Structures de données et algorithmes – TP1

Maria-Iuliana Dascalu
mariaiuliana.dascalu@gmail.com

Objectifs pour aujourd’hui

- exécuter et compiler des programmes en C
- identifier la structure d'un programme en C
- utiliser les opérations standard pour I/O
- définir des variables
- déclarer et mettre en œuvre des fonctions
- rendre des structures

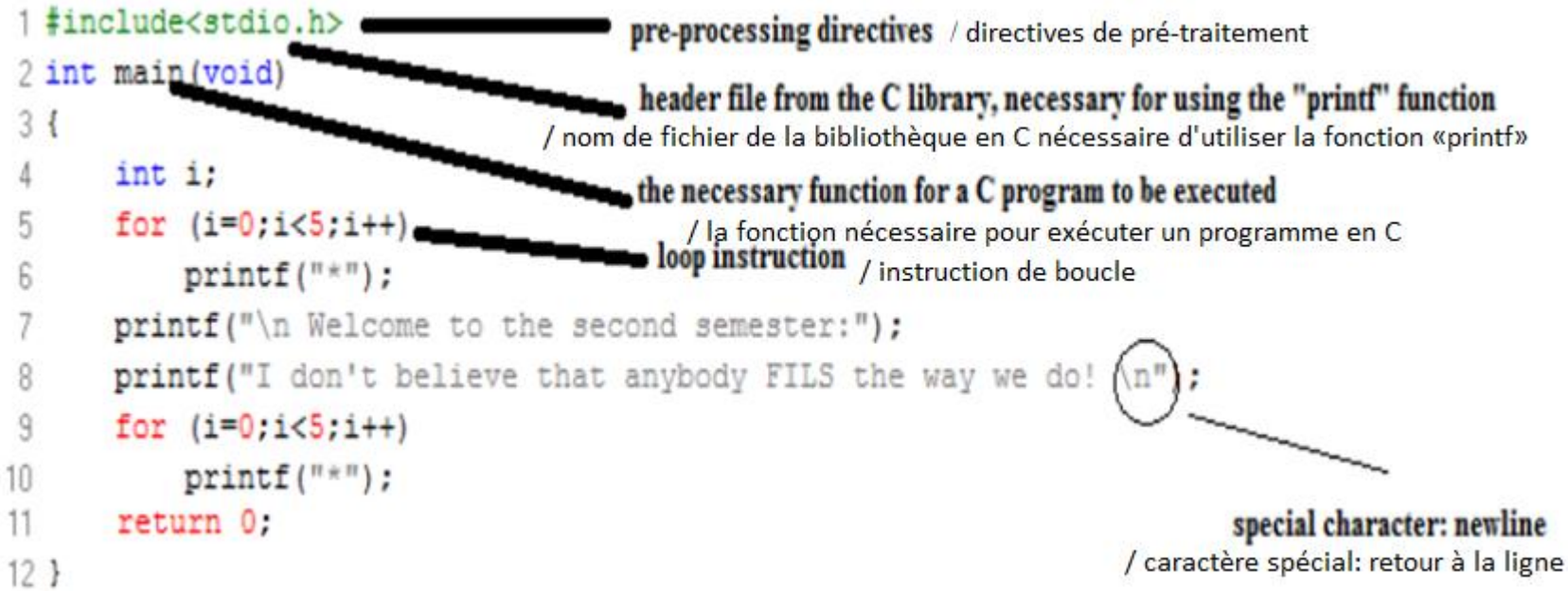
Logiciels

- C-Free 4.0 Standard (http://www.programarts.com/cfree_en/download.htm)

- Toute autre IDE (Integrated development environment= Environnement de développement intégré) ou compilateur pour C / C++ (e.g. GCC en Linux)

Exercice: Identifier la structure d'un programme typique en C!

- Attention, C est sensible à la casse!



•Un programme en C est écrit dans un fichier avec l'extension «.c»: le code source.
•Après la compilation, un autre fichier avec l'extension «.o» apparaît: le code objet.
•Après l'exécution, un autre fichier, avec l'extension «.exe» apparaît: l'exécutable.

Opérations standard de sortie

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int lungime_dreptunghi;
5     int latime_dreptunghi = 10;
6     int perimetru, arie;
7     lungime_dreptunghi = 20;
8     perimetru = 2 * ( lungime_dreptunghi + latime_dreptunghi );
9     arie=lungime_dreptunghi * latime_dreptunghi;
10    printf("Latimea dreptunghiului este = %d\n",latime_dreptunghi);
11    printf("Lungimea dreptunghiului este = %d\n",lungime_dreptunghi);
12    printf("Perimetrul dreptunghiului = %d\n", perimetru );
13    printf("Aria dreptunghiului = %d\n", arie);
14    return 0;
15 }
```

spécificateur de format pour les variables de type int

format specifier for int variables

- Autres spécificateurs de format:

%i or %d	int
%c	char
%f	float
%lf	double
%s	string

Signature de la fonction *printf*

- printf(control, par1, par2, …, parn);

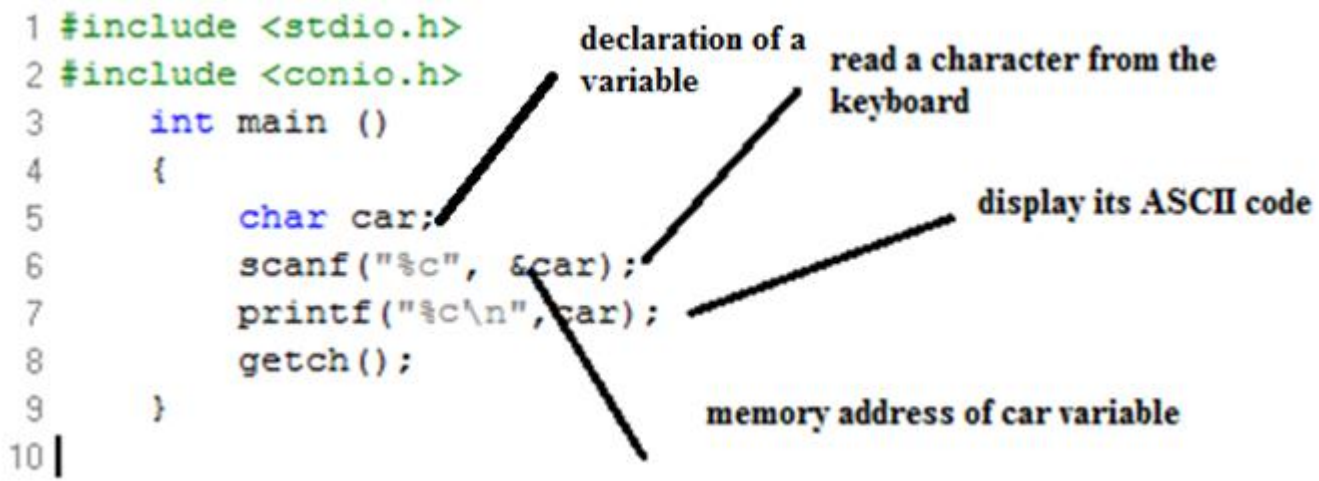
Où

- control = une chaîne qui définit les textes et les spécificateurs de format
- par1, par2, …, parn = expressions; leurs valeurs sont écrites en tenant compte du format de spécificateurs de contrôle

Exercice: quelles sont les règles pour la signature de *printf?*

- printf("*%4c**", getchar());
- printf("*%-4c**", getchar());

Opérations standard d'entrée



«Scanf» a la même signature que «printf» et elle est définie dans «stdio.h» .

Fonctions: déclaration et mise en œuvre

- Signature:

type_du_résultat_retourné nom_fonction
(liste_des_paramètres_formels)
{
déclaration_de_variables_locales;
instructions;
}

- Domaine de visibilité: les variables locales vs globales
- Passage de paramètre: par valeur

Exemple

- Notez l'utilisation de la bibliothèque math.h pour la fonction sqrt (la même signification que dans Java)
- Notez les structures de contrôle de flux (if, if-else, for, …)
- Notez la définition de la fonction et l'appel: la fonction de mise en œuvre calcule si un nombre est premier ou non

```c
#include <stdio.h>
#include <math.h>
int m,n,aux,y,i,j;
int prim(int x)
{
    int d;
    if (x%2==0) if(x==2) return 1;
                    else return 0;
        else for (d=3;d<=sqrt(x);d+=2)
                if (x%d==0) return 0;
    return 1;
}
void main(void)
{
    printf("Dati m si n :\n");
    scanf("%d%d",&m,&n);
    if (m>n) {aux=n;n=m;m=aux;}
    if (n%2 ==1) n--;
    if (m%2!=0) m++;
    if (m<=2) m=4;
    for (y=m;y<=n;y+=2)
    {       if (y==4) { printf("4=2+2\n");m+=2;}
            else  for (i=3;i<=y/2;i+=2)
                    if (prim(i) && prim(y-i))
                    {
                        printf("%d=%d+%d\n",y,i,y-i);
                        }
    }
}
```

Goldbach conjecture: Every even integer greater than 2 can be expressed as the sum of two primes.[

Structures

- un type de données défini par l'utilisateur qui permet de grouper des éléments hétérogènes
- une collection d'une ou plusieurs variables (champs), regroupées sous un même nom
- les membres d'une structure sont accessibles par «.»

Exemples

```
struct data {
unsigned char day;
unsigned char month;
unsigned long year;
char name_day[3];
char name_month[4];
};

typedef struct data data;
data today;
```

```
typedef struct data {
unsigned char day;
unsigned char month;
unsigned long year;
char name_day[4];
char name_year[4];
} data;
data today;
```

```
void writeDDMMMYYYY(data myDate)
{
    printf("%2d %s %4d ", myDate.day,
    myDate.name_month, myDate.year);
}
```

Exercices

- Ex1. Ecrivez un programme qui calcule la moyenne entre deux nombres flottants. Le résultat doit être affiché avec 2 décimales. Utilisez scanf et printf!
  - %.2f -> spécificateur de format pour float avec 2 décimales
- Ex2. Affichez le minimum des trois nombres flottants. Utilisez des fonctions.
- Ex3. Ecrivez un programme qui affiche les nombres impairs jusqu'à 25.
- Ex4. Ecrivez un programme qui lit un nombre de la console et écrit sur la console si le nombre est pair ou impair.
- Ex5. Concevoir une structure pour représenter les dates et écrire des fonctions que:
  - Vérifiez si une valeur variable de la structure est une date valide.
  - Calculez la prochaine date à une date donnée.
  - Calculez la date avant une date donnée.

Devoir

1. Terminez ex. 5.

2. Polynômes à coefficients entiers rares sont des polynômes des degrés grands et des nombreux coefficients égaux à 0. Elles peuvent être représentées par une structure de données définie par:

```
typedef struct
{
  int Coef;
  unsigned int Exponent;
}TMonom
typedef TMonom Tpolinom[50];
```

Écrivez des fonctions pour l'écriture et la lecture des polynômes rares.

BIBLIOGRAPHIE

- Dascalu, M.I. – Lecture notes, https://app.campadillo.com
- “C++ Programming Language”, Bjarne Stroustroup
- “Thinking in C++”, by Bruce Eckel & Chuck Allison
- “C++ Plus Data Structures”, by Nell Dale
- “Limbajele C si C++ pentru incepatori”(vol 1-C, vol 2-C++), by Liviu Negrescu (en roumain)
- http://casteyde.christian.free.fr/cpp/cours/online/book1.htm (en français)