

```

/*
Exercice 1. Echange de deux valeurs
Ecrire une fonction Echange qui réalise l'échange de deux valeurs entières passées en
argument. */

#include <stdio.h>

void Echange(int *, int *);

int main(void) {
    int x, y;

    printf("Entrez deux entiers: ");
    scanf("%d%d", &x, &y);
    printf("\nAvant d'appeler fonction Echange:\n\tx = %d\n\ty = %d\n", x, y);
    Echange(&x, &y);
    printf("\nAprès d'appeler fonction Echange:\n\tx = %d\n\ty = %d\n", x, y);

    return 0;
}

void Echange(int *a, int *b) {
    int temp=*a;
    *a = *b;
    *b = temp;
}

/* Exercice 2. Lecture contrôlée d'un entier strictement positif
Ecrire une fonction void LirePositif(int *n) qui réalise la lecture contrôlée d'un
entier strictement positif. Cette fonction affiche un message indiquant qu'une valeur
positive est attendue et répète la lecture jusqu'à ce que l'utilisateur entre un entier
positif. On pourra utiliser une variable locale pour la lecture de l'entier au clavier. */

#include <stdio.h>

void LirePositif(int *);

int main(void) {
    int value;

    LirePositif(&value);

    printf("\nLa valeur entree est %d\n", value);

    return 0;
}

void LirePositif(int *valeur) {
    do {
        printf("Entrez un entier positif ?: ");
        scanf("%d", valeur);
        if ((*valeur) < 0) {
            printf(" <--- %d est une valeur negative --->\n", (*valeur));
        }
    } while ((*valeur) < 0);
}

```

```

/* Exercice 3. Valeurs minimum et maximum d'une liste de n valeurs entières
• Comment une fonction peut-elle transmettre plus d'une seule valeur la partie du
programme qui l'a appelle ?
• Ecrire une fonction MinMax qui transmet à la partie appelante le minimum et le
maximum de n entiers lus au clavier, où n est un paramètre de la fonction.
• Ecrire un programme permettant de lire un entier positif n d'afficher le minimum
et le maximum de n valeurs lues au clavier.
*/

```

```

#include <stdio.h>

```

```

void MinMax(int n, int *min, int *max);
void LirePositif(int *valeur);

```

```

int main(void) {
    int n, min, max;

    LirePositif(&n);
    MinMax(n, &min, &max);
    if (n) {
        printf("Le minimum et Le maximum de %d valeurs lues au clavier sont:\n", n);
        printf("\tLe minimum est %d\n\tLe maximum est %d\n", min, max);
    }
    return 0;
}

```

```

void MinMax(int n, int *min, int *max) {
    int valeur_lue, compteur=0;

    if (n==0) { printf("\n<--- n = 0 --->\n\n"); return; }
    printf("\nEntrez %d valeurs entières\n", n);
    printf("Entrez la valeur N°%d ?--> ", ++compteur); scanf("%d",&valeur_lue);
    *min = valeur_lue;
    *max = valeur_lue;
    while (compteur < n) {
        printf("Entrez la valeur N°%d ?--> ", ++compteur);
        scanf("%d",&valeur_lue);
        if (valeur_lue < (*min)) *min = valeur_lue;
        if (valeur_lue > (*max)) *max = valeur_lue;
    }
}

```

```

void LirePositif(int *valeur) {
    do {
        printf("Entrez un entier positif ?: ");
        scanf("%d", valeur);
        if ((*valeur) < 0) {
            printf(" <--- %d est une valeur negative --->\n", (*valeur));
        }
    } while ((*valeur) < 0);
}

```

```

/* Exercice 4. Simplification de deux nombres représentant une fraction
• Ecrire une fonction Simplifie qui reçoit deux valeurs en paramètre représentant
le numérateur et le dénominateur d'une fraction et transforme ces valeurs en
numérateur et dénominateur d'une fraction équivalente simplifiée. Par exemple,
si les variables données initialement ont pour valeur 3 et 12, leur valeur après
exécution de la procédure sera 1 et 4 (  $3/12 = 1/4$  ). On supposera dans ce cas que
la procédure reçoit une fraction valide (c'est-à-dire, que le dénominateur n'est pas
nul).
• Ecrire une fonction qui réalise le même travail que la fonction Simplifie mais qui
indique une erreur grâce à la valeur retournée si le dénominateur de la première
fraction est nul (!= 0 si le déroulement est correct, 0 en cas d'erreur).
*/

```

```

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

```

```

int Simplifie(int *, int *);

```

```

int main(void) {
    int numérateur, dénominateur;

    printf("Entrez le numérateur : "); scanf("%d", &numérateur);
    printf("Entrez le dénominateur : "); scanf("%d", &dénominateur);

    if (Simplifie(&numérateur, &dénominateur)==EXIT_SUCCESS)
        printf("\nLa fraction simplifiée = %d/%d\n", numérateur, dénominateur);
    else
        printf("\n<--- Error - Simplifie() --->\n\n");

    return 0;
}

```

```

int Simplifie(int *numérateur, int *dénominateur) {
    int diviseur=2;

    if (*dénominateur != 0) {
        while ((*numérateur >= diviseur) && (*dénominateur >= diviseur)) {
            if ((*numérateur % diviseur == 0) && (*dénominateur % diviseur == 0)) {
                *numérateur /= diviseur;
                *dénominateur /= diviseur;
            }
            else
                diviseur++;
        }
        return EXIT_SUCCESS;
    }
    else
        return EXIT_FAILURE;
}

```