
TP découverte - Langage ADA

Exercice 1 Premiers programmes

Q1 . Ecrire un programme qui calcule le carré d'un nombre lu au clavier.

Q2 . Ecrire un programme qui affiche la table de multiplication d'un nombre lu au clavier. Essayez de bien aligner les colonnes si possible !

Exercice 2 Arguments d'un programme

Ecrire un paquetage packtest qui contient les fonctions et procédures suivantes :

- la procédure NomProg affiche le nom du programme en cours d'exécution
- la fonction NbrArg retourne le nombre d'arguments passés au programme en cours
- la fonction Arg prend en paramètre un entier N de type Natural et retourne l'argument numero N passé au programme en cours
- la procédure AffArg prend en paramètre un entier N de type Natural et affiche le numéro N de l'argument puis la chaîne de caractère représentant l'argument en une seule ligne de code

Vous testerez ce paquetage grâce à un programme principal qui affichera d'abord son nom puis la liste de ses arguments.

Exercice 3 Portée et visibilité d'une déclaration

Essayez de prévoir et d'expliquer le résultat de l'exécution de la procédure P puis programmez la pour vérifier votre prévision.

```
with Ada.Integer_Text_IO ;  
use Ada.Integer_Text_IO ;  
procedure P is  
  x : Integer ;  
  procedure Q is  
  begin  
    put(x) ;  
  end Q ;  
  procedure R is  
    x : Integer ;  
    procedure S is  
    begin  
      put(x) ;  
    end S ;  
  begin  
    x := 12 ;  
    Q ;  
    S ;  
    x := 78 ;  
    S ;  
    Q ;  
  end R ;  
begin
```

```

    x := 40 ;
    R ;
    x := 35 ;
    R ;
end P;

```

Exercice 4 Debuggage

Debuggez le programme `toto.adb` et expliquez chacune des erreurs commises. Une fois correct il doit afficher dans l'ordre, le carré de 2, le cube de 2, puis la valeur de 2^4 , la valeur de 5^4 , la valeur de 3^6 puis de 0^6 et enfin $4!$.

Exercice 5 A propos des chaînes de caractères

Q1 . Quelles sont les positions des caractères '0', '1', '2' ... '9' dans le type `character` ada ?

Q2 . Ecrire une fonction qui retourne le nombre de mots contenus dans la chaîne de caractères qui lui est passée en paramètre. Un mot est une succession de caractères différents des caractères espace et de ponctuation, et chaque chiffre compte pour un mot.

Q3 . Ecrire une fonction qui teste si un mot est un palindrome.

Q4 . Ecrire la fonction récursive sans paramètre `line` qui lit au clavier une chaîne de caractères de longueur quelconque terminée par un point et retourne cette chaîne de caractère.

Déclarez une variable de type `string` et utilisez la fonction `line` pour y stocker la chaîne de caractères lue au clavier.

Que se passe-t-il ? Trouvez une solution pour remédier au problème.

Remarque : la procédure `get_line(s : out STRING, nb : out NATURAL)` permet la saisie d'une chaîne de caractères. Comme la chaîne `s` doit être déclarée avant l'appel, celle-ci doit nécessairement être contrainte.

Exercice 6 A propos des tableaux

Q1 . Ecrire une procédure qui réalise la fusion de deux tableaux d'entiers triés (on intercale les valeurs pour obtenir un tableau de la taille la somme des longueurs des deux tableaux qui contient toutes les valeurs des deux tableaux et ce de manière triée).

Q2 . Ecrire une procédure qui calcule et affiche le nombre d'occurrences de chaque lettre dans une chaîne de caractères.

Q3 . Ecrire une procédure qui calcul la somme des éléments d'un tableau d'entiers à deux dimensions. Il faudra déclarer un type de tableaux non contraint.

Q4 . Ecrire une procédure qui calcul le produit des éléments de la diagonale d'un tableau d'entiers à deux dimensions.

Exercice 7 Jeu de hasard

Tirer un nombre aléatoire en ADA sur un type discret se fait comme ceci :

```

with ada.text_io, ada.integer_text_io, ada.numerics.discrete_random;
use ada.text_io, ada.integer_text_io;

```

```

procedure hasard is

```

```

-- crée une instance d'un paquetage de nombres aléatoires sur les INTEGER
package Random_Integer is new Ada.Numerics.Discrete_Random(INTEGER);
use Random_Integer;

seed : GENERATOR; -- la graine qui permet d'initialiser les tirages

begin
  reset(seed); -- creation de la graine
  for i in 1..10 loop
    put(random(seed));
  end loop;
end hasard;

```

Q1 . Créer une procédure qui demande un nombre n à l'utilisateur et qui crée un tableau de longueur n initialisé aléatoirement par des entiers.

Q2 . Modifiez la procédure pour que le tableau soit rempli uniquement de nombres entre 1 et n .

Q3 . Modifiez la procédure pour que le tableau ne contienne qu'une seule fois chaque nombre entre 1 et n .