

Annexe 02 : Traduction des instructions algorithmiques en C

Introduction

Le but de cette annexe est de traduire les instructions algorithmiques vues dans le chapitres 1 jusqu'au chapitre 7 en C.

Traduction des instructions du chapitre 01

Instruction algorithmique				Instruction C			
Variable i : entier				int i ;			
Constante PI = 3.14				# define PI 3.14			
Réel		Caractère		Double ou float		Char	
Variable nom : Chaîne*20				char nom[20] ;			
Ecrire("message")				// en c printf("message") ; /* N'oublier pas d'ajouter #include<stdio.h> */ // en c++ cout << "message" ; /* N'oublier pas d'ajouter #include<iostream.h> */			
Ecrire("La valeur de x est ", x) Lire(x) // on suppose que x est entier				printf("La valeur de x est %d ", x) ; scanf("%d" , & x) ;			
double	float	char		%f	%f	%c	%s
i ← 4		i ← i + 1		i = 4 ;		i + = 1 ;	
Div	Mod	Et		(int)(x/y)		%	&&
Non	Ou	x exposant y		!		Pow(x,y)	

Ecrire("\n")		printf("\n")	
Ecrire("Le résultat est ", r) // on suppose que x est réel		printf("Le résultat est %f ", r) ;	
Ecrire(a, b, b)		printf(a, b, c)	
Lire(a, b, c) /* on suppose que a est entier, b est réel et c est de type caractère*/		scanf(“%d %f %c “, & a, & b & c) ;	
Vrai	Faux	Toute valeur différente de 0	0
>	<	<=	>=
= (égal à)	<>	== (égal à)	!= (différent de)

Note :

En C, il a une différence entre *majuscule* et *minuscule*. Autrement dit, le langage C/C++ est sensible à la casse.

```
char nom[20] ;
// la variable nom est différent de la variable Nom de l'instruction
suivante :
char Nom[20] ;
// Qui sont différents de la variable NOM de l'instruction suivante :
char NOM[20] ;
```

Traduction des instructions du chapitre 02

Instruction algorithmique	Instruction C
---------------------------	---------------

<p>Si ... alors instruction(s) FinSI</p>	<p>/* La condition doit être entre des parenthèses. Pas de Then après la condition */ if (condition) { instruction(s) ; } Note: Pour une seule instruction ; les accolades ne sont pas obligatoires : Exemple : if (7 > 2) printf("bonjour") ;</p>
<p>Si ... alors instruction(s) Sinon instruction(s) FinSI</p>	<p>if (condition) { instruction(s) ; } else { instruction(s) ; }</p>

<p>Selon ... faire ... Sinon ... FinSelon</p>	<pre>switch (variable) { case valeur1 : instruction(s) ; break ; case valeur2 : instruction(s) ; break ; case valeurs... : instruction(s) ; break ; default: instruction(s) : break ; }</pre>
<p>Pour i ← 0 jusqu'à 10 instruction(s) FinPour</p>	<pre>For(i = 0; i < 10; i ++) { instruction(s) ; }</pre>
<p>Pour i ← 1 jusqu'à 10 pas 2 instruction(s) FinPour</p>	<pre>For(i = 1; i <= 10; i = i +2) { instruction(s) ; }</pre>
<p>Quitter Pour</p>	<p>break ;</p>
<p>TantQue ... Faire instruction(s) FinTantQue</p>	<pre>while (condition) { instruction(s) ; }</pre>

Traduction des instructions du chapitre 03

Instruction algorithmique	Instruction C/ C++
---------------------------	--------------------

Variable T : tableau[1 .. 5] de réel	Float T[5] ; Note : L'indice du premier élément est 0 et nom pas 1.
Variable T : tableau[1..5, 1..10] de réel	Float T[5][10] ;

Traduction des instructions du chapitre 04

Instruction algorithmique	Instruction C/ C++
Type structure personne Nom : chaîne*30 Prenom : chaîne*30 EndStruct	Struct personne { Char Nom[30] ; Char Prenom[30] ; };
Variable e : personne	Struct personne e ;
e.Nom ← "Sabir"	e.Nom = "Sabir"

Traduction des instructions du chapitre 05

Instruction algorithmique	Instruction C/ C++
Fonctions liées aux chaînes de caractères	
Longueur(ch1)	strlen(ch1)
Concat(ch1, ch2)	strcat(ch1, ch2)
Comp(ch1,ch2)	strcmp(ch1,ch2)
Fonctions mathématiques et trigonométriques	
Sqrt(x)	sqrt(x)
Abs(x)	fabs(x)
Cos(angle)	cos(angle)
Sin(angle)	sin(angle)
Tan(angle)	tan(angle)

Traduction des instructions du chapitre 06

Déclaration d'une procédure

<p>Instruction algorithmique</p>	<pre>// Déclaration d'une procédure Procédure nom_proc(p1: type, p2 : type, ...) Variables x1 : type ... xn : type Début Instruction(s) FinProc</pre>
<p>Instruction C</p>	<p>La notion de procédure n'existe pas en C. Elle est remplacée par la notion de fonction.</p>

<p style="text-align: center;">Déclaration d'une fonction</p>	
<p>Instruction algorithmique</p>	<pre>// Déclaration d'une fonction Fonction nom_Fonct(p1: type, p2 : type, ...) : type Variables x1 : type ... xn : type Début Instruction(s) Retourner expression FinFonct</pre>
<p>Instruction C</p>	<pre>// Déclaration d'une fonction Type nom_Fonct(type p1, type p2, ...) { type x1; ... type xn; Instruction(s) Return expression }</pre>

<p>En C ou C++, le programme principal est une fonction.</p>	<pre>// en c Void main() { instruction(s) ; } /* en c++ */ Int main() { ... ; Return 0 ; }</pre>
---	--

Appel d'une fonction	
Instruction algorithmique	Nom_Fonct(liste_de_paramètres)
Instruction C	Nom_fonct liste_de_paramètres Ou bien y = Nom fonct (liste de paramètres)

Traduction des instructions du chapitre 07

On vous conseille de voir un ouvrage spécialisé pour le traitement du fichier en C.