

---

## Travaux pratiques 6 : représentation bornée des données ; lecture de données au clavier

---

**Correction.** Note aux chargés de TD.

- Le premier exercice fait référence au cours.
- Le dernier exercice est à réserver à ceux qui ont fini, il n'y a pas de corrections à donner. Un exemple similaire sera peut-être donné à l'examen pour récompenser ceux qui travaillent par eux-mêmes. à voir...
- À nouveau, bien insister sur la procédure permettant d'attaquer un problème de programmation (ils auront à faire ça à l'examen). La procédure :
  - on se donne des exemples
  - on trouve un algorithme en français
  - on traduit l'algorithme en C en s'aidant de commentaires issus de l'algorithme en français
  - on teste sur les exemples qu'on s'est donnés
- N.B. : le "programme vide" voit son écriture simplifiée en n'indiquant plus dans le main la déclaration des variables, ni la valeur de retour. Si certains ont encore des problèmes avec ça, il faut repousser.

### 1 Représentation bornée des données

Il est demandé de taper les deux programmes suivants et d'expliquer la sortie du programme, en s'aidant éventuellement du cours.

1. Exécuter le programme suivant et expliquer sa sortie.

```
/* declaration de fonctionnalités supplémentaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf, scanf */
#include <limits.h> /* INT_MAX, INT_MIN */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    int a;

    printf("INT_MAX = %d\n", INT_MAX);
```

```

    printf("INT_MIN = %d\n",INT_MIN);

    printf("Entrer un entier plus grand que INT_MAX ou plus petit que INT_MIN :\n");
    scanf("%d",&a);
    printf("Vous avez saisi l'entier %d.\n",a);

    return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */

```

**Correction.** La sortie est la suivante (noter que scanf affecte INT\_MAX ou INT\_MIN) :

sur une archi 32 bits (celle des machines de TP normalement):

```

1033$ ./a.out
INT_MAX = 2147483647
INT_MIN = -2147483648
Entrer un entier plus grand que INT_MAX ou plus petit que INT_MIN :
999999999999
Vous avez saisi l'entier 2147483647.
1034$ ./a.out
INT_MAX = 2147483647
INT_MIN = -2147483648
Entrer un entier plus grand que INT_MAX ou plus petit que INT_MIN :
-999999999999
Vous avez saisi l'entier -2147483648.

```

2. Exécuter le programme suivant et expliquer sa sortie.

```

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

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    int i;

    for(i = 0; i <= INT_MAX; i = i + 100000)
    {
        printf("%d\n",i);
    }
    /* i > INT_MAX */
}

```

```

    return EXIT_SUCCESS;
}

```

```

/* implantation de fonctions utilisateurs */

```

**Correction.**  $i = i + 100000$ , c'est pour boucler plus vite.

Ca boucle car  $i$  jamais  $> \text{INT\_MAX}$ . La rep des entiers signés fait que lorsque la retenue est perdue, la rep est celle de  $\text{INT\_MIN}$  (vu en cours). Attention  $\text{INT\_MAX} + 1 == \text{INT\_MIN}$ .

## 2 Mini-calculatrice

Écrire un programme qui demande à l'utilisateur d'entrer une expression simple à évaluer et qui affiche la valeur de l'expression. L'expression à saisir est très simple et suit la forme suivante : `nombre_réel opérateur nombre_réel`, avec opérateur étant un des quatre opérateurs arithmétiques '+', '-', '\*', ou '/'. Deux exemples d'exécution sont les suivants :

```

Entrez une expression de la forme : nombre operateur nombre
15.9 * 1.234
15.9 * 1.234 = 19.6206

```

```

Entrez une expression de la forme : nombre operateur nombre
12 - 12
12 - 12 = 0

```

**Correction.**

```

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

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    double nombre_g; /* membre gauche de l'expression */
    double nombre_d; /* membre droit de l'expression */
    char op; /* operateur */
    double expr; /* resultat de l'expression */

    /* saisie expression */
    printf("Entrez une expression de la forme : nombre operateur nombre\n");
    scanf("%lg",&nombre_g);
    scanf(" %c",&op);
    scanf("%lg",&nombre_d);

```

```

/* calcul valeur expression */
/* cas mutuellement exclusif */
if(op == '+') /* addition */
{
    expr = nombre_g + nombre_d;
}

if(op == '-') /* soustraction */
{
    expr = nombre_g - nombre_d;
}

if(op == '*') /* multiplication */
{
    expr = nombre_g * nombre_d;
}

if(op == '/') /* division */
{
    expr = nombre_g / nombre_d;
}

/* affichage resultat */
printf("%g %c %g = %g\n",nombre_g,op,nombre_d,expr);

return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */

```

### 3 Affichage d'un demi-carré droit d'étoiles

Écrire un programme qui affiche un demi-carré droit d'étoiles de côté spécifié par l'utilisateur. Deux exemples d'exécution sont les suivants :

Entrer la taille du demi-carré :

5

Affichage d'un demi-carre droit d'etoiles de cote 5.

```

*
**
***
****
*****

```

Entrer la taille du demi-carré :

2

Affichage d'un demi-carre droit d'etoiles de cote 2.

\*  
\*\*

### Correction.

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

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    int cote; /* cote du demi-carré droit en nb d'etoiles a saisir par l'utilisateur */
    int i; /* var. de boucle */
    int j; /* var. de boucle */

    /* saisie cote */
    printf("Entrer la taille du demi-carré :\n");
    scanf("%d",&cote);

    /* affichage du demi-carre droit */
    printf("Affichage d'un demi-carre droit d'etoiles de cote %d.\n",cote);

    for(i = 1;i <= cote;i = i + 1) /* chaque numero de ligne d'étoiles */
    {
        /* affiche les blancs */
        for(j = 0;j < cote - i;j = j + 1) /* chaque colonne de blancs */
        {
            /* affiche un blanc */
            printf(" ");
        }
        /* j >= cote - i */

        /* affiche autant d'etoiles que le numero de ligne */
        for(j = 0;j < i;j = j + 1) /* chaque colonne d'etoiles */
        {
            /* affiche une etoile */
            printf("*");
        }
        /* j >= i */

        /* passe a la ligne suivante */
        printf("\n");
    }
}
```

```
    }  
    /* i > cote */  
  
    return EXIT_SUCCESS;  
}  
  
/* implantation de fonctions utilisateurs */
```