Algorithmique

Problèmes

Quelques problèmes de la vie courante :

- cuisiner un plat d'endives au jambon
- aller à la gare Montparnasse depuis ce bâtiment
- calculer x^n où x est un réel positif et n est un entier positif

Problèmes

Quelques problèmes de la vie courante :

- cuisiner un plat d'endives au jambon
- aller à la gare Montparnasse depuis ce bâtiment
- calculer x^n où x est un réel positif et n est un entier positif

problème = relation entre deux objets :

Problèmes

Quelques problèmes de la vie courante :

- cuisiner un plat d'endives au jambon
- aller à la gare Montparnasse depuis ce bâtiment
- calculer x^n où x est un réel positif et n est un entier positif

problème = relation entre deux objets :

- **Entrée** : ingrédients, ustensiles, ...
- **Sortie** : un plat d'endives au jambon

Problèmes

Quelques problèmes de la vie courante :

- cuisiner un plat d'endives au jambon
- aller à la gare Montparnasse depuis ce bâtiment
- calculer x^n où x est un réel positif et n est un entier positif

problème = relation entre deux objets :

- **Entrée** : un point de départ et une destination
- **Sortie** : un trajet entre ces deux lieux

Problèmes

Quelques problèmes de la vie courante :

- cuisiner un plat d'endives au jambon
- aller à la gare Montparnasse depuis ce bâtiment
- calculer x^n où x est un réel positif et n est un entier positif

problème = relation entre deux objets :

- **Entrée** : un réel positif x et un entier n
- **Sortie** : le réel x^n

Cuisiner un plat d'endives au jambon

Préparer les endives : enlever les feuilles jaunies et la partie dure et amère. Nettoyer et jeter dans de l'eau bouillante. Faire cuire environ 40 mn à ébullition.
Faire fondre la crème fraîche dans une casserole à feu doux. Laisser mijoter quelques instants et ajouter sel, poivre et une pincée d'herbes de provence. Egoutter les endives et les habiller d'une demi-tranche de jambon.
Les placer dans un plat beurré et recouvrir avec la sauce. Saupoudrer de gruyère râpé. Mettre au four quelques instants jusqu'à ce que le gruyère soit fondu.

Actions

Préparer les endives : enlever les feuilles jaunies et la partie dure et amère. Nettoyer et jeter dans de l'eau bouillante. Faire cuire environ 40 mn à ébullition.
Faire fondre la crème fraîche dans une casserole à feu doux. Laisser mijoter quelques instants et ajouter sel, poivre et une pincée d'herbes de provence. Egoutter les endives et les habiller d'une demi-tranche de jambon.
Les placer dans un plat beurré et recouvrir avec la sauce. Saupoudrer de gruyère râpé. Mettre au four quelques instants jusqu'à ce que le gruyère soit fondu.

Objets

Préparer les endives : enlever les feuilles jaunies et la partie dure et amère. Nettoyer et jeter dans de l'eau bouillante. Faire cuire environ 40 mn à ébullition.
Faire fondre la crème fraîche dans une casserole à feu doux. Laisser mijoter quelques instants et ajouter sel, poivre et une pincée d'herbes de provence. Egoutter les endives et les habiller d'une demi-tranche de jambon.
Les placer dans un plat beurré et recouvrir avec la sauce. Saupoudrer de gruyère râpé. Mettre au four quelques instants jusqu'à ce que le gruyère soit fondu.

Indications de durée

Préparer les endives : enlever les feuilles jaunies et la partie dure et amère. Nettoyer et jeter dans de l'eau bouillante. Faire cuire environ 40 mn à ébullition.
Faire fondre la crème fraîche dans une casserole à feu doux. Laisser mijoter quelques instants et ajouter sel, poivre et une pincée d'herbes de provence. Egoutter les endives et les habiller d'une demi-tranche de jambon.
Les placer dans un plat beurré et recouvrir avec la sauce. Saupoudrer de gruyère râpé. Mettre au four quelques instants jusqu'à ce que le gruyère soit fondu.

Itinéraire jusqu'à la gare Montparnasse

		Allez jusqu'à : NOBEL - Bus RATP		3 mn
1	BUS 312	**Direction :** NOISY-CHAMPS-RER-DESCARTES **de :** NOBEL **à :** NOISY-CHAMPS-RER-DESCARTES	 16h15 16h17	2 mn
		Correspondance à : Noisy-Champs - RER		7 mn
2	RER A	**Direction :** Poissy **de :** Noisy-Champs **à :** Chatelet-Les Halles	 16h29 16h51	22 mn
		Correspondance à : Les Halles - Metro		7 mn
3	M 4	**Direction :** Porte d'Orleans (General Leclerc) **de :** Les Halles **à :** Montparnasse-Bienvenue	 16h59 17h09	10 mn
		Allez jusqu'à : Gare Montparnasse (SNCF)		5 mn

Calcul d'une puissance d'un nombre

Comment calculer x^n ?

Calcul d'une puissance d'un nombre

Comment calculer x^n ?

Pseudo – code

```
Fonction puiss(x : Reel, n : Entier)

Var temp,i : Entier

Debut

  temp := 1

  Si (x<0) alors
   Renvoyer ("L'entier est négatif !")
  FinSi

  Pour i de 1 à n faire
   temp:=temp*x
  FinPour

  Renvoyer temp

Fin
```

Calcul d'une puissance d'un nombre

Comment calculer x^n ?

Pseudo – code

```
Fonction puiss(x : Reel, n : Entier)

Var temp,i : Entier

Debut

  temp := 1

  Si (x<0) alors
   Renvoyer ("L'entier est négatif !")
  FinSi

  Pour i de 1 à n faire
   temp:=temp*x
  FinPour

  Renvoyer temp

Fin
```

Java

```
int temp = 1;

if (x<0) {
   System.out.println("L'entier est négatif ! ")
   }
else{
   for (int i=1; i<n; i++) {
      temp = temp*x;
      }
   return temp;
   }
}
```

Algorithme

Un algorithme sert à résoudre un problème.

Algorithme = suite d'instructions qui conduit à un résultat ...

Algorithme

Un algorithme sert à résoudre un problème.

Algorithme = suite d'instructions qui conduit à un résultat ...

... si les instructions sont exécutées correctement

... si les instructions sont compréhensibles par l'exécutant

Algorithme

Un algorithme sert à résoudre un problème.

Algorithme = suite d'instructions qui conduit à un résultat ...

... si les instructions sont exécutées correctement

... si les instructions sont compréhensibles par l'exécutant

L'algorithme est correct si ce résultat est celui qu'on souhaite, incorrect sinon.

En informatique

- exécutant = ordinateur
- algorithme écrit sous forme de programme dans un langage de programmation (C,C++,Java,Pascal,...)

Structure d'un algorithme

- en-tête : nom de l'algo, paramètres d'entrée et leur type, ...
- déclaration de variables
- corps de l'algorithme

Structure d'un algorithme

- en-tête : nom de l'algo, paramètres d'entrée et leur type, ...
- déclaration de variables
- corps de l'algorithme

```
Fonction puiss(x : Reel, n : Entier)
```

Structure d'un algorithme

- en-tête : nom de l'algo, paramètres d'entrée et leur type, ...
- déclaration de variables
- corps de l'algorithme

```
Fonction puiss(x : Reel, n : Entier)

Var temp,i : Entier
```

Structure d'un algorithme

- en-tête : nom de l'algo, paramètres d'entrée et leur type, ...
- déclaration de variables
- corps de l'algorithme

```
Fonction puiss(x : Reel, n : Entier)

Var temp,i : Entier

Debut

  temp := 1

  Si (x<0) alors
   Renvoyer ("L'entier est négatif !")
  FinSi

  Pour i de 1 à n faire
   temp:=temp*x
  FinPour

  Renvoyer temp

Fin
```

Quelques remarques sur la présentation

- séparer les différentes parties de la structure
- aller à la ligne après chaque instruction
- indenter le pseudo-code

Quelques remarques sur la présentation

- séparer les différentes parties de la structure
- aller à la ligne après chaque instruction
- indenter le pseudo-code

```
Fonction puiss(x : Reel, n : Entier)
Var temp,i : Entier
Debut temp := 1 Si (x<0) alors
Renvoyer ("L'entier est négatif !")
FinSi Pour i de 1 à n faire
temp:=temp*x FinPour Renvoyer temp Fin
```

Quelques remarques sur la présentation

- séparer les différentes parties de la structure
- aller à la ligne après chaque instruction
- indenter le pseudo-code

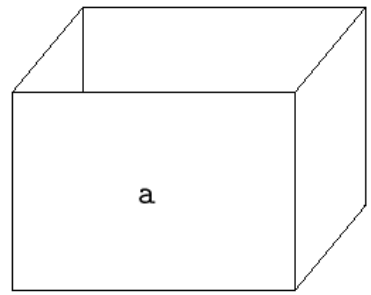
```
Fonction puiss(x : Reel, n : Entier)
Var temp,i : Entier
Debut temp := 1 Si (x<0) alors
Renvoyer ("L'entier est négatif !")
FinSi Pour i de 1 à n faire
temp:=temp*x FinPour Renvoyer temp Fin
```

$\Rightarrow$

```
Fonction puiss(x : Reel, n : Entier)

Var temp,i : Entier

Debut

  temp := 1

  Si (x<0) alors
   Renvoyer ("L'entier est négatif !")
  FinSi

  Pour i de 1 à n faire
   temp:=temp*x
  FinPour

  Renvoyer temp

Fin
```

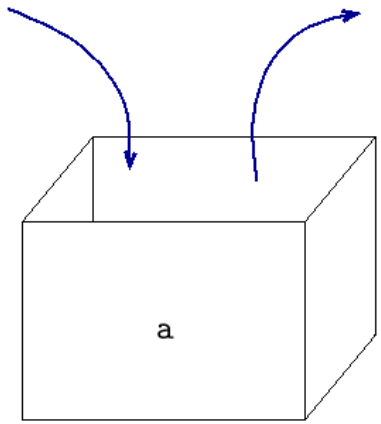
Pourquoi des variables ?

- besoin de stocker des résultats intermédiaires ou définitifs
- dans des variables



Pourquoi des variables ?

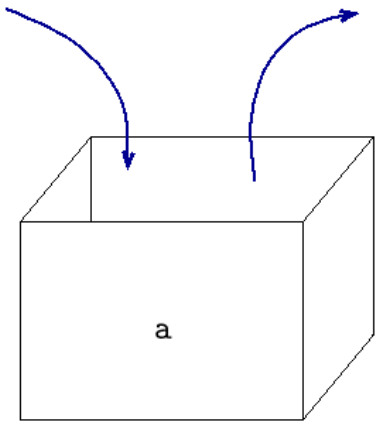
- besoin de stocker des résultats intermédiaires ou définitifs
- dans des variables



- accéder aux variables
- modifier les variables

Pourquoi des variables ?

- besoin de stocker des résultats intermédiaires ou définitifs
- dans des variables



- accéder aux variables
- modifier les variables

! une variable possède à un moment donné une valeur et une seule !

Entiers, réels

- int
- $0, -1, 14, \ldots$
- opérations usuelles : $+$, $-$, $*$
- opération modulo : $7 \text{ div } 2 = 1$

Entiers, réels

- int
- $0, -1, 14, \ldots$
- opérations usuelles : $+$, $-$, $*$
- opération modulo : 7 div $2 = 1$

- float
- $0.03, \sqrt{2}, -14, 7, \ldots$
- opérations usuelles : $+$, $-$, $*$ et $/$

Caractères, chaînes de caractères

- char
- 'a', 'z', '8', '&', ...
- comparaison : 'a' < 'z'

Caractères, chaînes de caractères

- char
- 'a', 'z', '8', '&', ...
- comparaison : 'a' < 'z'

- string
- *abba*, *toto*23, ...
- longueur : length(*abba*)= 4
- concaténation . : *abba*.*toto*23 = *abbatoto*23
- comparaison, sous-chaîne, ...

Booléens

- boolean
- *true* et *false* (ou encore 1 et 0)
- . . . à suivre . . .

Types de bases

Ces types sont appelés types de bases.

Types de bases

Ces types sont appelés types de bases.
On verra plus tard d'autres types plus complexes (tableaux, listes chaînées, . . .)

Affectation de variables

- symbole d'affectation $:=, \leftarrow$
- une affectation ne modifie que ce qui est à gauche du symbole $:=$

```
a:=2
b:=3
a:=b+1
```

Affectation de variables

- symbole d'affectation $:=, \leftarrow$
- une affectation ne modifie que ce qui est à gauche du symbole $:=$

```
a:=2
b:=3
a:=b+1
```

$\Rightarrow$ Exercices !

Instructions

On trouve quatre principaux type d'instructions :

- affectation de variables
- lecture/écriture
- tests
- boucles

Lecture/écriture

Interaction avec l'utilisateur de l'algorithme :

Lecture/écriture

Interaction avec l'utilisateur de l'algorithme :

- lecture de paramètres

Lecture/écriture

Interaction avec l'utilisateur de l'algorithme :

- lecture de paramètres (nom des stations de départ et d'arrivée)

Lecture/écriture

Interaction avec l'utilisateur de l'algorithme :

- lecture de paramètres (nom des stations de départ et d'arrivée)
- écriture du résultat final, indications

Lecture/écriture

Interaction avec l'utilisateur de l'algorithme :

- lecture de paramètres (nom des stations de départ et d'arrivée)
- écriture du résultat final, indications (itinéraire)

Logique booléenne

- Variables booléennes prennent 2 valeurs possibles : true et false (1 et 0)
-

Opérateurs booléens

Trois opérateurs principaux :

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$
- conjonction (ET logique) : $a \wedge b$

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$
- conjonction (ET logique) : $a \wedge b$
- disjonction (OU logique) : $a \vee b$

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$
- conjonction (ET logique) : $a \wedge b$
- disjonction (OU logique) : $a \vee b$

Tables de vérité :

a	$\overline{a}$
1	0
0	1

,

$\wedge$	0	1
0	0	0
1	0	1

et

$\vee$	0	1
0	0	1
1	1	1

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$
- conjonction (ET logique) : $a \wedge b$
- disjonction (OU logique) : $a \vee b$

Tables de vérité :

a	$\overline{a}$
1	0
0	1

,

$\wedge$	0	1
0	0	0
1	0	1

et

$\vee$	0	1
0	0	1
1	1	1

! Attention au parenthésage des expressions !

Opérateurs booléens

Trois opérateurs principaux :

- négation : $\overline{x}$
- conjonction (ET logique) : $a \wedge b$
- disjonction (OU logique) : $a \vee b$

Tables de vérité :

a	$\overline{a}$
1	0
0	1

,

$\wedge$	0	1
0	0	0
1	0	1

et

$\vee$	0	1
0	0	1
1	1	1

! Attention au parenthésage des expressions !

$\Rightarrow$ Exercices !