

Exercice 1 : Lecture de fichier sur le *heap*

On souhaite lire un fichier dont la taille n'est pas connue avant l'exécution du programme. Un espace mémoire doit alors être réservé sur le *heap*. Écrire la fonction `load` permettant de charger l'ensemble du fichier en mémoire dans la variable `data`.

```
int main(int argc, char*argv[]) {
    assert(argc > 1);
    FILE *fp = fopen(argv[1], "r");
    assert(fp != NULL);
    char *data;
    load(&data, fp);
    fclose(fp);
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 2 : Membre flexible

Une structure avec un membre flexible est défini comme suit :

```
typedef struct point {
    int x; int y;
} Point;

typedef struct record {
    int id;
    Point points[];
} Record;
```

(a) Quelle est la taille de la structure `record` en bytes ?

.....

.....

(b) Déclarez la variable `r` puis allouer l'espace mémoire nécessaire pour y stocker 10 points.

.....

.....


```
typedef struct measurements {  
    size_t elements; // Nombre d'éléments dans le set  
    size_t capacity; // Capacité en éléments de l'espace alloué  
    int data[];  
} Measurements;
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 4: Sudoku

Dans un jeu de Sudoku, on souhaite allouer l'espace nécessaire pour y stocker une grille dont la taille (nombre de colonnes) est déterminée à l'exécution du programme.

La grille est stockée dans un pointeur sur un tableau à deux dimensions, dont chaque dimension est la taille de la variable `columns`.

- (a) Écrire la structure de donnée et allouer cette grille en initialisant toutes les valeurs à zéro.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- (b) Affichez la grille sur la sortie standard.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....