Introduction au langage C

D.E ZEGOUR

École Supérieure d'Informatique

ESI

Introduction au langage C

Structure d'un programme

Fichiers "inlude"
Déclaration des variables globales.

Fonction1

Fonction2

*

*

Fonctionn

Programme principal qui commence par main()

Introduction au langage C

Exemple d'un programme C

```
#include <stdio.h>
#include <stdlib.h>

typedef int bool ;

#define True 1
#define False 0
```

```
/** Implémentation **\: TABLEAU D' ENTIERS**/
/** Tableaux **/

typedef int Typeelem ;
typedef Typeelem * Typevect ;

Typeelem Element ( Typevect V , int I1  )
 {
  return  *(V + I1-1 ) ;
 }

void Aff_element ( Typevect V  , int I1 ,  Typeelem Val )
 {
  *(V + I1-1 ) = Val ;
 }
```

Introduction au langage C

Exemple d'un programme C

```
/** Variables du programme principal **/
Typevect V;
int Val;

/** Prototypes des fonctions **/
void Init();
bool  Recherche_binaire (int *Donnee) ;
```

```
void Init()
   {
     /** Corps du module **/
    Aff_element(V , 1   , 15 ) ;
    Aff_element(V , 2   , 156 ) ;
    Aff_element(V , 3   , 1567 ) ;
    Aff_element(V , 4   , 15678 ) ;
    Aff_element(V , 5   , 156789 ) ;
}
```

Introduction au langage C

Exemple d'un programme C

```
bool  Recherche_binaire (int *Donnee)
   {
     /** Variables locales **/
     int I;
     bool Trouv;
     bool Arret;
     /** Corps du module **/
    I  =  1 ;
    Trouv  =  False ;
    Arret  =  False ;
    while( ( I <= 5 ) && ! Trouv && ! Arret)
     if( Element(V , I   ) == *Donnee)
       Trouv  =  True;
     else
        if( Element(V , I   ) > *Donnee)
           Arret  =  True;
       else
           I  =  I + 1;
    return Trouv ;
   }
```

Introduction au langage C

Exemple d'un programme C

```
int main(int argc, char *argv[])
   {
    V = malloc(5 * sizeof(int));
    Init() ;
    scanf ( " %d", &Val ) ;
    printf ( " %d", Recherche_binaire(&Val) );

     system("PAUSE");
   }
```

Introduction au langage C

Types de données simples

Le langage C manipule plusieurs classes d'objets. Les plus utilisées sont les suivantes avec leurs équivalents en PASCAL :

C	PASCAL
char	char
int	integer
float	real

En C, le type booleen n'existe pas.
La valeur 0 désigne Faux
Toute valeur différente de 0 désigne Vrai.

Introduction au langage C

Définition des types

Un type ou classe d'objets est défini par :

```
struct Nomdestructure
{
  Champ1;
  Champ2;
   ...
  Champn;
}
```

Champ1, ... Champn sont des définitions de variables de type quelconque.

Les variables de cette classe sont définies par :

Nomdestructure N1, N2, ... ;

L'accès se fait par le chemin et l'utilisation des points.

N1.Champ1

Introduction au langage C

Tableaux

Un tableau est déclaré de la façon suivante :

Type nomdetableau(taille)

L'indice du premier élément est 0.
Le dernier élément a comme indice taille - 1.

Exemple : int A(20); char P(10)

Introduction au langage C

Chaînes de caractères

Une chaîne de caractères est un tableau de caractères. Toute chaîne doit se terminer par le caractère nul '\0'.

Quelques opérations sur les chaînes de caractères :

strcat (s1, s2) : s1 reçoit la concaténation de s1 et s2.
strcmp(s1, s2) : rend la valeur 0 si s1 = s2, une valeur positive si s1 > s2 et une valeur négative si s1 < s2.
strcpy(s1, s2) : copier s2 dans s1.
strlen(s) : nombre de caractères dans s, y compris '\0'

Introduction au langage C

Expressions

Donnons ci-après la correspondance entre les opérateurs PASCAL et C :

	C	PASCAL
Arithmétique	+, -, *, /, %	+, -, *, /, Mod
Logique	!, &&, ||	Not, And, OR
Relation	<, <=, >, >=, !=, ==	<, <=, >, >=, <>, =

Introduction au langage C

Instructions

Les principales instructions C sont les suivantes :

Bloc	:	{ déclarations; inst1; inst2; .... instn }
Affectation	:	V = Expression
Instruction composée	:	{ inst1; inst2; .....instn }
Conditionnelle	:	if (expression) inst
Alternative	:	if (expression)inst else inst
Boucle While	:	while (expression) inst
Boucle For	:	for (initialisation; condition; expression) inst

Introduction au langage C

Quelques raccourcis

i++	i = i + 1
X+= Y	X = X + Y
A*=B+C	A = A * (B+C)

Introduction au langage C

Opérations simples d'entrées/sorties

getchar() : lit un caractère de l'écran. Exemple : ch = getchar()
putchar() :écrit un caractère sur l'écran. Exemple : putchar(ch)

gets() : lit une chaîne de caractères de l'écran. Exemple : gets(s)
puts() : écrit une chaîne de caractères sur l'écran. Exemple : puts(s)

printf("chaîne de contrôle1", liste des arguments) : lit des données à partir de l'écran dirigées par des formats.

scanf("chaîne de contrôle2", liste des arguments) : écrit des données sur écran dirigées par des formats.
"chaîne de contrôle1" est composée de deux choses :

a) les caractères à imprimer
b) les formats des arguments qui commencent par %.

Introduction au langage C

Format d'entrée sortie

%c caractère
%d, %i décimal
%f décimal flottant
%s chaîne de caractères
%o octal
%x hexadécimal
\n saut de ligne
\f saut de page
\\ arrière

Les formats peuvent être précédés d'une longueur. C'est le nombre de caractères sur lesquels sera écrite la donnée.

Exemple : printf("Ceci est un %s %d %c", exemple, 10,'.') donne
Ceci est un exemple 10.

Exemple : scanf("%d %S3, &cpr, &add)
lecture d'un décimal dans cpr puis lecture d'une chaîne de caractères dans add.

Introduction au langage C

Autres fonctions d'entrées/ sorties

fscanf() : similaire à la fonction scan() sauf que les données sont lues à partir d'un fichier.

fprintf() : similaire à la fonction printf sauf que les sorties se font sur un fichier.

fopen() : ouvre un fichier pour utilisation et rend un pointeur qui identifie ce fichier.

fclose() : ferme un fichier préalablement ouvert.

feof() : prédicat égal à vrai si la fin du fichier est rencontrée.

rewind() : positionnement au début du fichier.

Introduction au langage C

Autres fonctions d'entrées/ sorties

L'ouverture se fait comme suit : fp = fopen("nom du fichier physique", mode)
fp est déclarée par File *fp.

Mode prend les valeurs suivantes :

rm : ouvrir le fichier pour lecture
wm : créer un fichier pour écriture
r+m : ouvrir un fichier pour lecture/écriture
w+m : créer un fichier pour lecture/écriture

Pour utiliser toutes les fonctions que nous venons de décrire, il faut inclure le fichier STDIO.H dans la partie déclaration du programme: #include <stdio.h>

Introduction au langage C

Procédures et fonctions

En C, on ne parle que de fonctions. On peut toutefois simuler des procédures en utilisant le mot-clé void.

Fonctions
type Nomedefonction (listes de déclaration de paramètres)
{
  Déclaration de variables locales
  Instructions
}

L'instruction return doit exister et remet la valeur de la fonction.

Introduction au langage C

Exemple : Procédure et fonctions

```
void Interchanger( int *X, int *Y);
{
  int Temp;
  Temp = *X;
  *Y = *X ;
  *Y = Temp
   }
main()
{
  X = 10;
  Y = 20;
  Interchanger(&X, &Y);
  printf("X = %d, Y = %d", X, Y);
}
```

Introduction au langage C

Les fichiers en mode bufférisé

fread(&buffer, sizeof(buffer), 1, fp) : lecture de l'enregistrement courant dans la zone Buffer.
sizeof() est une fonction qui rend la taille d'un objet(en octets).

fwrite(&buffer, sizeof(buffer), 1, fp) : écriture de la zone Buffer sur le fichier à la position courante.

Pour lire et ecrire plusieurs enregistrements d'un seul coup, remplacer le 1 par n.

fseek(fp, déplacement, origine) : se positionner directement sur une position du fichier.

Déplacement est le nombre d'enregistrements à partir de l'origine.
Origine peut être soit 0 (début du fichier), soit 1 (position courante), soit 2(fin du fichier)

Introduction au langage C

Les fichiers en mode bufférisé

Buffer est une structure d'un certain type

Le fichier logique fp est déclaré par FILE *fp. Les opérations fopen et feof sont utilisées dans ce mode.

Pour utiliser toutes les fonctions que nous venons de décrire,
il faut inclure le fichier STDIO.H dans la partie déclaration du programme :#include <stdio.h>

Introduction au langage C

Exemple

```
#include <stdio.h>
struct  Typebloc
     {
       int Nb;
       int Tab[10];
     };
FILE *Fp ;
struct Typebloc  Buffer;
int I;
```

```
main()
   {
     Fp = fopen("F.c", "w+b");
     for ( I=0; I<10; I++)
        {
          Buffer.Nb = 10*I;
          Buffer.Tab[1] =50*I;

          fwrite(&Buffer, sizeof( Buffer), 1, Fp);
       }
```

Introduction au langage C

Exemple

```
rewind(Fp);
for ( I=0; I<10; I++)
    {

       fread(&Buffer, sizeof(Buffer), 1, Fp);
       printf(" Nbb = %d,    Tab(1) = %d\n" , Buffer.Nb,
       Buffer.Tab[1]);
    }
```

Nb = 0, Tab(1) = 0
Nb = 10, Tab(1) = 50
Nb = 20, Tab(1) = 100
Nb = 30, Tab(1) = 150
Nb = 40, Tab(1) = 200
Nb = 50, Tab(1) = 250
Nb = 60, Tab(1) = 300
Nb = 70, Tab(1) = 350
Nb = 80, Tab(1) = 400
Nb = 90, Tab(1) = 450

Introduction au langage C

Exemple

```
Buffer.Nb = 666;
Buffer.Tab[1] = 999;
I=6;
fseek (Fp, (long) ( (I-1)* sizeof( struct Typebloc ) )  , 0 );
fwrite(&Buffer, sizeof( Buffer), 1, Fp);
```

Fichier modifié:
Bloc 6 : 666,[999,...]

Introduction au langage C

Exemple

```
rewind(Fp);
printf( " taille du Buffer = %d \n", sizeof(  Buffer) );
for ( I=0; I<10; I++)
    {
       fread(&Buffer, sizeof(Buffer), 1, Fp);
       printf(" Nb = %d,    Tab(1) = %d\n " , Buffer.Nb,
       Buffer.Tab[1]);
    }
```

taille du Buffer = 44
Nb = 0, Tab(1) = 0
Nb = 10, Tab(1) = 50
Nb = 20, Tab(1) = 100
Nb = 30, Tab(1) = 150
Nb = 40, Tab(1) = 200
Nb = 666, Tab(1) = 999
Nb = 60, Tab(1) = 300
Nb = 70, Tab(1) = 350
Nb = 80, Tab(1) = 400
Nb = 90, Tab(1) = 450

Introduction au langage C

Exemple

```
I=7;
printf(" Lecture du %d -ieme bloc \n",I);
fseek (Fp, (long) ( (I-1)  * sizeof( struct Typebloc))  , 0 );
fread(&Buffer, sizeof(Buffer), 1, Fp);
printf(" Nb = %d,    Tab(1) = %d\n " , Buffer.Nb,
Buffer.Tab[1]);
}
```

Lecture du 7 -ieme bloc
Nb = 60,   Tab(1) = 300