

Licence 1 - section B

TP 3 d'éléments d'informatique

Catherine RECANATI – Département d'Informatique – Institut Galilée

Semaine du 21 au 25 novembre 2016

Ce préambule est le même à peu de choses près que celui du TP2. Vous pouvez passer directement à la section Lecture de données utilisateur entrées au clavier si vous avez déjà compris les manipulations à effectuer pour mettre les programmes de ce TP dans un répertoire particulier.

1 Lecture de données utilisateur entrées au clavier

Exercice 1.1 Lecture de données avec contrôle des entrées

1. Ecrire un programme qui demande à l'utilisateur d'entrer un nombre entier impair compris entre 1 et 30, et qui réitère cette demande tant que le nombre entré au clavier par l'utilisateur ne satisfait pas ces conditions.
2. Même question pour la lecture d'un caractère de l'alphabet français (lettre majuscule ou minuscule).

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf, scanf */

/* declarations des constantes et types utilisateurs */

/* declarations des fonctions utilisateurs */
int saisir_nbimpair_inf_30();
char saisir_caractere();

/* fonction principale */
int main()
{
    int a;
    /* char c;   pour la 2eme question */

    a = saisir_nbimpair_inf_30 ();
    /* c = saisir_caractere ();   pour la 2eme question */

    /* affichages optionnels */
    printf("Vous avez saisi le nombre %d \n", a);
    /* printf("Vous avez saisi le caractere %c \n", c); pour la 2eme question */

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
int saisir_nbimpair_inf_30()
{
    int n = 0;

    while ( ((n>= 1) && (n <=30)) && (n%2 != 0) ) {
        printf("Entrez un nombre impair entre 1 et 30: ");
        scanf("%d", &n);
    }
    return n;
}
```

```

/* pour la 2eme question */
char saisir_caractere()
{
    char c = '\0';

    while ( ( (c>='a') && (c<='z') ) && ( (c>='A') && (c<='Z') ) )
    {
        printf("Entrez une lettre de l'alphabet: ");
        scanf("%c", &c);
    }
    return c ;
}

```

Exercice 1.2 Saisie de données

1. Que fait le programme suivant ?

```

/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf, scanf */

/* declarations des constantes et types utilisateurs */

/* declarations des fonctions utilisateurs */

/* fonction principale */
int main()
{
    int a;
    double b;
    char c;

    printf("Entrez un nombre entier puis un nombre réel puis un caractère : ");

    scanf("%d",&a);
    scanf("%lg",&b);
    scanf(" %c",&c);

    printf("Vous avez saisi %d puis %g puis %c.\n",a,b,c);

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */

```

Ce programme :

- déclare 3 variables a,b et c, respectivement de type entier, réel (rationnel) et caractère ;
- demande à l'utilisateur de saisir 3 valeurs, respectivement de type entier, réel (rationnel) et caractère ;
- affecte la valeur entière saisie à la variable entière a (même type sinon quelle est la signification ? le compilateur détecte cette erreur sémantique lors de l'analyse sémantique mais il acceptera de compiler en faisant une conversion automatique de type => source de bugs difficiles à détecter)
- affecte la valeur réelle saisie à la variable réelle/rationnelle b ;
- affecte le caractère saisie à la variable caractère c ;
- affiche les valeurs pour montrer les affectations réalisées

2. Faire la trace du programme en considérant que l'utilisateur saisit au clavier : un 1 suivi de la touche *Enter* (= caractère "retour chariot" ou "aller à la ligne"), ensuite 12.2 suivi de *Enter* puis le caractère c suivi de *Enter*.

ligne	a	b	c	affichage (sortie/écriture à l'écran)
initialisation	?	?	?	
16				Entrez un nombre entier puis un nombre réel puis
18	1			
19		12.2		
20			'c'	
22				Vous avez saisi 1 puis 12.2 puis c.

Vous pouvez leur faire remarquer que si la lecture du caractère `c` s'était faite avec `"%c"` au lieu de `" %c"`, la variable `c`, contiendrait `'\n'`, caractère qui suit la chaîne "12.2" dans la mémoire tampon enregistrant tous les caractères effectivement tapés au clavier.

2 Écriture de fonctions

Exercice 2.1 Déclaration et définition de fonctions

Pour les questions suivantes il faut donner la déclaration et la définition de chaque fonction. Vous pouvez faire l'exercice une première fois en donnant uniquement les déclarations, puis le reprendre pour écrire les définitions complètes de chaque fonction.

1. Écrire la fonction `cube` qui prend en entrée un entier et qui renvoie le cube de cet entier.
2. Écrire la fonction `saisie_entier` sans argument, qui demande à l'utilisateur de saisir un nombre entier et le retourne.
3. Écrire la fonction `imprime_adresse` qui prend en argument une variable de type `int` et affiche son adresse dans le format `%p`.
4. Écrire la fonction `est_majeur` qui prend en entrée un entier représentant l'âge en années d'une personne et renvoie `TRUE` si cette personne est majeure et `FALSE` sinon (on considérera que ces deux constantes utilisateurs sont bien déclarées).
5. Écrire la fonction `somme` qui prend en entrée un entier n et qui renvoie $\sum_{i=1}^{i=n} i$.
6. Écrire la procédure `afficher_rectangle` qui prend en entrée deux entiers, largeur et hauteur, et affiche un rectangle d'étoiles de ces dimensions.

```
int cube(int x);
int saisie_entier();
void imprime_adresse(int x);
int est_majeur(int age);
int somme(int n);
void afficher_rectangle(int largeur, int hauteur);
%int binomial(int n, int p);
%nt autre_somme(int n);
\begin{small}
int cube(int x)
{
    return x*x*x;
}

int saisie_entier()
{
    int n = 0;

    printf("Entrez un nombre entier: ");
    scanf("%d", &n);
    return n;
}

void imprime_adresse (int x)
{
    printf("adresse : %p", &x);
}

int est_majeur(int age)
{
    if (age < 18)
    {
        return FALSE;
    }
    /* sinon */
    return TRUE;
}
```

```

}

int somme(int n)
{
    int i;
    int somme = 0;

    for (i = 1; i <= n; i = i + 1)
    {
        somme = somme + i;
    }

    return somme; /* ou bien juste : return n * (n+1) / 2; */
}

void afficher_rectangle(int largeur, int hauteur)
{
    int i; /* lignes */
    int j; /* colonnes */

    for (i = 0; i < hauteur; i = i + 1) /* pour chaque ligne */
    {
        /* afficher largeur etoiles et un saut de ligne */
        for (j = 0; j < largeur; j = j + 1) /* pour chaque colonne */
        {
            printf("*");
        }
        printf("\n");
    }
}

```