

## Revision intra :

### Question 1 -

Écrire un programme qui

- définit un type struct **matrix** permettant de représenter une matrice carrée de doubles (on suppose ici que l'ordre de la matrice est **10**);
- définit une fonction **initialize\_matrix(m,v)** qui reçoit une matrice **m** et initialise chacune de ses entrées à **v**;
- définit une fonction **print\_matrix(m)** permettant d'afficher le contenu de la matrice sur stdout.

(1 points) **struct** matrix {

};

(2 points) **void** initialize\_matrix(**struct** matrix \*m, **double** v)

(2 points) **void** print\_matrix(**struct** matrix \*m)

## Question 2

Complétez la fonction incomplète dans le fichier suivant :

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdbool.h>

/**
 * retourne l'année dans la chaîne reçue
 * @param : une chaîne de caractères contenant une date au format ISO 8601 (YYYY-MM-JJ)
 * @return: un entier correspondant à l'année de la date fournie, ou -1 si l'année est invalide.
 *         L'année est invalide si sa longueur est différente de 4 caractères ou si elle contient
 *         des caractères autres que des chiffres.
 */
int get_year(char s[]) {
    // TODO //
}

// Retourne vrai si la date reçue est valide. Faux sinon
bool is_valid_year(char s[]) {
    // TODO //
}

int main(void) {
    char *date = "2023-03-06";
    printf("Result :%d ", get_year(date));
    return 0;
}
```

(5 points) `bool is_valid_year (char s[])`

(5 points) `int get_year (char s[])`

(5 points) `int get_year (char s[])`

**Question 1.** ..... (15 points)

(a) (5 points) Qu'est-ce qui est affiché à l'écran par ce programme?

```
#include <stdio.h>

int main() {
    int a = 44;
    int b = 66;
    int *pa, *pb;
    pa = &a;
    pb = &b;
    printf("%d %d\n", *pa, *pb);
    *pa = *pb;
    printf("%d %d\n", a, b);
}
```

(a) \_\_\_\_\_

(b) (5 points) Qu'est-ce qui est affiché à l'écran par ce programme?

```
#include <stdio.h>

void main() {
    int value = 77, num = 99;
    int *pv = &value, *pn = &num;
    int **ppi;
    ppi = &pv;
    printf("%d\n", **ppi);
    ppi = &pn;
    printf("%d\n", **ppi);
}
```

(b) \_\_\_\_\_

(c) (5 points) Que fait la fonction suivante?

```
int mystere(int x) {
    int i = 0;
    while (x != 0) {
        x &= x + (x & -x);
        i++;
    }
    return i;
}
```

**Question 2.** .....(20 points)

Considérez le programme suivant (qui compile sans erreur ni avertissement) :

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

char *repeter1(char c, int n) {
    int i;
    char *s;
    for (i = 0; i < n; ++i) {
        s[i] = c;
    }
    return s;
}

char *repeter2(char c, int n) {
    int i;
    char *s = (char*)malloc(n * sizeof(char));
    for (i = 0; i < n; ++i) {
        s[i] = c;
    }
    return s;
}

void repeter3(char c, int n, char *s) {
    int i;
    for (i = 0; i < n; ++i) {
        s[i] = c;
    }
}

int main() {
    printf("%s\n", repeter1('*', 10));
    printf("%s\n", repeter2('*', 10));
    char s[10];
    repeter3('*', 10, s);
    printf("%s\n", s);
}
```

- (a) (15 points) Les fonctions `repeter1`, `repeter2` et `repeter3` présentent chacune un ou plusieurs problèmes. Quels sont-ils ?
- (b) (5 points) Proposez une fonction `repeter` qui corrige chacun de ces problèmes et montrez un exemple d'utilisation correcte de votre fonction.

**Question 6.** ..... (30 points)

- (a) (5 points) Quel est le résultat affiché par le programme suivant ?

```
#include <stdio.h>

int main() {
    int a[2][3] = {9,8,7,6,5,4};
    printf("%d\n", *((a+1)+1));
    return 0;
}
```

(a) \_\_\_\_\_

- (b) (5 points) Vrai ou faux ? En supposant
- $\text{sizeof}(\text{int}) = 2$
- et
- $\text{sizeof}(\text{float}) = 4$
- , la valeur
- $\text{sizeof}(\text{union bidon})$
- est nécessairement supérieure ou égale à 8.

```
union bidon {
    struct essence {
        float x;
        int y;
    };
    struct machin {
        float z;
        float t;
    };
    float u;
};
```

(b) \_\_\_\_\_

- (c) (5 points) Quel est le résultat affiché par le programme suivant ?

```
#include <stdio.h>

int main() {
    char s[10] = "+---+---+++";
    int i;

    for (i = 0; i < 7; ++i) {
        if (s[i] != s[i+1]) {
            s[i+2] = s[i];
        }
    }
    printf("%s\n", s);
    return 0;
}
```

(c) \_\_\_\_\_

(d) (5 points) Quel est le résultat affiché par le programme suivant ?

```
#include <stdio.h>

int x = 9;

int f(int x) {
    int y = 1;
    x += 4;
    x *= y;
    return x;
}

int main() {
    int x = 10;
    f(x);
    printf("%d\n", x);
    return 0;
}
```

(d) \_\_\_\_\_

(e) (5 points) Quel est le résultat affiché par le programme suivant ?

```
#include <stdio.h>

#define CUBE(x) x * x * x

int main() {
    int x = 2, y = 3, z;
    z = y - CUBE(x++);
    printf("%d\n", x + y + z);
    return 0;
}
```

(e) \_\_\_\_\_

**Question 7.** ..... (10 points)

Le plus grand commun diviseur de deux nombres naturels  $a$  et  $b$  est défini comme le plus grand nombre naturel  $p$  qui est à la fois un diviseur de  $a$  et de  $b$ . On le note par  $\text{pgcd}(a, b)$ .

Voici quelques exemples :

$$\begin{array}{ll} \text{pgcd}(2, 7) = 1 & \text{pgcd}(14, 21) = 7 \\ \text{pgcd}(1, 8) = 1 & \text{pgcd}(0, 8) = 8 \end{array}$$

Notez que  $\text{pgcd}(0, 0)$  n'est pas bien défini (ou est égal à  $+\infty$ , car tout nombre naturel divise 0).

Par ailleurs, la fonction  $\text{pgcd}$  satisfait la propriété que si  $a \geq b$ , alors  $\text{pgcd}(a, b) = \text{pgcd}(a - b, b)$ . Il est donc possible de définir un algorithme récursif basé sur la formule suivante :

$$\text{pgcd}(a, b) = \begin{cases} a & \text{si } b = 0, \\ b & \text{si } a = 0, \\ \text{pgcd}(a - b, b) & \text{si } a \geq b \end{cases}$$

Donnez l'implémentation d'une fonction C dont le prototype est

```
int pgcd(int a, int b);
```

qui calcule le plus grand commun diviseur de deux nombres naturels  $a$  et  $b$ . Vous pouvez supposer que  $a$  et  $b$  ne sont pas tous deux nuls.

**Attention !** vous devez traiter le cas où  $a < b$ . *Indice* : la fonction  $\text{pgcd}$  est symétrique, c'est-à-dire qu'on a  $\text{pgcd}(a, b) = \text{pgcd}(b, a)$ , peu importe  $a$  et  $b$ .