

Exercice 1 : Macros sans paramètres

- (a) Quelle sera la valeur affichée par le programme ci-dessous ?

```
#include <stdio.h>

#define ONE 1
#define NINE 9
#define TEN ONE + NINE

int main() {
    printf("%d\n", NINE * TEN);
}
```

L'expression `NINE * TEN` est évaluée comme `9 * 1 + 9`, ce qui donne 18.

- (b) Quelles modifications faut-il apporter pour que le résultat soit correct ?

Il est nécessaire de rajouter des parenthèses dans la définition de la macro.

```
#define TEN (ONE + NINE)
```

Exercice 2 : Macros avec paramètres

- (a) Expliquer pourquoi ce programme affiche 5, et comment corriger ce problème.

```
#include <stdio.h>
#define MAX(i, j) i > j ? i : j

int main(void) { printf("%d\n", MAX(5, 2) + 4); }
```

L'appel de la macro `MAX(5,2)` est évalué comme `5 > 2 ? 5 : 2`. Mais l'addition qui suit donne l'expression : `5 > 2 ? 5 : 2 + 4`, soit `true ? 5 : 6` en raison des priorités des opérateurs. Afin de résoudre ce problème il est nécessaire de protéger par des parenthèses toutes les variables d'une macro.

```
#define MAX(i, j) ((i) > (j) ? (i) : (j))
```

- (b) Expliquez pourquoi ce programme affiche la valeur 7.

```
#define ABS(x) ((x) >= 0 ? (x) : -(x))

int main(void)
{
    int i = 5;
    int res = ABS(++i);
    printf("%d\n", res);
}
```

Une fois de plus le préprocesseur agit comme du remplacement de chaînes, et donc l'expression `ABS(++i)` est évaluée comme `((++i) >= 0 ? (++i) : -(++i))`. Comme `++i` est supérieur à 0, l'opérateur conditionnel évalue et retourne la première expression soit `++i`. Au final, `i` aura été incrémenté deux fois.

Il n'existe pas de solution pour éviter ce genre de problème, en revanche on prendra l'habitude de bien saisir les macros en MAJUSCULES pour bien les identifier, et éviter de donner à une macro des expressions qui peuvent poser problème.

L'approche correcte consiste à ne pas utiliser de macros, mais plutôt utiliser une fonction **inline**.

```
inline int abs_int(int x) {
    return x >= 0 ? x : -x;
}
```

Exercice 3: Compilation conditionnelle

Qu'affichent les programmes ci-dessous ?

(a)

```
#include <stdio.h>

#define TEST 0

int main(void)
{
    #ifdef TEST
        printf("#ifdef\n");
    #else
        printf("#else\n");
    #endif
    #ifndef TEST
        printf("#ifndef\n");
    #endif
    #if !defined TEST
        printf("#if !defined\n");
    #endif
    #if defined TEST
        printf("#if defined\n");
    #endif
    #if TEST
        printf("#if\n");
    #endif
}
```

```
#ifdef
#if defined
```

(b)

```
#include <stdio.h>

//#define TEST 0

int main()
{
    #ifdef TEST
        printf("#ifdef\n");
    #else
        printf("#else\n");
    #endif
    #ifndef TEST
        printf("#ifndef\n");
    #endif
}
```

```
#if !defined TEST
    printf("#if !defined\n");
#endif
#if defined TEST
    printf("#if defined\n");
#endif
#if TEST
    printf("#if\n");
#endif
}
```

```
#else
#ifndef
#if !defined
```

Exercice 4: Lecture de code

Calcul du taux d'intérêt.

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

#define ZONE_SWISS

#ifdef ZONE_SWISS
#define ZONE_NAME "Suisse"
#define CURRENCY "CHF"
#elif defined ZONE_EUROPE
#define ZONE_NAME "Europe"
#define CURRENCY "EUR"
#elif defined ZONE_AFRICA
#define ZONE_NAME "Afrique"
#define CURRENCY "CFA"
#elif defined ZONE_USA
#define ZONE_NAME "États-Unis"
#define CURRENCY "USD"
#else
#error Ce programme n'est pas compatible pour cette zone géographique
#endif

int main(void)
{
    double capital;
    printf("Programme de calcul des intérêts au %s pour la %s.\n",
        __DATE__, ZONE_NAME
    );
    printf("Capital en %s ?", CURRENCY);
    scanf("%lf", &capital);
    printf("Taux d'intérêts annuel en %% \? : ");
    double interest_rate;
    scanf("%lf", &interest_rate);
    printf("Montant annuel des intérêts : %g %s\n",
        capital * interest_rate / 100,
        CURRENCY);
}
```

- (a) Que fait ce programme ? Que voit l'utilisateur à l'écran ?

```
echo "1000\n5\n" | ./a.out
```

```
Programme de calcul des intérêts au 2022-04-11 pour la Suisse.
Capital en CHF ? 1000
Taux d'intérêts annuel % \? 5
Montant annuel des intérêts : 50 CHF
```

- (b) Que faut-il faire pour adapter rapidement ce programme à une autre zone géographique ?

| Il suffit de remplacer la définition du symbole `ZONE_SWISS` par le symbole d'une autre zone.

- (c) Quel changement y aura-t-il si on remplace la définition `ZONE_SWISS` par `ZONE_AFRICA` ?

| Le nom de la zone géographique affichée sera **Afrique** et l'unité monétaire utilisée sera **CFA**.

- (d) Quel changement y aura-t-il si on remplace la définition `ZONE_SWISS` par `ZONE_JAPAN` ?

Le programme n'a pas été prévu pour fonctionner avec cette zone. Lors de la compilation du programme un message d'erreur sera affiché.

(e) Comment compiler le programme depuis la ligne de commande pour n'importe quelle zone ?

Il faut tout d'abord retirer la définition 'ZONE_SWISS' du programme, puis compiler en utilisant par exemple :

```
cc -DZONE_SWISS interest.c -ointerest
```