

TP1. Introduction en programmation en C: programmes élémentaires

Dimitri Galayko

1 Saisir et compiler les programmes élémentaires

Saisissez et compilez les programmes étudiés en cours d'introduction, et donnés en annexe du document:

- le programme "hello_world"
- le programme "saisir_valeur"
- le programme "saisir_valeur avec cycles".

2 Modifier les programmes élémentaires

On va modifier le programme "saisir valeur".

Exo 2.1. Modifiez le traitement de la valeur saisie au clavier (dans le code original, il s'agit d'un calcul de la puissance 4 de la valeur). Le programme modifié devra retourner :

- la parité de la valeur saisie: si la valeur saisie est paire, afficher "La valeur <valeur> est paire", sinon "La valeur <valeur> est impaire". La séquence "<valeur>" doit être remplacée par la valeur saisie.

Exo 2.2. Demander à l'utilisateur de saisir deux valeurs, et afficher leurs produit.

3 Les dessins élémentaires

Ecrivez un programme qui dessine un losange avec les caractères "*" :

```
  *
 ***
*****
*****
*****
*****
*****
***
  *
```

Modifiez le programme pour qu'un rectangle soit dessiné.

4 Cycles

L'utilisation du cycle do...while a été présenté en cours. Ecrivez les programmes suivantes en utilisant cet outil:

Exo 4.1. Une factorielle. Modifiez le programme "saisir_valeur" pour que la valeur affichée soit égale à la factorielle de la valeur saisie.

Exo 4.2. Somme des nombres de 0 jusqu'à n. Modifiez le dernier programme pour que le résultat soit la somme de tous les entiers, depuis 0 jusqu'à la valeur saisie par l'utilisateur.

5 Programmes interactives

On conçoit le programme qui réalise le jeu suivant:

- Le programme propose à l'utilisateur de penser un nombre dans l'intervalle donnée (par ex., 1 et 50).
- Le programme tente de deviner le nombre saisi par l'utilisateur. Il propose un nombre. L'utilisateur doit dire, si le nombre qu'il a pensé est plus grand (il met alors 1), plus petit (il met alors -1) que le nombre proposé par l'ordinateur. Si le nombre proposé par l'ordinateur est celui que l'utilisateur a pensé, l'utilisateur saisi 0.
- En fonction de la réponse de l'utilisateur, l'ordinateur propose un nouveau nombre.
- Les étapes se répètent jusqu'à ce que l'ordinateur ait deviné le nombre. Un message "le nombre a été deviné" est alors affiché.

Exo 5.1. Proposez un algorithme que doit suivre l'ordinateur pour deviner le nombre en temps minimal,

Exo 5.2. Avec l'algorithme proposé, en combien de pas au maximum le nombre sera deviné ?

Exo 5.3. Ecrivez le programme qui réalise ce jeu.

6 Annexes: les programmes donnés en cours d'introduction

Listing 1: Le programme Hello world

```
// Ce fichier réalise un programme qui affiche "Hello , world"
// Cette ligne commençant par des // est un commentaire. On peut y mettre n'importe
//    quoi. On l'utilise pour expliquer le programme qui est écrit

// La directive de preprocesseur #include permet de connecter une bibliothèque C
//    nécessaire pour l'exécution du programme.
// la bibliothèque "stdlib" est une bibliothèque standard
#include <stdlib.h>

// La fonction principale: chaque programme le contient
main() {

// Cette commande affiche une ligne dans le terminal
    printf("Hello , _world!\n");

// cette commande termine le programme. Elle est optionnelle. Le zéro entre les
//    parenthèses est la valeur retournée au système d'exploitation, sans intérêt pour
//    nous pour l'instant.
    return(0);

}
```

Listing 2: Le programme Saisir une valeur

```
// Ce programme demande à l'utilisateur de saisir une valeur entière inférieure à 10,
// et calcule/affiche la puissance 4 de cette valeur. Si la valeur saisie est
// supérieure ou égale à 10, un message d'erreur est généré.

#include <stdio.h>

main() {
    // On deux variables de type entier
    // La déclaration des variables avant leur utilisation est obligatoire en langage C
    // le mot clé "int" signifie le type des variables : integer (entier). a et c sont
    // les noms des variables. Ils peuvent être arbitraires.
    int a, c;

    // Printf affiche une phrase invitant à saisir une valeur
    // Remarquez qu'il n'y a pas de symbole \n pour passer à la ligne:
    // l'utilisateur saisira la valeur au même niveau.
    printf("Entrez une valeur entière inférieure à 10: ");

    // Cette fonction permet de lire une valeur saisie au clavier.
    // entre guillemets, on insère l'expression du format qui donne le type de la valeur
    // lue (entière, à virgule flottante, hexadécimale, etc).
    // %i signifie "entier décimal"
    // après la virgule, on donne la variable vers laquelle on enregistre la valeur lue.
    // Cette variable doit avoir été déclarée comme entière.
    // Le signe "et commercial" devant la variable signifie "indirection": on verra ce
    // que cela signifie dans la suite des cours.
    scanf("%i", &a);

    // On utilise de nouveau la fonction printf pour afficher la valeur que l'utilisateur
    // a saisi. Ici, entre les guillemets, on met la phrase, et dedans on voit la chaîne
    // de caractère "%i". A la place de cette chaîne, la fonction printf affichera la
    // valeur qu'elle lira dans la variable donnée plus loin (ici, a). La variable doit
    // être du type entière (%i signifie "entier décimal", comme pour scanf).
    printf("La valeur saisie est %i\n", a);

    // La bifurcation (l'aiguillage): en fonction de la valeur de la variable a qui
    // contient maintenant la valeur saisie par l'utilisateur, on fait deux actions
    // différentes. On compare a et 10 avec l'expression a<10.
    //
    // Si la vérification est vraie (a inférieure à 10), on exécute le code qui se trouve
    // entre les accolades immédiatement après if.
    if (a<10){
        // on calcule la puissance 4 de a, et enregistre le résultat dans c
        c=a*a*a*a;
        // on affiche la valeur de c, de la même manière que l'on a fait pour a plus haut
        printf("La valeur de %i puissance 4 est %i\n", a, c);
    }
    // Sinon, si la vérification est fausse, on exécute le code entre accolades qui suit
    // la commande else
    else{
        // On informe l'utilisateur de son erreur, et on ne fait rien d'autre.
        printf("Vous avez saisi une valeur plus grande que 10. Au revoir.\n");
    }
    // on termine le programme
    return(0);
}
```

Listing 3: Le programme Saisir une valeur avec une boucle

```
// Ce programme demande à l'utilisateur de saisir une valeur entière inférieure à 10,
// et calcule/affiche la puissance 4 de cette valeur. A la différence du programme "
// saisir_valeur", si l'utilisateur a saisi une valeur plus grande ou égale à 10, le
// programme redemande l'utilisateur de saisir une valeur inférieure à 10, jusqu'à ce
// que la valeur saisie ne soit correcte.

#include <stdio.h>

main() {

    int a, c;

    // On introduit une instruction de cycle "do-while". Un cycle est un mécanisme du
    // langage C qui permet de répéter un bout de code plusieurs fois, jusqu'à ce qu'une
    // condition appelée "condition de poursuite" ou "condition du cycle" ne soit
    // invalidée (fausse). Dans le cycle do-while, la condition de poursuite est vérifiée
    // après chaque execution du code. Le code répété s'appelle "corps du cycle" ou "
    // corps de la boucle", il est enfermé entre les accolades entre do et while.

    do{

        printf("Entrez une valeur entière inférieure à 10: ");

        scanf("%i", &a);

        printf("La valeur saisie est %i\n", a);

    }while(a>=10);
    // Le code est répété tant que la valeur saisie est fausse, c.a.d., tant que a est
    // supérieur ou égale à 10.

    // Si on est sorti du cycle, ce que la valeur saisie a été correcte. On calcule alors
    // la puissance 4 de cette valeur.
    c=a*a*a*a;

    // et on affiche le résultat.
    printf("La valeur de %i puissance 4 est %i\n", a, c);

    return(0);
}
```