

# Projet

## 1 Introduction

L'objectif de ce projet est de créer un programme `magicsquare` qui permette de résoudre un carré magique. Un carré magique est un carré de  $n$  lignes et  $n$  colonnes où chaque case comporte un nombre. Leur disposition est telle que la somme des nombres de chaque ligne, colonne et diagonale donne une constante appelée constante magique. Cette constante magique vaut  $(n \times (n^2 + 1))/2$  où  $n$  est le nombre de lignes (ou de colonnes) du carré considéré. Dans ce projet, on se focalisera exclusivement sur les carrés magiques de  $n$  lignes et  $n$  colonnes où  $n$  est un nombre pair non divisible par 4.

Votre programme devra admettre un argument qui correspond à la taille du carré magique. Par exemple :

```
./magicsquare 10
```

devra résoudre un carré magique de taille 10x10.

## 2 Algorithme de résolution

Pour résoudre un carré magique d'ordre pair, il suffit de le découper en 4 quadrants de même taille. Par exemple, un carré magique 6x6 pourra être découpé en 4 quadrants de 3x3 comme illustré sur la figure 1. Chaque quadrant se voit attribuer un intervalle de nombres (de 1 à 9 pour le quadrant supérieur gauche, de 10 à 18 pour le quadrant inférieur droit, cf. figure 1). Il faut ensuite résoudre chaque quadrant indépendamment comme étant un carré magique d'ordre impair avec la méthode dite *siamoise* : il faut commencer par écrire le premier nombre de la plage assignée au quadrant dans la case centrale de la première ligne. Vous devez ensuite remplir les autres cases en vous déplaçant d'une ligne vers le haut et d'une colonne vers la droite. Il y a 3 exceptions à prendre en compte après un déplacement :

- si vous vous retrouvez au-dessus de la première ligne du quadrant, restez dans la même colonne mais placez le nombre dans la dernière ligne du quadrant ;
- si vous vous retrouvez à droite de la dernière colonne du quadrant, restez dans la même ligne mais placez le nombre dans la première colonne ;
- si vous vous retrouvez sur une case déjà complétée, il faut revenir à la dernière case que vous avez complétée et écrire le nombre dans la même colonne mais sur la ligne en dessous.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 8  | 1  | 6  | 26 | 19 | 24 |
| 3  | 5  | 7  | 21 | 23 | 25 |
| 4  | 9  | 2  | 22 | 27 | 20 |
| 35 | 28 | 33 | 17 | 10 | 15 |
| 30 | 32 | 34 | 12 | 14 | 16 |
| 31 | 36 | 29 | 13 | 18 | 11 |

FIGURE 1 – carré magique 6x6 découpé en 4 quadrants 3x3

Ensuite il faut échanger des cases entre la partie supérieure et la partie inférieure. Le nombre et la position des cases à échanger dépend de la taille des quadrants. Dans le cas d'un carré magique 6x6, il faut échanger les 3 cases rouges avec les 3 cases bleues comme illustré sur la figure 2. Le carré magique 6x6 final est illustré sur la figure 3. Vous trouverez en annexe des exemples pour des carrés magiques de taille 10x10 et 14x14 afin de mettre en évidence le schéma d'échange des cases et les résultats finaux.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 8  | 1  | 6  | 26 | 19 | 24 |
| 3  | 5  | 7  | 21 | 23 | 25 |
| 4  | 9  | 2  | 22 | 27 | 20 |
| 35 | 28 | 33 | 17 | 10 | 15 |
| 30 | 32 | 34 | 12 | 14 | 16 |
| 31 | 36 | 29 | 13 | 18 | 11 |

FIGURE 2 – cases à échanger

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 35 | 1  | 6  | 26 | 19 | 24 |
| 3  | 32 | 7  | 21 | 23 | 25 |
| 31 | 9  | 2  | 22 | 27 | 20 |
| 8  | 28 | 33 | 17 | 10 | 15 |
| 30 | 5  | 34 | 12 | 14 | 16 |
| 4  | 36 | 29 | 13 | 18 | 11 |

FIGURE 3 – carré magique final

### 3 Spécifications

Votre programme devra déléguer la résolution d'un quadrant à un processus unique. Par exemple, un carré magique 6x6 devra être découpé en 4 quadrants 3x3, chaque quadrant étant attribué à un processus. Lorsqu'un processus a terminé la résolution de son quadrant, il transmet au processus père le contenu du quadrant via un tube anonyme. Il ne devra y avoir qu'un seul tube anonyme entre le père et tous les fils. Lorsque le père souhaite récupérer le quadrant calculé par un fils, il doit lui envoyer un signal (API POSIX), puis se mettre en attente de lecture dans le tube. Après la récupération des données de chaque fils, le père sera en charge d'écrire le résultat du carré magique sur la sortie standard. Les structures de données utilisées pour stocker les nombres du carré magique devront obligatoirement être allouées à l'aide de `malloc`.

Pour simplifier l'exercice, en cas d'erreur, on arrêtera l'exécution avec un message approprié.

Vous respecterez scrupuleusement le modèle d'affichage utilisé dans l'exemple, de manière à ce que les tests puissent être automatisés (voir les jeux de tests présents dans le `Makefile` disponible sur Moodle). Vous n'utiliserez aucun fichier intermédiaire. Bien sûr, vous n'utiliserez que les primitives systèmes pour tout ce qui est gestion des tubes, des processus et des signaux.

### 4 Exemple

```
turing> ./magicsquare 6
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
turing>
```

### 5 Travail demandé

On demande :

- de réaliser le programme `magicsquare` ;
- d'enrichir le `Makefile` fourni avec vos propres jeux de tests, pour vérifier que votre programme est bien conforme aux spécifications et pour atteindre la meilleure mesure de couverture de votre code ;
- de rédiger un rapport décrivant l'architecture générale de votre solution, les structures de données utilisées et les informations transmises entre les processus et leur format.

Votre implémentation doit fonctionner sur `turing`. Vous vous attacherez à respecter les spécifications et à le prouver par vos jeux de tests, ainsi qu'à veiller à ne pas avoir de fuite de mémoire.

Vous noterez qu'un `Makefile` est disponible sur Moodle pour tester votre programme. Il crée un exécutable à partir du fichier source `.c` unique se trouvant dans le répertoire courant, puis lance une série de jeux de tests.

Ces tests ne sont pas exhaustifs mais ils vérifient que votre programme fonctionne conformément à l'énoncé. Si l'un des tests présents dans le `Makefile` échoue, vous pouvez vérifier la raison de l'erreur et/ou rejouer le test à la main : les commandes exécutées sont affichées en clair dans le terminal. N'hésitez pas à enrichir ce `Makefile` de nouveaux tests pour qu'il soit plus complet, notamment pour assurer la meilleure couverture possible de votre code. Les tests supplémentaires devront obligatoirement se trouver à la fin du `Makefile` fourni dans une cible unique. Le nom de cette cible devra être ajouté à la variable `TESTSUP`.

## 6 Modalités de remise

Le projet est à réaliser individuellement. Votre projet doit contenir :

- le fichier source ;
- le rapport au format PDF ;
- le `Makefile` enrichi de vos jeux de tests et la mesure de couverture obtenue avec `gcov`.

Vous déposerez votre projet, débarrassé de tout fichier binaire autre que votre rapport, sous forme d'une archive au format `tar.gz` sur Moodle dans l'espace prévu à cet effet, avant le mercredi 2 mai 2018 à 20 h.

## 7 Annexes

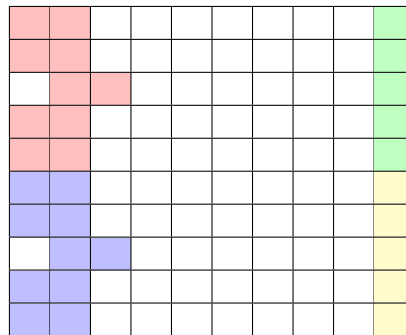


FIGURE 4 – carré magique 10x10 - cases rouges à échanger avec les bleues, cases vertes à échanger avec les jaunes

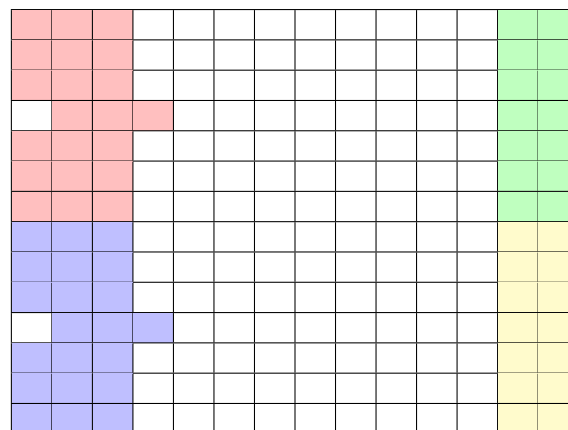


FIGURE 5 – carré magique 14x14 - cases rouges à échanger avec les bleues, cases vertes à échanger avec les jaunes

|    |    |     |    |    |    |    |    |    |    |
|----|----|-----|----|----|----|----|----|----|----|
| 92 | 99 | 1   | 8  | 15 | 67 | 74 | 51 | 58 | 40 |
| 98 | 80 | 7   | 14 | 16 | 73 | 55 | 57 | 64 | 41 |
| 4  | 81 | 88  | 20 | 22 | 54 | 56 | 63 | 70 | 47 |
| 85 | 87 | 19  | 21 | 3  | 60 | 62 | 69 | 71 | 28 |
| 86 | 93 | 25  | 2  | 9  | 61 | 68 | 75 | 52 | 34 |
| 17 | 24 | 76  | 83 | 90 | 42 | 49 | 26 | 33 | 65 |
| 23 | 5  | 82  | 89 | 91 | 48 | 30 | 32 | 39 | 66 |
| 79 | 6  | 13  | 95 | 97 | 29 | 31 | 38 | 45 | 72 |
| 10 | 12 | 94  | 96 | 78 | 35 | 37 | 44 | 46 | 53 |
| 11 | 18 | 100 | 77 | 84 | 36 | 43 | 50 | 27 | 59 |

FIGURE 6 – carré magique 10x10 final

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 177 | 186 | 195 | 1   | 10  | 19  | 28  | 128 | 137 | 146 | 99  | 108 | 68  | 77  |
| 185 | 194 | 154 | 9   | 18  | 27  | 29  | 136 | 145 | 105 | 107 | 116 | 76  | 78  |
| 193 | 153 | 155 | 17  | 26  | 35  | 37  | 144 | 104 | 106 | 115 | 124 | 84  | 86  |
| 5   | 161 | 163 | 172 | 34  | 36  | 45  | 103 | 112 | 114 | 123 | 132 | 85  | 94  |
| 160 | 162 | 171 | 33  | 42  | 44  | 4   | 111 | 113 | 122 | 131 | 140 | 93  | 53  |
| 168 | 170 | 179 | 41  | 43  | 3   | 12  | 119 | 121 | 130 | 139 | 141 | 52  | 61  |
| 169 | 178 | 187 | 49  | 2   | 11  | 20  | 120 | 129 | 138 | 147 | 100 | 60  | 69  |
| 30  | 39  | 48  | 148 | 157 | 166 | 175 | 79  | 88  | 97  | 50  | 59  | 117 | 126 |
| 38  | 47  | 7   | 156 | 165 | 174 | 176 | 87  | 96  | 56  | 58  | 67  | 125 | 127 |
| 46  | 6   | 8   | 164 | 173 | 182 | 184 | 95  | 55  | 57  | 66  | 75  | 133 | 135 |
| 152 | 14  | 16  | 25  | 181 | 183 | 192 | 54  | 63  | 65  | 74  | 83  | 134 | 143 |
| 13  | 15  | 24  | 180 | 189 | 191 | 151 | 62  | 64  | 73  | 82  | 91  | 142 | 102 |
| 21  | 23  | 32  | 188 | 190 | 150 | 159 | 70  | 72  | 81  | 90  | 92  | 101 | 110 |
| 22  | 31  | 40  | 196 | 149 | 158 | 167 | 71  | 80  | 89  | 98  | 51  | 109 | 118 |

FIGURE 7 – carré magique 14x14 final