

Les enregistrements

Exercices

Nombres complexes

Exercice 1 : Représentation des nombres complexes

Un nombre complexe z est entièrement défini par ses parties réelle a et imaginaire b .

$$z = a + ib$$

Déclarez en PASCAL un type COMPLEXE.

Exercice 2 : Attributs d'un nombre complexe

Réalisez les fonctions donnant les attributs d'un nombre complexe (partie réelle, partie imaginaire, module, argument)

Exercice 3 : Arithmétique sur les complexes

Réalisez les fonctions nécessaires à l'arithmétique sur les complexes (addition, soustraction, multiplication, division, inverse, multiplication par un réel, conjugué).

Exercice 4 : Égalité de complexes

Réalisez un prédicat qui teste l'égalité de deux nombres complexes.

Exercice 5 : Entrée/sortie

Réalisez deux procédures nommée `lireComplexe` et `afficherComplexe` chargée des opérations d'entrée/sortie des objets de type COMPLEXE.

Exercice 6 : Une suite de nombres complexes

On considère la suite $(z_n)_{n \in \mathbb{N}}$ de nombres complexes définie par son premier terme

$$z_0 = 0$$

et la relation de récurrence

$$z_{n+1} = z_n^2 + c$$

où c est une constante complexe.

Q 1 . Réalisez une fonction nommée f qui renvoie le nombre complexe $f(z, c) = z^2 + c$.

Q 2 . Quelle instruction permet de calculer le terme z_n , n étant un entier naturel donné, la constante c étant donnée.

Q 3 . On peut montrer que selon la constante c la suite $(z_n)_{n \in \mathbb{N}}$ est

– soit bornée, et dans ce cas

$$\forall n \in \mathbb{N} \quad |z_n| < 2$$

– soit non bornée, et dans ce cas

$$\exists n \in \mathbb{N} \quad |z_n| \geq 2$$

Q 3.1. Est-il facile/possible de programmer le prédicat

```
// suitebornée(c) ⇔ la suite définie par c est bornée
function suitebornée(c : COMPLEXE) : BOOLEAN;
```

Q 3.2. Programmez la fonction

```
// bornefranchie(c, max) = -1 si ∀n ∈ [0, max] ||z_n| < 2
// bornefranchie(c, max) = le plus petit entier n ∈ [0, max] tq |z_n| ≥ 2
function bornefranchie(c : COMPLEXE; max : CARDINAL) : INTEGER;
```

Les dates

On rappelle la déclaration du type de données permettant de manipuler des dates vue en cours

```
type
  JOUR = (LUNDI, MARDI, MERCREDI, JEUDI,
          VENDREDI, SAMEDI, DIMANCHE);
  QUANTIEME = 1..31;
  MOIS = (JANVIER, FEVRIER, MARS, AVRIL,
          MAI, JUIN, JUILLET, AOUT,
          SEPTEMBRE, OCTOBRE, NOVEMBRE, DECEMBRE);
  ANNEE = CARDINAL;
  DATE = record
    jour      : JOUR;
    quantieme : QUANTIEME;
    mois      : MOIS;
    annee     : ANNEE;
  end {DATE};
```

Exercice 7 : Pâques

La date de Pâques correspond au 1er dimanche après la première pleine lune qui suit l'équinoxe de printemps. Soient les données suivantes :

$$\begin{aligned}a &= \text{année modulo } 19 \\b &= \text{année modulo } 4 \\c &= \text{année modulo } 7 \\d &= (19 * a + 24) \bmod 30 \\e &= (2 * b + 4 * c + 6 * d + 5) \bmod 7\end{aligned}$$

alors

$$n = 22 + d + e$$

est le numéro du jour à partir du 1er mars correspondant au dimanche de Paques.

Réalisez une fonction qui calcule la date de Pâques à partir de l'année.

Exercice 8 : Le nom du jour

Utilisation de la formule de Zeller pour le calcul du numéro du jour (0 pour dimanche, 1 pour lundi, ...) à partir du triplet d'entiers (q, m, a) Si $m \geq 3$ utiliser $m - 2$ au lieu de m sinon utiliser $m + 10$ et $a - 1$ au lieu de a Soient s et u les quotient et reste de la division de a par 100 Soit

$$f = q + u - 2s + \frac{u}{4} + \frac{s}{4} + \frac{26m - 2}{10}$$

Le reste de la division de f par 7 donne le numéro du jour.

Écrire une fonction en PASCAL permettant de calculer le nom du jour en clair (dimanche, lundi, ...)

Exercice 9 : Constructeur de date

Réalisez une fonction qui à partir de ses trois paramètres de type QUANTIEME, MOIS et ANNEE retourne une date. Cette fonction aura en particulier la charge de calculer le nom du jour.

Exercice 10 : Lendemain et veille

Écrivez les fonctions `lendemain` et `veille` qui retournent les dates du lendemain et de la veille d'une date passée en paramètre.

Exercice 11 : Date valide ?

Lorsqu'un programmeur définit une constante de type `DATE`, il peut commettre des erreurs comme par exemple

```

const AUJOURDHUI : DATE = (jour      : LUNDI;
                             quantieme: 1;
                             mois      : MARS;
                             annee     : 2005););

```

or le 1er mars 2005 est un mardi.

Q 1 . Quelles sont les autres sources d'erreurs possibles ?

Q 2 . Réaliser un prédicat `dateValide` qui indique si la valeur du paramètre de type `DATE` représente une date valide.

Tableaux d'enregistrements

Dans cette partie les tableaux sont déclarés de la façon suivante

```

type
  INDICE  = a .. b;
  TABLEAU = array[INDICE] of <truc>;

```

le type `<truc>` étant défini dans les exercices qui suivent.

Exercice 12 : Codes postaux

Dans cet exercice le type `<truc>` est

```

type
  CODE_POSTAL = record
    ville      : STRING;
    code       : STRING;
  end {record};

```

et la constante est déclarée

```

const
  CODES_POSTAUX : TABLEAU = (...);

```

Q 1 . Quelle expression donne le nom de la ville à l'indice k du tableau `CODES_POSTAUX`? et son code postal ?

Q 2 . Comment faire pour vérifier si le tableau défini par la constante `CODES_POSTAUX` est trié par ordre croissant des noms de ville ?

Q 3 . On suppose que le tableau est trié par ordre croissant des noms de ville, et on se donne la ville v . Donnez un algorithme pour afficher le code postal de cette ville (il se peut que plusieurs villes aient le même nom, et il se peut aussi qu'aucune ville ne correspond à ce nom).

Q 4 . On se donne un code postal c . Écrivez un programme qui affiche les noms des villes ayant c pour code postal (il se peut que plusieurs villes aient le même code postal, et il se peut aussi qu'aucune ville ne correspond au code postal).