

Algorithmique



*éduc*Algo

Exemple

- Problème

Soit N un entier naturel donné,
calculer $S_N = 0 + \dots + N$

Exemple

- Problème

Soit N un entier naturel donné,
calculer $S_N = 0 + \dots + N$

- Premier procédé : calcul direct

$$S_N = \frac{N \times (N + 1)}{2}$$

Exemple

- Problème

Soit N un entier naturel donné,
calculer $S_N = 0 + \dots + N$

- Premier procédé : calcul direct

$$S_N = \frac{N \times (N + 1)}{2}$$

- Second procédé : récurrence

$$S_0 = 0 \quad \text{et} \quad S_{K+1} = S_K + (K + 1)$$

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

• Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :
(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :
(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :
(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
 - 4 Poser $K = K + 1$
 - 5 Poser $S = S + K$
 - 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :
(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
 - 4 Poser $K = K + 1$
 - 5 Poser $S = S + K$
 - 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4
4	$K + 1$	2	$K = 2$	5

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4
4	$K + 1$	2	$K = 2$	5
5	$S + K$	3	$S = 3$	3

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :
(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
 - 4 Poser $K = K + 1$
 - 5 Poser $S = S + K$
 - 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4
4	$K + 1$	2	$K = 2$	5
5	$S + K$	3	$S = 3$	3
3	$K < N$	faux		6

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4
4	$K + 1$	2	$K = 2$	5
5	$S + K$	3	$S = 3$	3
3	$K < N$	faux		6
6	S	3	Afficher 3	

Soit N un entier naturel donné, calculer $S_N = 0 + \dots + N$

Procédé : récurrence avec $S_0 = 0$ et $S_{K+1} = S_K + (K + 1)$

Exécution avec $N = 2$:

Programme :

(N supposé connu)

- 1 Poser $K = 0$
- 2 Poser $S = 0$
- 3 Tant que $K < N$ faire :
- 4 Poser $K = K + 1$
- 5 Poser $S = S + K$
- 6 Afficher S

Ligne	Calcul	Résultat	Action	Suite
1	0	0	$K = 0$	2
2	0	0	$S = 0$	3
3	$K < N$	vrai		4
4	$K + 1$	1	$K = 1$	5
5	$S + K$	1	$S = 1$	3
3	$K < N$	vrai		4
4	$K + 1$	2	$K = 2$	5
5	$S + K$	3	$S = 3$	3
3	$K < N$	faux		6
6	S	3	Afficher 3	

Variables : les variables N , S , K contiennent des valeurs

Actions : Poser, Afficher

Calculs : conditions : $K < N$, expressions : 0 , $K + 1$, $S + K$

Modèle

- ▶ Algorithmes particuliers

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques
 - ▶ Gödel, Kleene, Church

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques
 - ▶ Gödel, Kleene, Church
 - ▶ Turing, Von Neumann

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques
 - ▶ Gödel, Kleene, Church
 - ▶ Turing, Von Neumann
- ▶ Informatique

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques
 - ▶ Gödel, Kleene, Church
 - ▶ Turing, Von Neumann
- ▶ Informatique
 - ▶ Ordinateurs

Modèle

- ▶ Algorithmes particuliers
 - ▶ Euclide, Al Khuwarizmi
- ▶ Théories mathématiques
 - ▶ Gödel, Kleene, Church
 - ▶ Turing, Von Neumann
- ▶ Informatique
 - ▶ Ordinateurs
 - ▶ Programmation

Modèle

► Machine

Modèle

- ▶ Machine
mémoire : programme et données

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions
- ▶ Fonctionnement

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions
- ▶ Fonctionnement
 1. On introduit un programme dans la mémoire

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions
- ▶ Fonctionnement
 1. On introduit un programme dans la mémoire
 2. On initialise la mémoire avec des données d'entrée

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions
- ▶ Fonctionnement
 1. On introduit un programme dans la mémoire
 2. On initialise la mémoire avec des données d'entrée
 3. On lance l'exécution, ce qui produit une séquence d'étapes
 - A chaque étape, le processeur agit en suivant le schéma :
 - ▶ Ligne : Quelle ligne du programme exécuter ?
 - ▶ Calcul : Quel calcul faire, sur quelles données ?
 - ▶ Action : Que faire avec le résultat ?
 - ▶ Suite : Quelle ligne du programme exécuter ensuite ?

Modèle

- ▶ Machine
 - mémoire : programme et données
 - processeur : calculs, actions, ordonnancement des actions
- ▶ Fonctionnement
 1. On introduit un programme dans la mémoire
 2. On initialise la mémoire avec des données d'entrée
 3. On lance l'exécution, ce qui produit une séquence d'étapes
A chaque étape, le processeur agit en suivant le schéma :
 - ▶ Ligne : Quelle ligne du programme exécuter ?
 - ▶ Calcul : Quel calcul faire, sur quelles données ?
 - ▶ Action : Que faire avec le résultat ?
 - ▶ Suite : Quelle ligne du programme exécuter ensuite ?
 4. On récupère les résultats dans la mémoire ou sur l'écran

Langage

- ▶ Un algorithme est défini sous forme de texte

Langage

- ▶ Un algorithme est défini sous forme de texte
- ▶ Ce texte est écrit dans un certain langage

Langage

- ▶ Un algorithme est défini sous forme de texte
- ▶ Ce texte est écrit dans un certain langage
- ▶ Il existe des milliers de langages

Langage

- ▶ Un algorithme est défini sous forme de texte
- ▶ Ce texte est écrit dans un certain langage
- ▶ Il existe des milliers de langages
- ▶ Nous utiliserons *éduc*Algo

Langage

- ▶ Un algorithme est défini sous forme de texte
- ▶ Ce texte est écrit dans un certain langage
- ▶ Il existe des milliers de langages
- ▶ Nous utiliserons *éduc*Algo
- ▶ *éduc*Algo est composé d'*instructions*

Instructions élémentaires

► Affectation

« Affecter une *valeur* à une variable *X* »

Mettre la *valeur* dans la variable *X*

La *valeur* est obtenue comme le résultat du calcul d'une *expression*

dans *éducAlgo* : Poser *variable* = *expression*

Instructions élémentaires

► Affectation

« Affecter une *valeur* à une variable *X* »

Mettre la *valeur* dans la variable *X*

La *valeur* est obtenue comme le résultat du calcul d'une *expression*

dans *éducAlgo* : Poser *variable* = *expression*

► Affichage

Afficher la valeur d'une *expression*

dans *éducAlgo* : Afficher : *expression*

Instructions élémentaires

► Affectation

« Affecter une *valeur* à une variable *X* »

Mettre la *valeur* dans la variable *X*

La *valeur* est obtenue comme le résultat du calcul d'une *expression*

dans *éducAlgo* : Poser *variable* = *expression*

► Affichage

Afficher la valeur d'une *expression*

dans *éducAlgo* : Afficher : *expression*

► Commentaire

Cette instruction ne fait rien

dans *éducAlgo* : (*message*)

Instructions composées : Structures de contrôle

► Test

dans *éduc*Algo :

Si *condition* alors faire :

| *instructions*

sinon faire :

| *instructions*

Instructions composées : Structures de contrôle

► Test

dans *éduc*Algo :

```
Si condition alors faire :  
    | instructions  
sinon faire :  
    | instructions
```

► Boucle

dans *éduc*Algo :

```
Tant que condition faire :  
    | instructions
```

Instructions composées : Structures de contrôle

► Test

dans *éducAlgo* :

Si *condition* alors faire :
| *instructions*
sinon faire :
| *instructions*

► Boucle

dans *éducAlgo* :

Tant que *condition* faire :
| *instructions*

► Stop

dans *éducAlgo* :

Stop

Intérêt pédagogique

- Formaliser, modéliser, abstraire

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer
 - ▶ dans un langage précis et rigoureux

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer
 - ▶ dans un langage précis et rigoureux
 - ▶ en s'adressant à quelqu'un d'autre

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer
 - ▶ dans un langage précis et rigoureux
 - ▶ en s'adressant à quelqu'un d'autre
- ▶ Retarder, déléguer

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer
 - ▶ dans un langage précis et rigoureux
 - ▶ en s'adressant à quelqu'un d'autre
- ▶ Retarder, déléguer
 - ▶ le temps : on reporte l'exécution à plus tard

Intérêt pédagogique

- ▶ Formaliser, modéliser, abstraire
- ▶ Planifier, organiser, construire
- ▶ Expliciter, communiquer
 - ▶ dans un langage précis et rigoureux
 - ▶ en s'adressant à quelqu'un d'autre
- ▶ Retarder, déléguer
 - ▶ le temps : on reporte l'exécution à plus tard
 - ▶ l'action : on délègue l'exécution à quelqu'un d'autre



éducAlgo

Somme des N premiers entiers

?

Soit un nombre entier $N \geq 1$
Calculer et afficher S , la somme des nombres entiers de 1 à N :
 $S = 1 + 2 + \dots + N$

Entrée

Soit un nombre N entier ≥ 1

Traitement

Poser $K = 1$

Poser $S = 0$

Tant que $K < N$ faire :

 Poser $S = S + K$

 Poser $K = K + 1$

Afficher Somme = S

Écrire

Instructions

Affectation

Affichage

Commentaire

Contrôles

Boucle

Test

Stop

Variables

K N S

Expressions

K N S 0 1

$K + 1$ $K - 1$ $S + K$

Conditions

$K > 0$ $K < N$ $K \leq N$

Solution du système



250 ns

