorithme qui permet d'afficher tous les éléments du vecteur tab divisibles par une valeur val donnée.

Exercice 3 :

1. Ecrire une action paramétrée `VABS()` qui, étant donné un entier, calcule sa valeur absolue.

2. Soit T un vecteur de N valeurs entières quelconques (N<=10). Ecrire un algorithme qui détermine la valeur maximale de T, puis en utilisant la fonction précédente affiche la somme des valeurs absolues de tous les nombres de T (sauf la valeur maximale).

Exemple: Pour T=

6	-15	30	-9	140	28	-1

Somme = 6 + 15 + 30 + 9 + 28 + 1

Exercice 4 :

1. Ecrire une action paramétrée `FACT()` qui, étant donné un entier A, calcule sa factorielle.
2. Ecrire une action paramétrée `PUISS()` qui, étant donné un entier A, un entier N, calcule la puissance A^n
3. En utilisant ces deux actions paramétrées, écrire un algorithme qui calcule la valeur de la somme S, tel que : $S = X/1 ! + X^2/2 ! + X^3/3 ! + ... + X^n/n !$

Exercice 5 :

Ecrire une action paramétrée qui permet de vérifier si un mot de longueur L est palindrome. (un mot palindrome est un mot qui se lit indifféremment de gauche à droite ou de droite à gauche. Ex : le mot « radar » est un palindrome)

Exercice 6:

1. Écrire deux actions paramétrées à un argument entier et une valeur de retour entière permettant de préciser si l'argument reçu est multiple de 2 (pour la première fonction) ou multiple de 3 (pour la seconde fonction).
2. Utiliser ces deux actions paramétrées dans un algorithme principal qui lit un nombre entier et qui précise s'il est pair, multiple de 3 et/ou multiple de 6, comme dans cet exemple :

Exécution 1 :

```
Donnez un entier : 9
Il est multiple de 3
```

Exécution 2 :

```
Donnez un entier : 12
Il est pair
Il est multiple de 3
IL est divisible par 6
```

Exercice 7:

Ecrire une Action paramétrée `LongChaine()` qui renvoie la longueur d'une chaîne de caractères.

Exercice 8 :

Écrire une fonction récursive qui calcule la somme des n premiers nombres entiers positifs. Décrire son déroulement avec un schéma de RAM pour n=4.

Exercice 9 :

Écrire une procédure récursive qui permet de trouver le mot miroir d'un mot donné. Ex : le mot miroir de PAVILLON est NOLLIVAP.

Exercices Supplémentaires (facultatifs)

Examen 2011/12

On veut écrire des actions paramétrées permettant d'effectuer des opérations concernant des matrices carrées et des vecteurs binaires (contenant des 0 et 1). On définit le poids d'un vecteur ou d'une matrice comme étant le nombre de 1 qu'il (ou elle) contient.

Ecrire une action paramétrée ***estVecteurBinaire*** qui prend en argument un entier n et un tableau d'entiers V (de taille n) et teste s'il s'agit d'un vecteur binaire (si V ne contient que des 0 ou des 1).

1. Ecrire une action paramétrée ***remplaceBitVecteur*** qui prend en argument un entier n et un vecteur binaire V (de taille n) et qui remplace toutes les occurrences de 1 avec 0 et vice-versa.
2. Ecrire une action paramétrée ***premierDernierBit1*** qui prend en argument un entier n, un vecteur binaire V (de taille n). Cette action paramétrée doit calculer les deux positions du premier et du dernier 1 contenu dans le vecteur. A la sortie, ces deux positions doivent être placées dans deux variables globales ***first*** et ***last***.
3. Ecrire une action paramétrée ***poidsVecteur*** qui prend en argument un entier n et un vecteur binaire V (de taille n) et renvoie le poids de V (un entier compris entre 0 et n) (vous devez utiliser l'action paramétrée ***premierDernierBit1***).
4. Ecrire le programme principale (main) qui :
   - 4.1. Demande de remplir une matrice 5x5.
   - 4.2. Vérifie que la matrice est binaire (pour cela mettez chaque ligne de la matrice dans un vecteur ***MatLigne***).

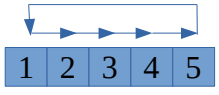
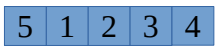
   Si la matrice est binaire :

   - 4.3. Calcule les poids de chaque ligne de la matrice. Ces poids seront stockés dans le tableau ***PoidsLignes***.
   - 4.4. Remplace les 1 de la dernière ligne de la matrice par 0 et vice-versa.
   - 4.5. Affiche la matrice et le tableau ***PoidsLignes***.

Examen 2014/2015

Écrire une action paramétrée **récursive** qui réalise un décalage circulaire à droite (i.e. chaque élément est déplacé vers la case suivante et on suppose que le suivant du dernier est le premier) des éléments d'un tableau d'entiers. L'action paramétrée ne doit pas utiliser un 2ieme tableau.

Exemples :

1. Le décalage du tableau Vect=  donne  .

2. Le décalage du tableau Vect=  donne  .