# Избыточность (Redundancy)

В системе присутствует избыточное количество необходимых вам компонент. И при падении одного из таких избыточных компонент, все должно продолжать работать. При таком подходе проектирования можно выделить две стратегии:

- active-active
- active-passive

https://habrahabr.ru/post/118496/

# Active-Active

Одновременная работа с двумя идентичными компонентами.

Например в системе, где клиент получает цены котировок, он может подписаться сразу в два разных компонента и получать цены одновременно из двух мест. Если один из таких компонентов упадет, то клиент вообще не заметит, что в системе случилась какая-то проблема, что является несомненным плюсом данного подхода.

# Active-Passive

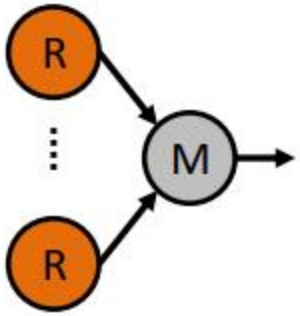
Данная стратегия, представляет собой только один постоянно рабочий компонент, в случае падения которого, автоматически поднимается второй, восстанавливает состояние и берет на себя всю работу.

http://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-redundancy-terminology-20130115-v01.pdf

# Activation of Redundancy

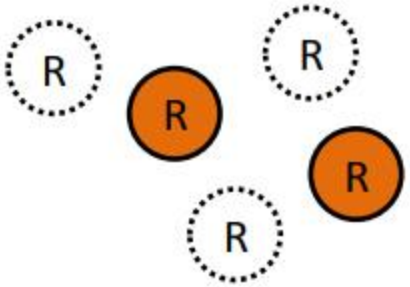
- ***Static** Redundancy*:

  Redundancy (redundant streams, links, topologies, …) is continuously used by the service of interest, regardless whether faults are present or not. Faults are tolerated by e.g. *fault masking.*




- ***Dynamic** Redundancy*:

  Redundancy is activated on demand by a service of interest in presence of faults typically after *reconfiguration*.



http://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-redundancy-terminology-20130115-v01.pdf

# Static vs. Dynamic Redundancy

| | Static Redundancy | Dynamic Redundancy |
|---|---|---|
| **Resource utilization** | Resource intensive:<br>Resources are always use, regardless whether faults are present or not.<br>Faults are tolerated by *fault masking* | Resource friendly:<br>Resources are used on demand in presence of faults by *reconfiguration* |
| **Timing behavior** | Requires no failover time, i.e. the time consumption is low | Additional failover time required |
| **Reliability** | Provides highest short term reliability | Provides high long-term reliability |

- *Hybrid Redundancy*:
  Comprises various mixed forms of *static* and *dynamic* redundancy to overcome drawbacks of *static* and *dynamic* redundancy.
  Example:
  *Fault masking* until a certain threshold, followed by *reconfiguration*.

http://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-redundancy-terminology-20130115-v01.pdf

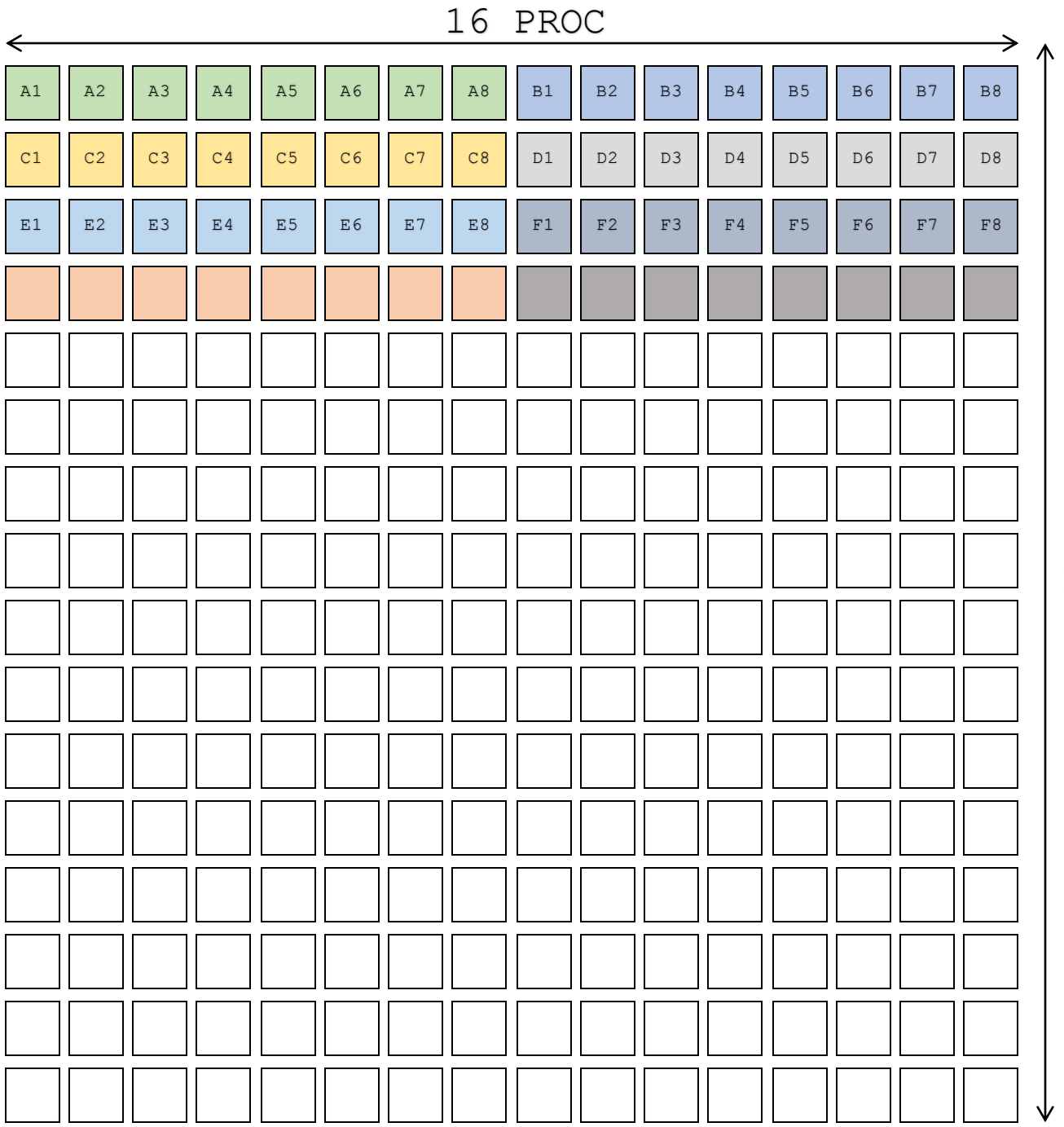
# Types of Redundancy

- ***Structural** Redundancy*:
  Multiple resources are used to provide redundancy, e.g. sending a frame twice via disjoint links, ports, paths, …

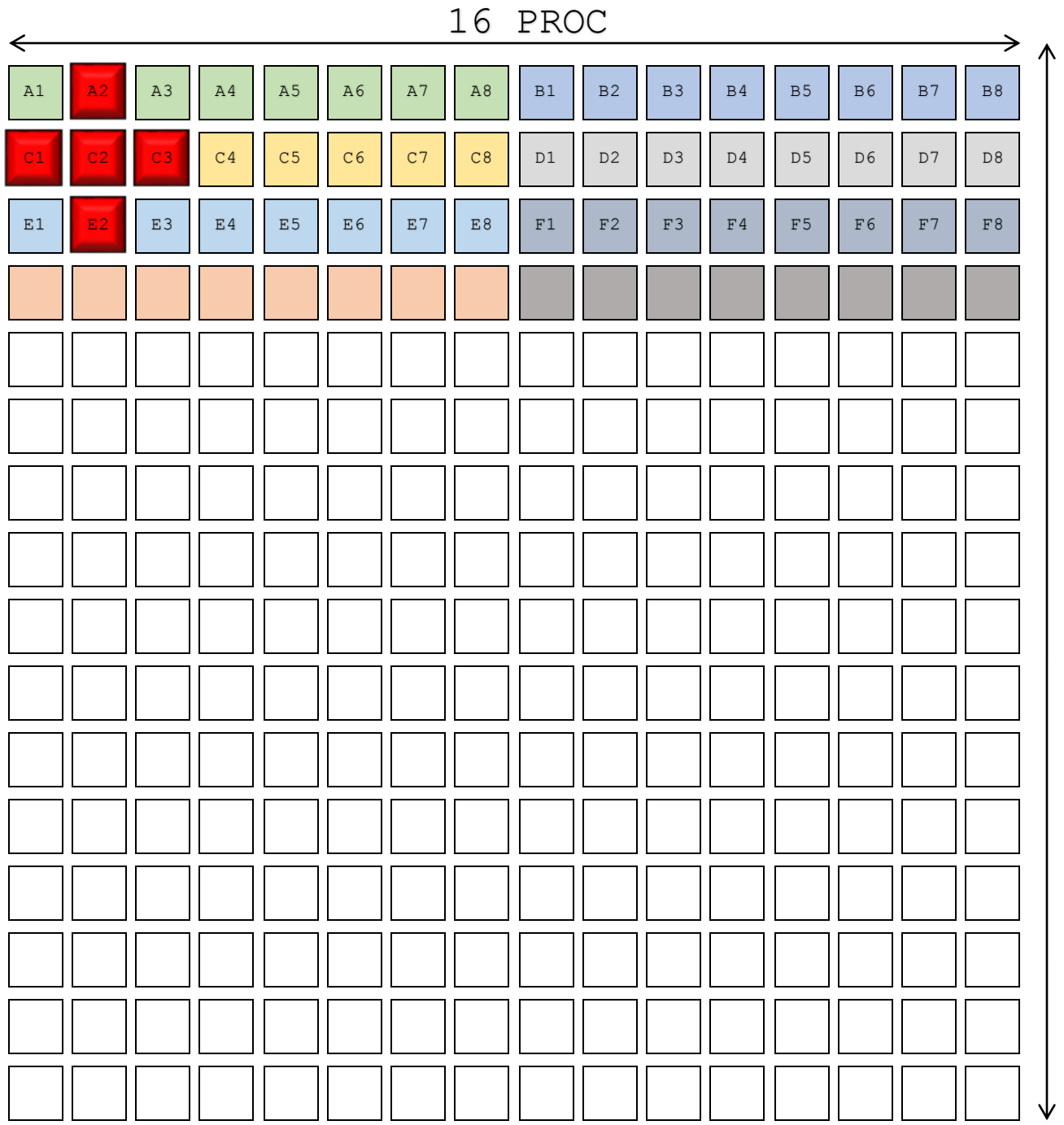

- ***Time** Redundancy*:
  One resource is used longer to provide redundancy, e.g. sending a frame twice via one link, port, path, …


- Both types can appear in a restricted form (cmp. [La], shown here for time redundancy).

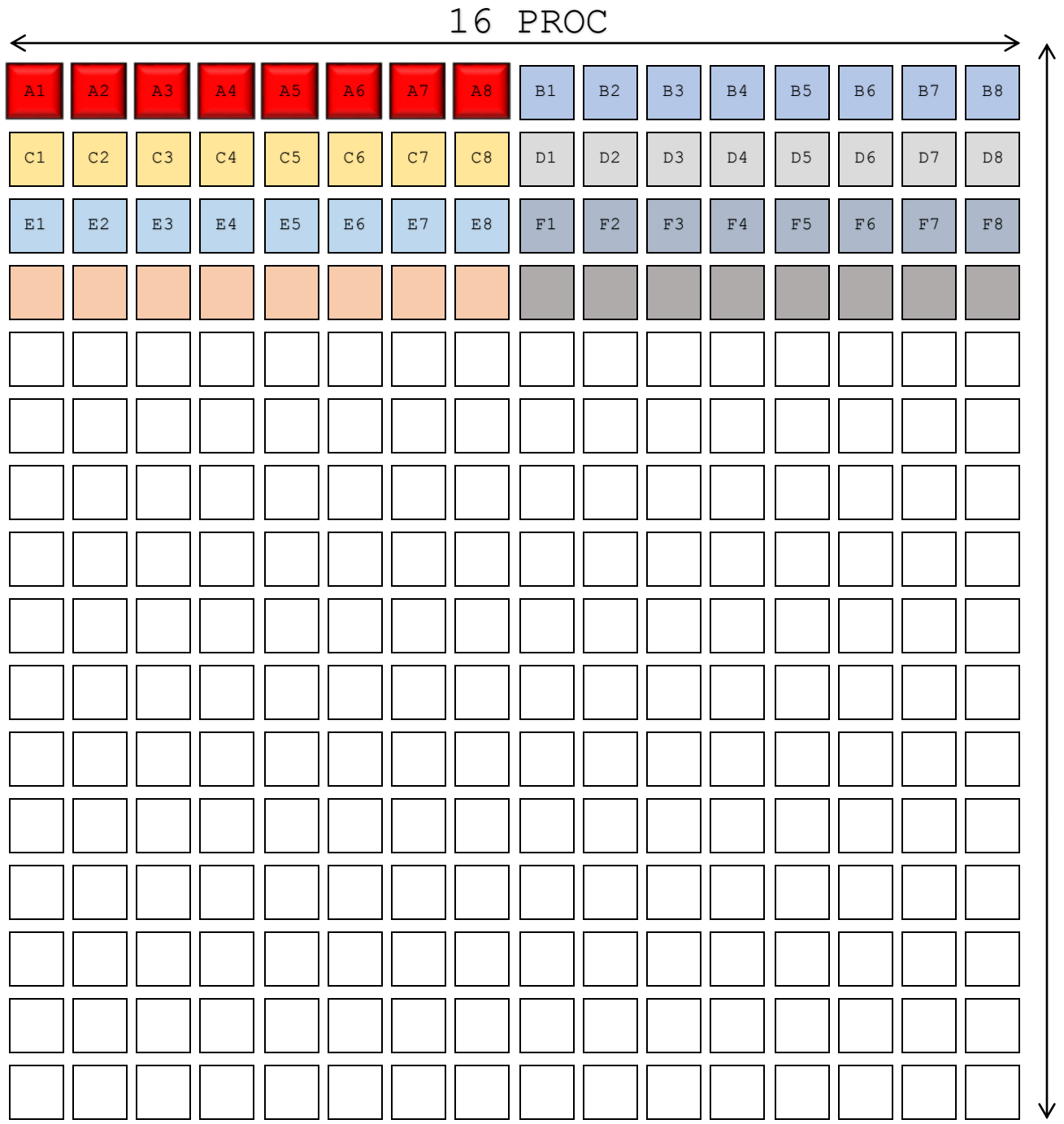

- For specific mechanisms, time- and structural redundancy are not mutual exclusive, e.g. majority voting over three frames, two sent via on link and the third via another link.



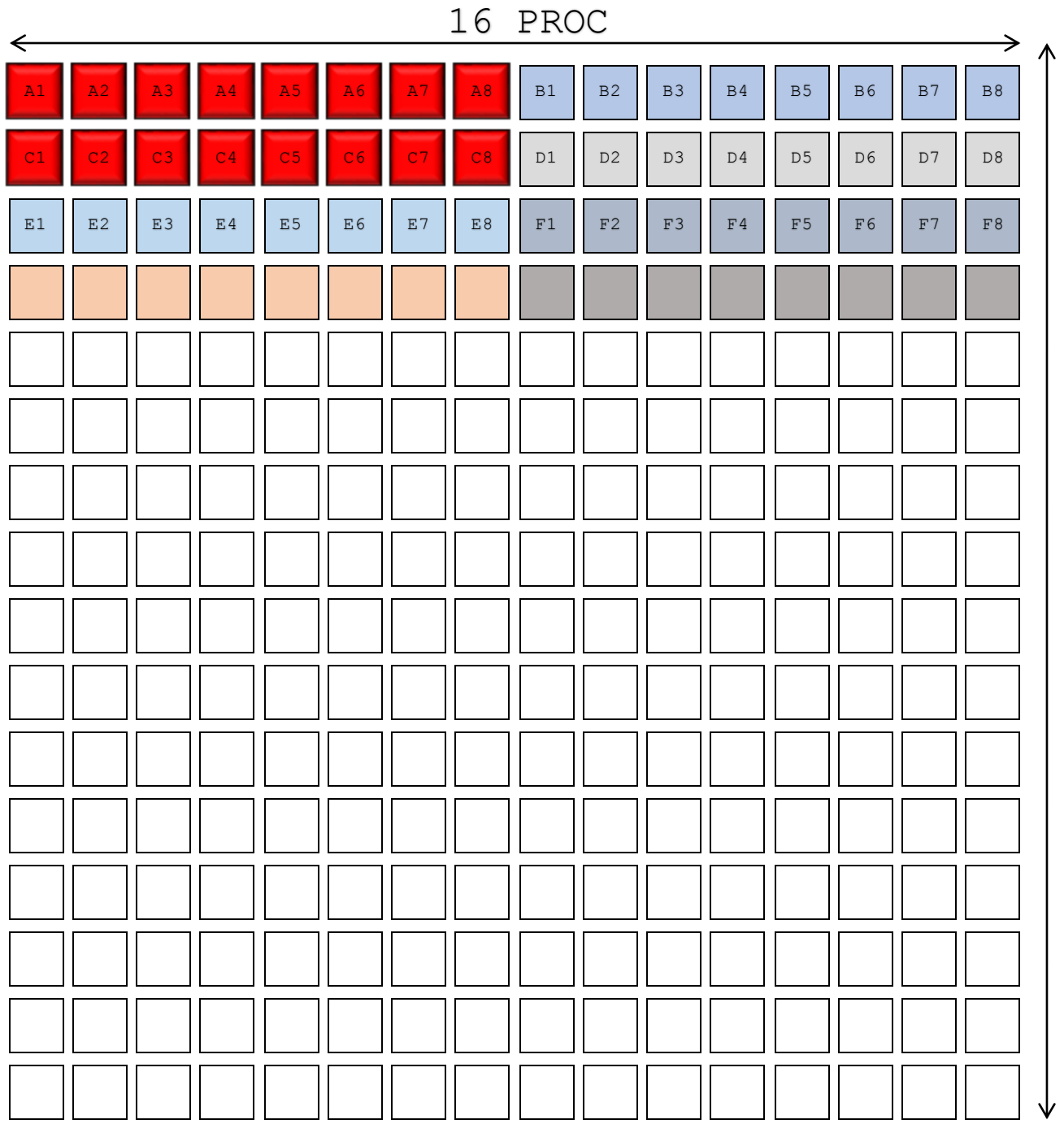

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```



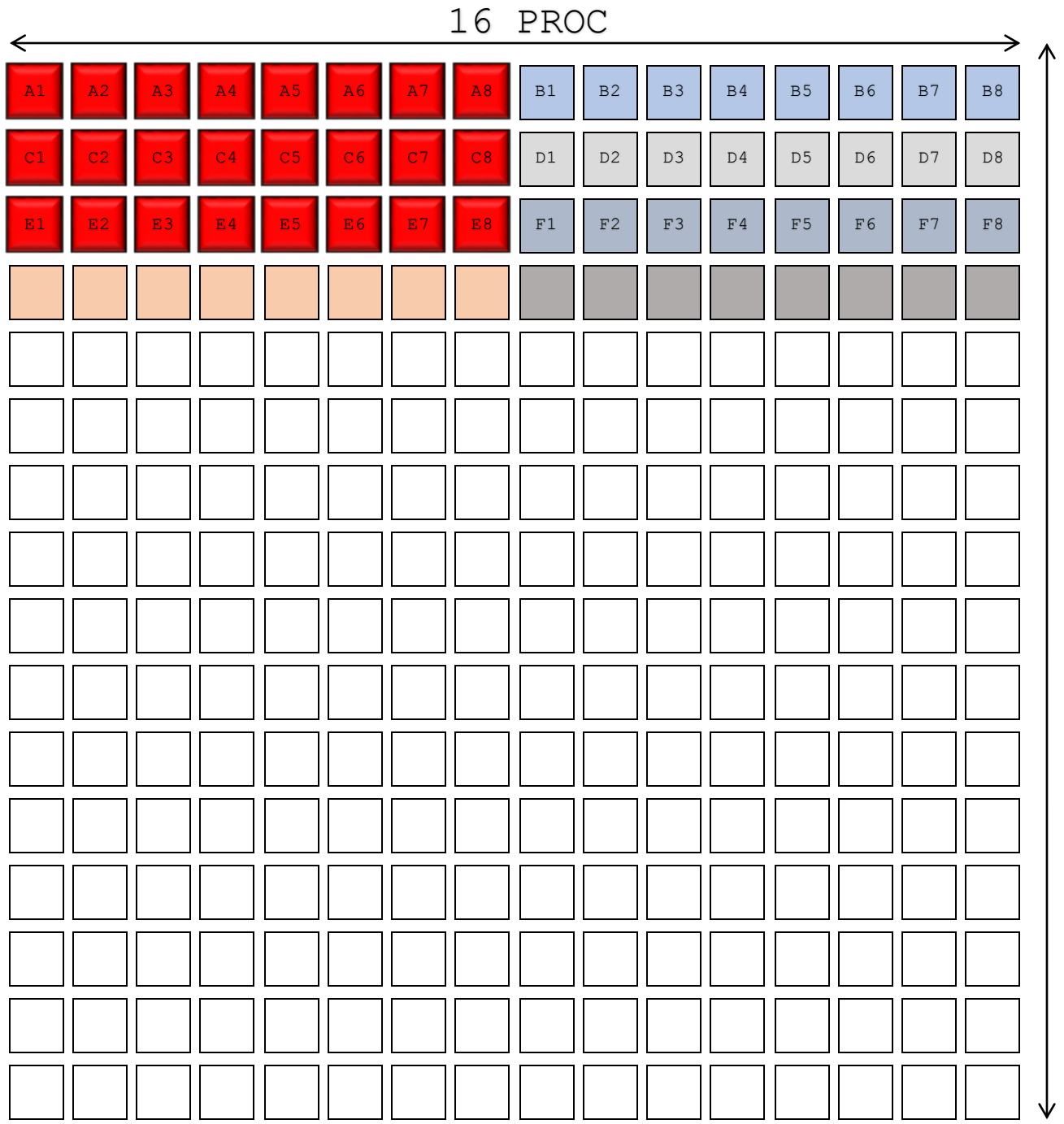

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```



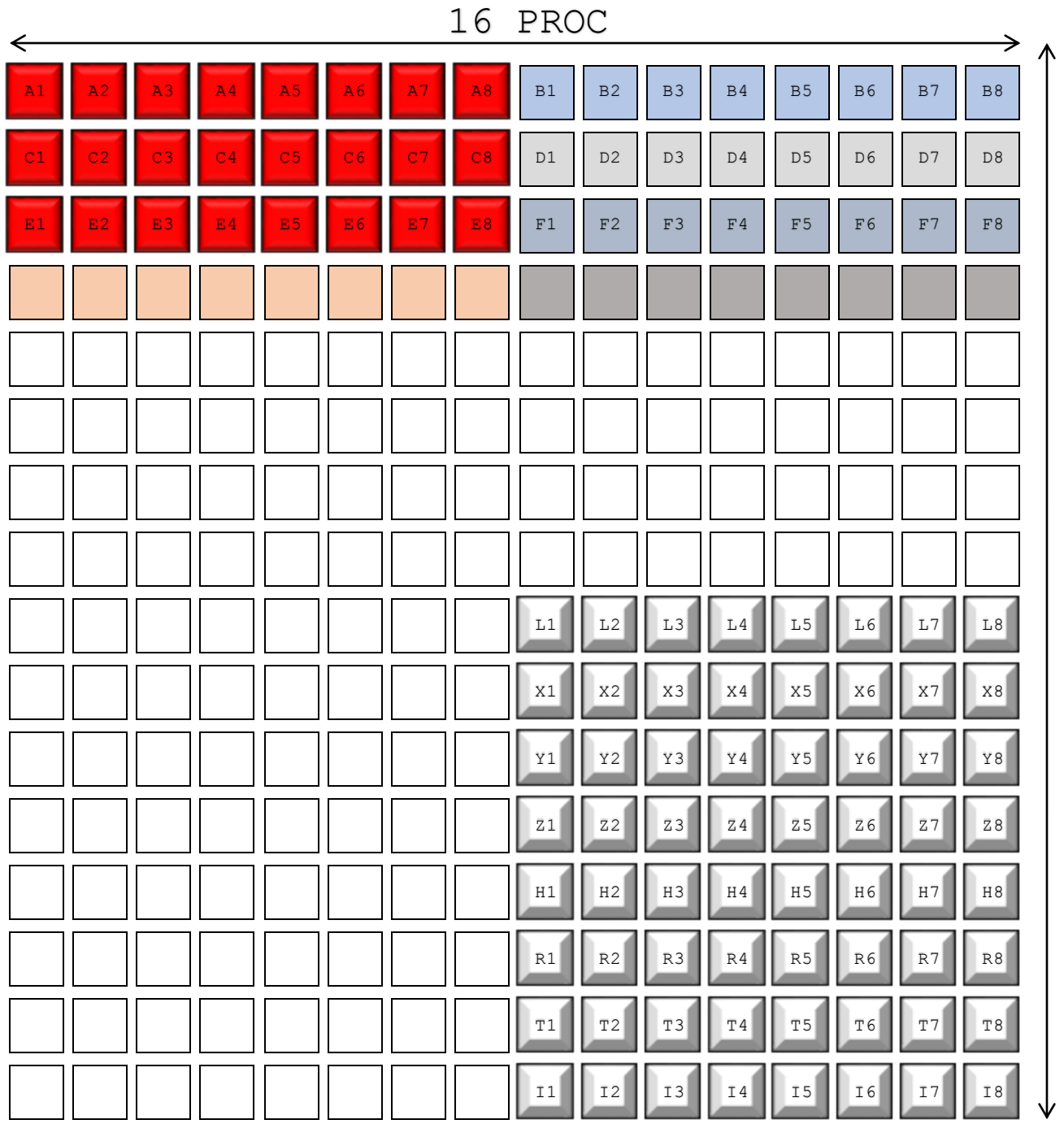

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```



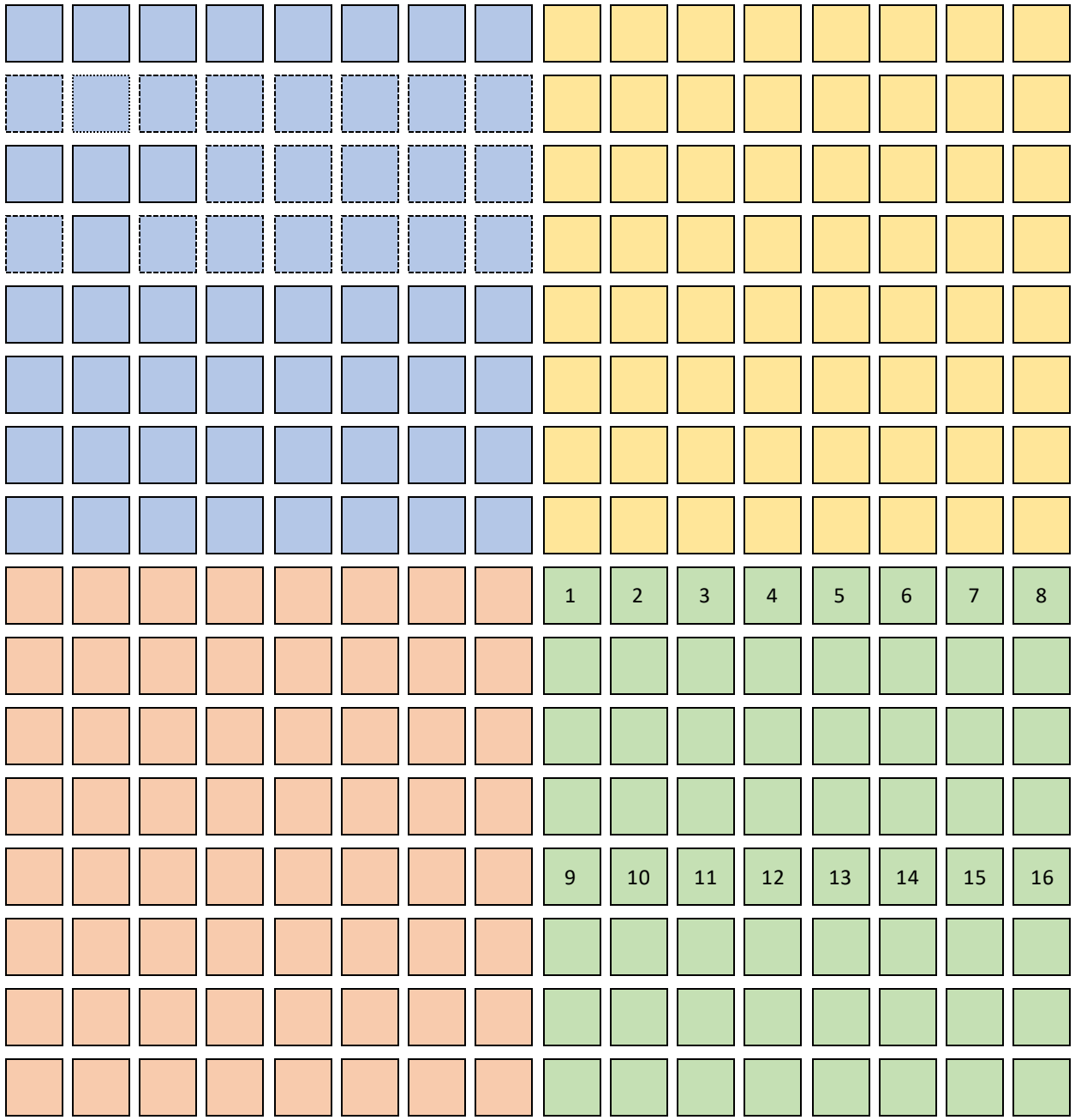

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```

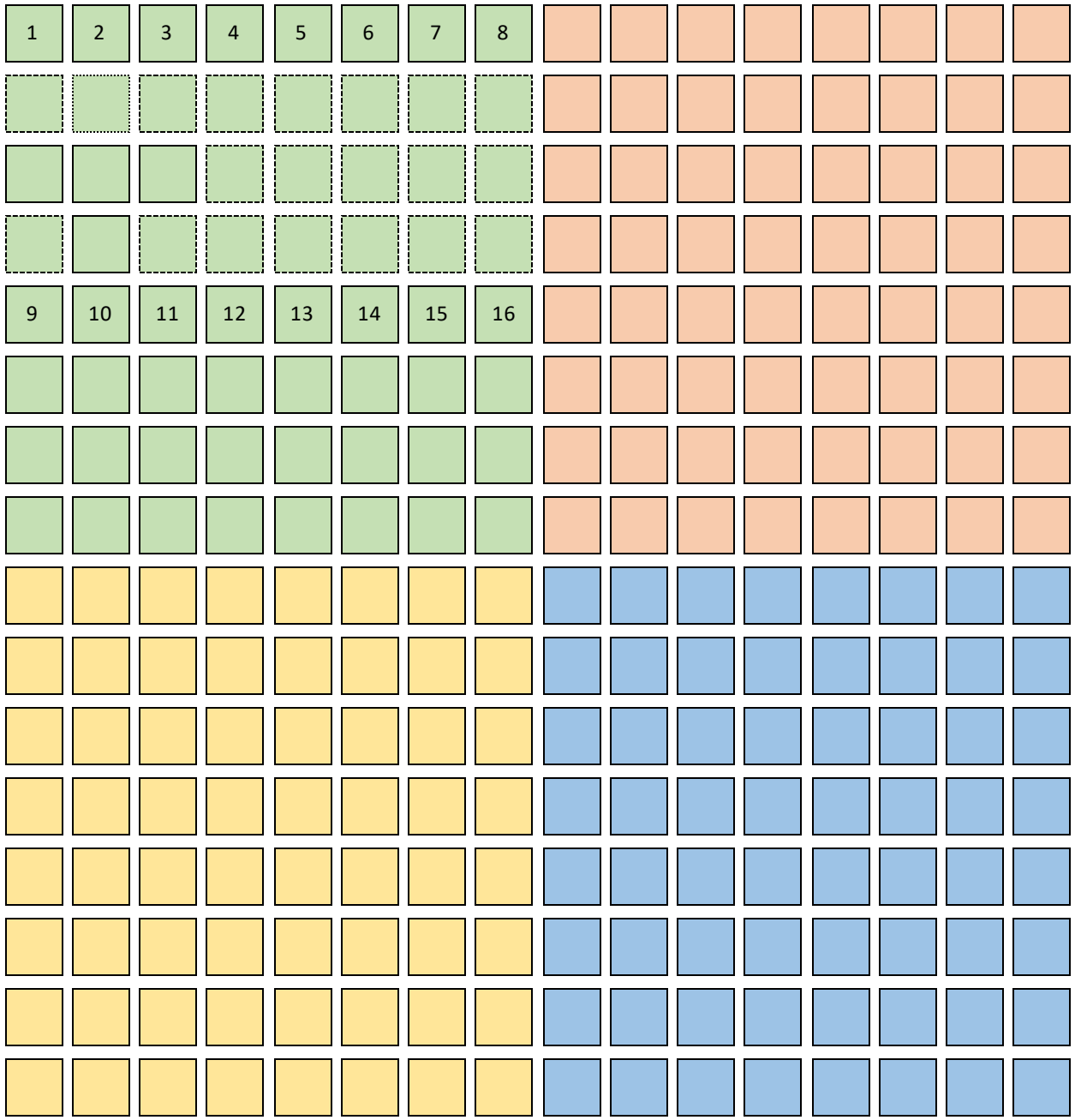


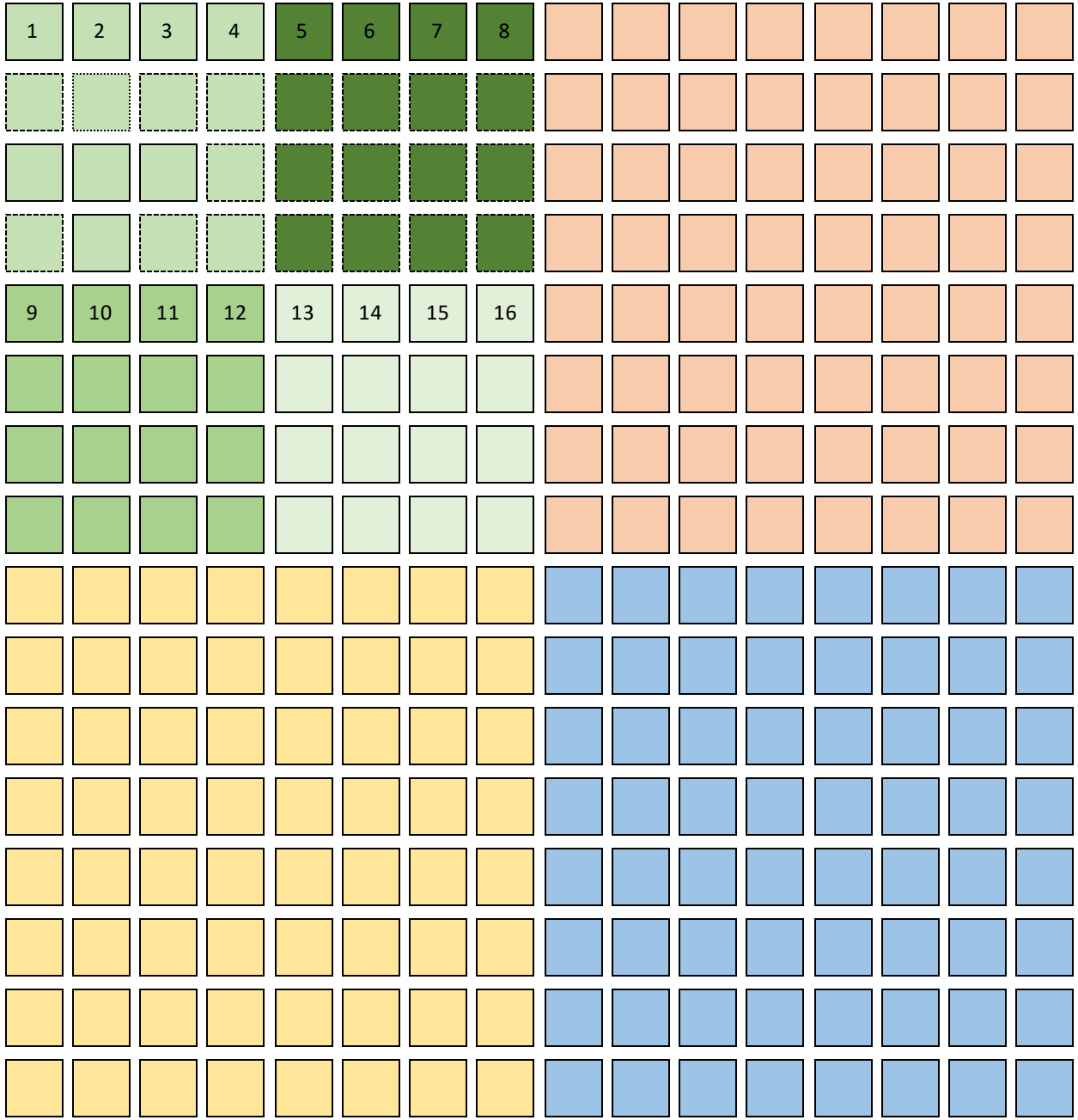

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```

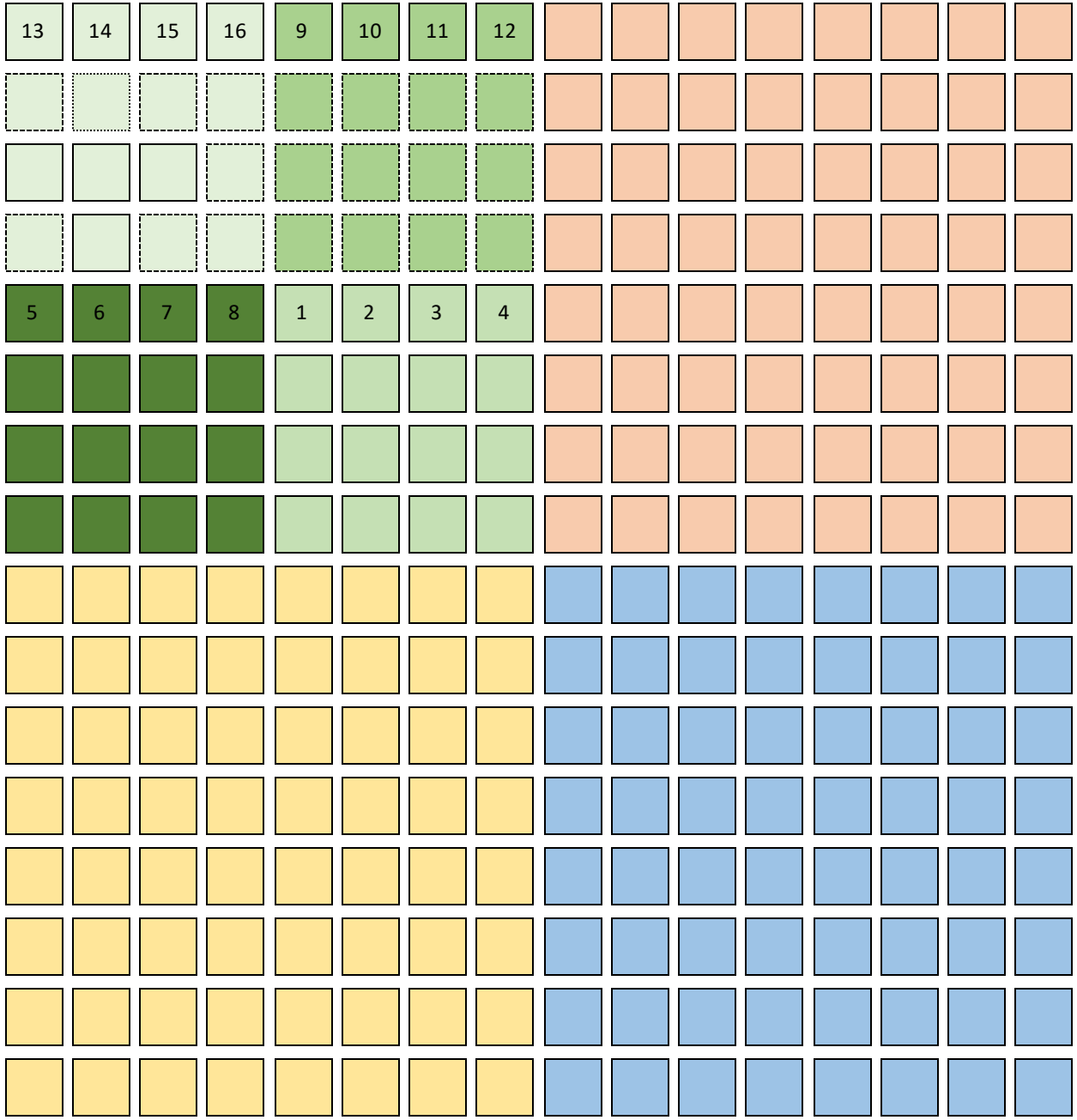


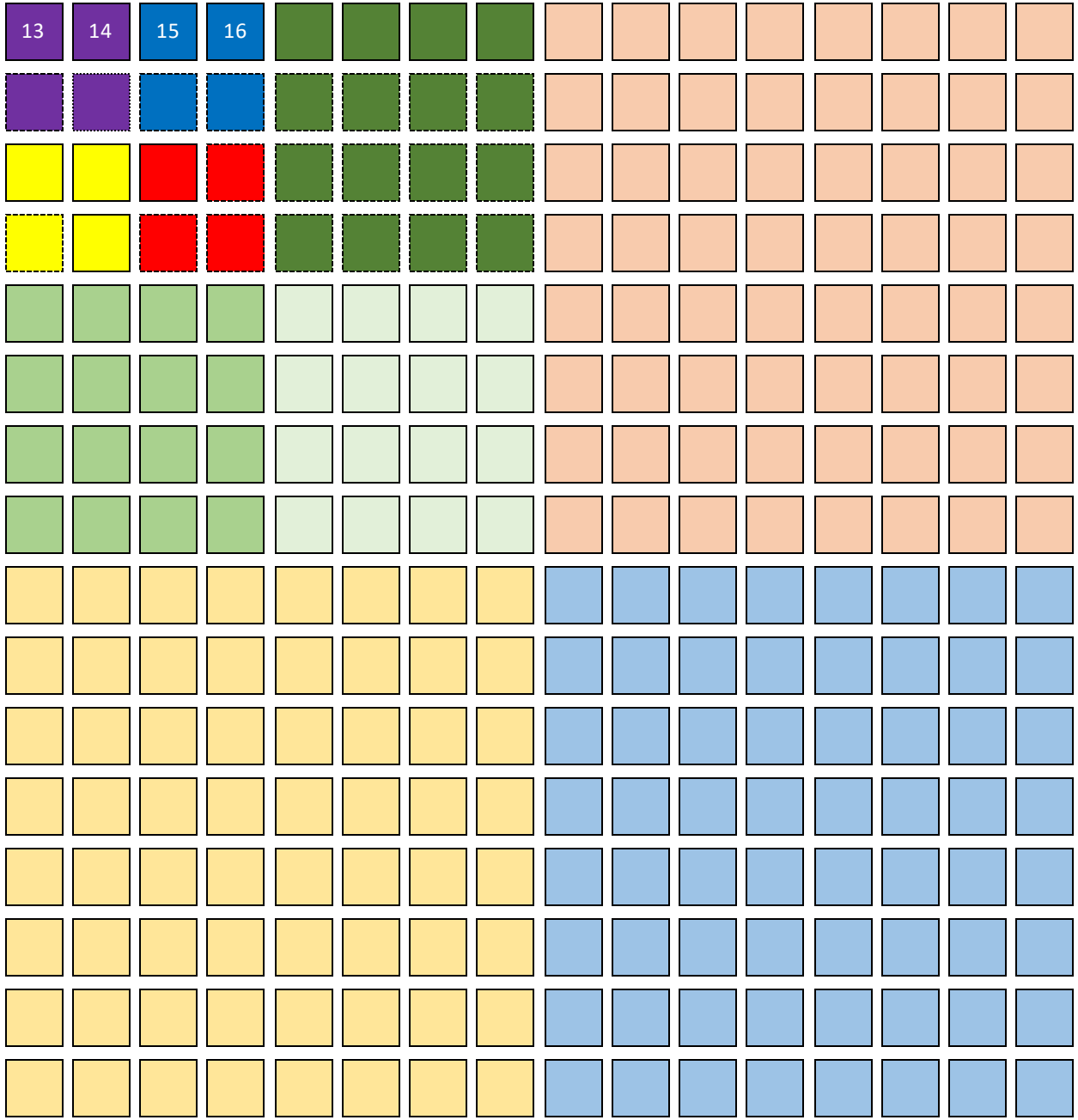

```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```

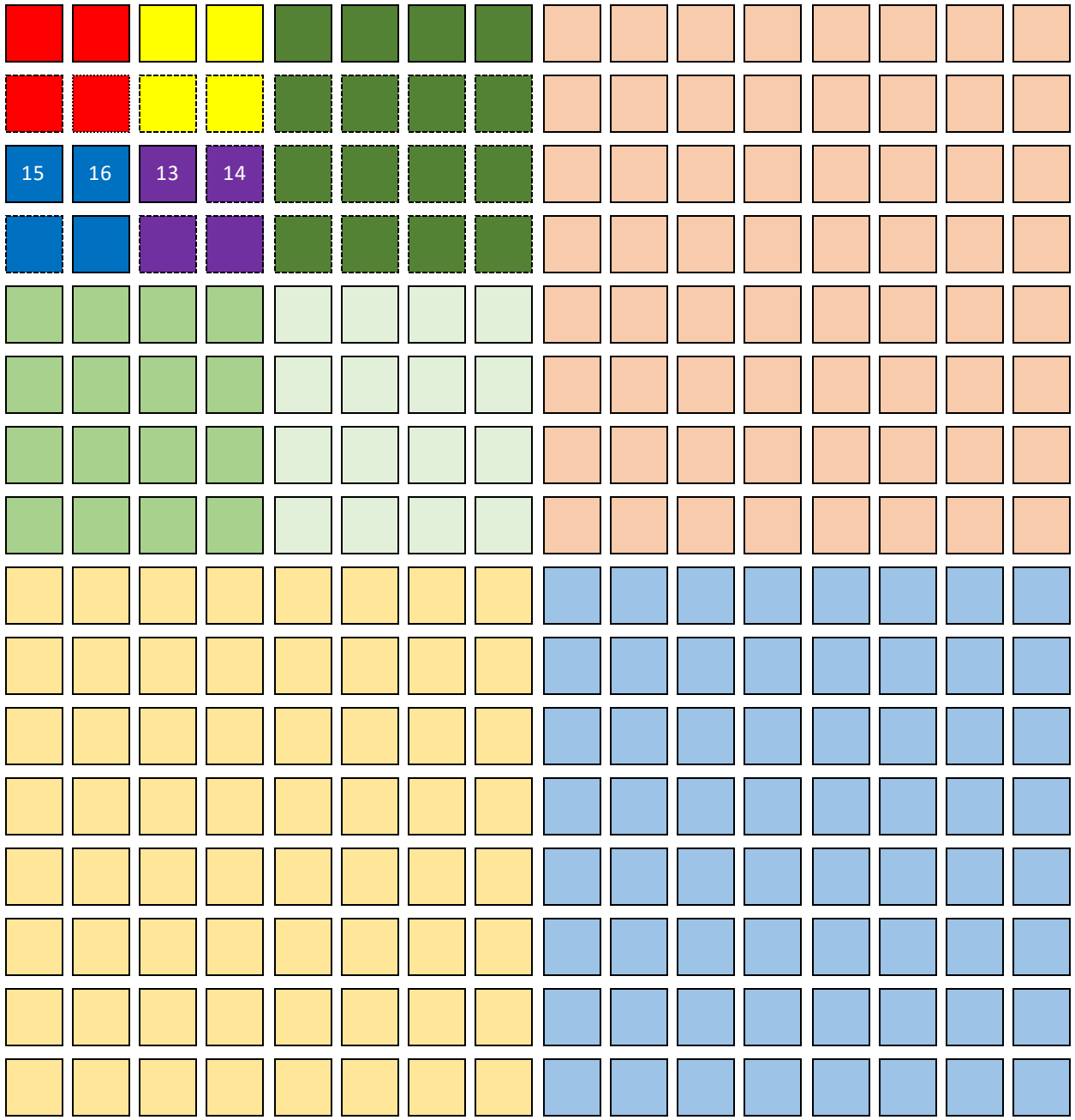

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16

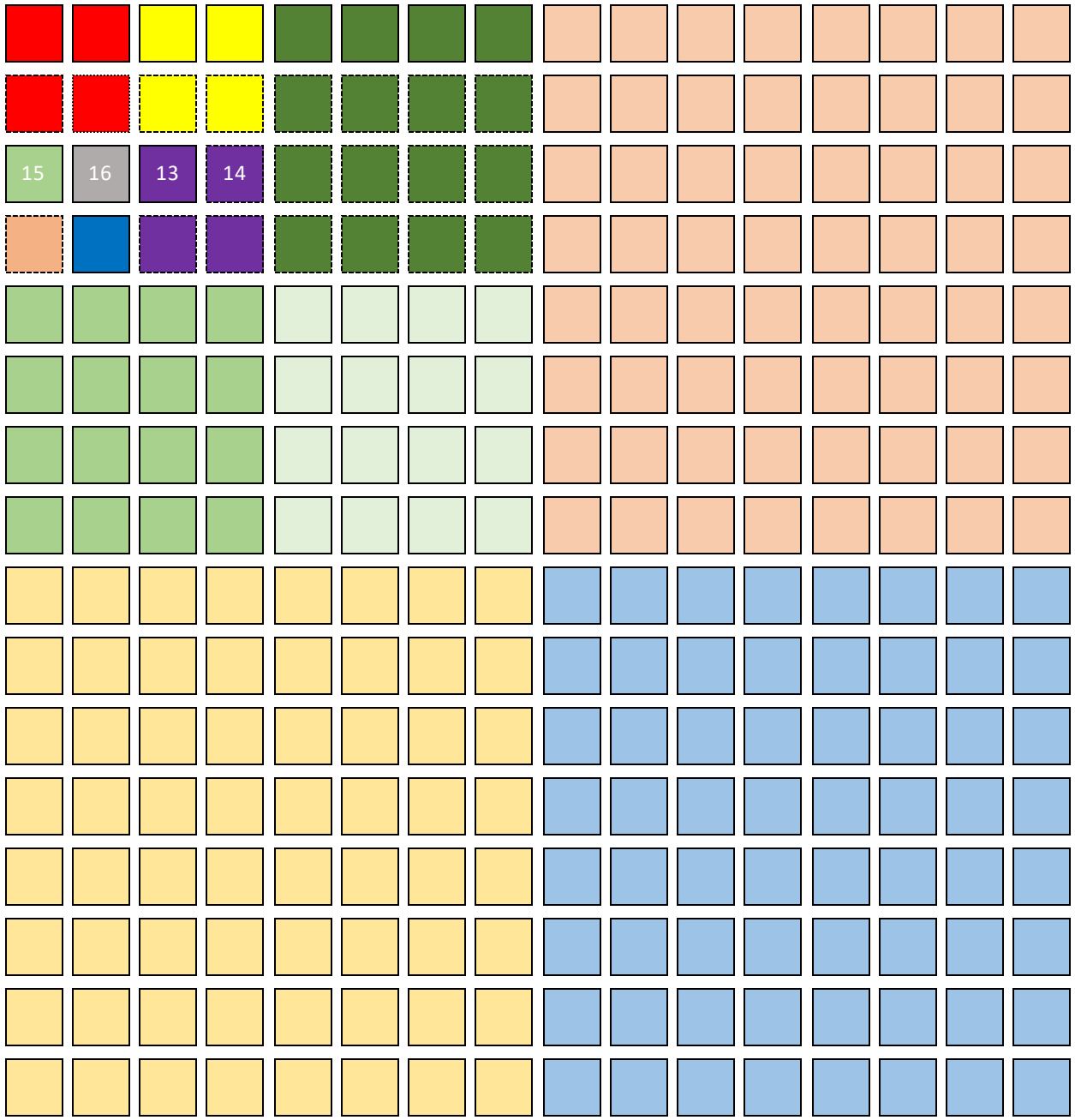

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

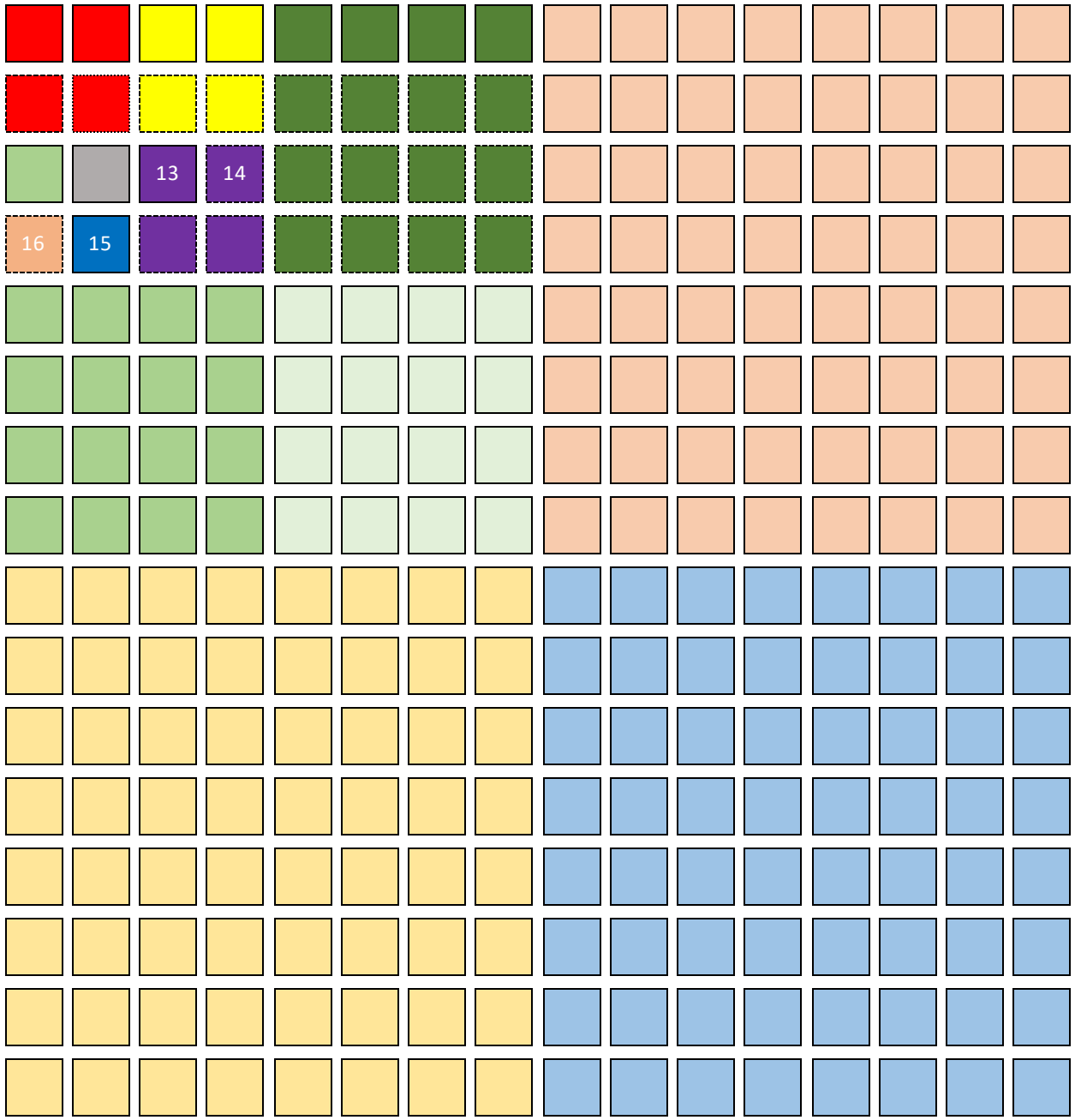

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

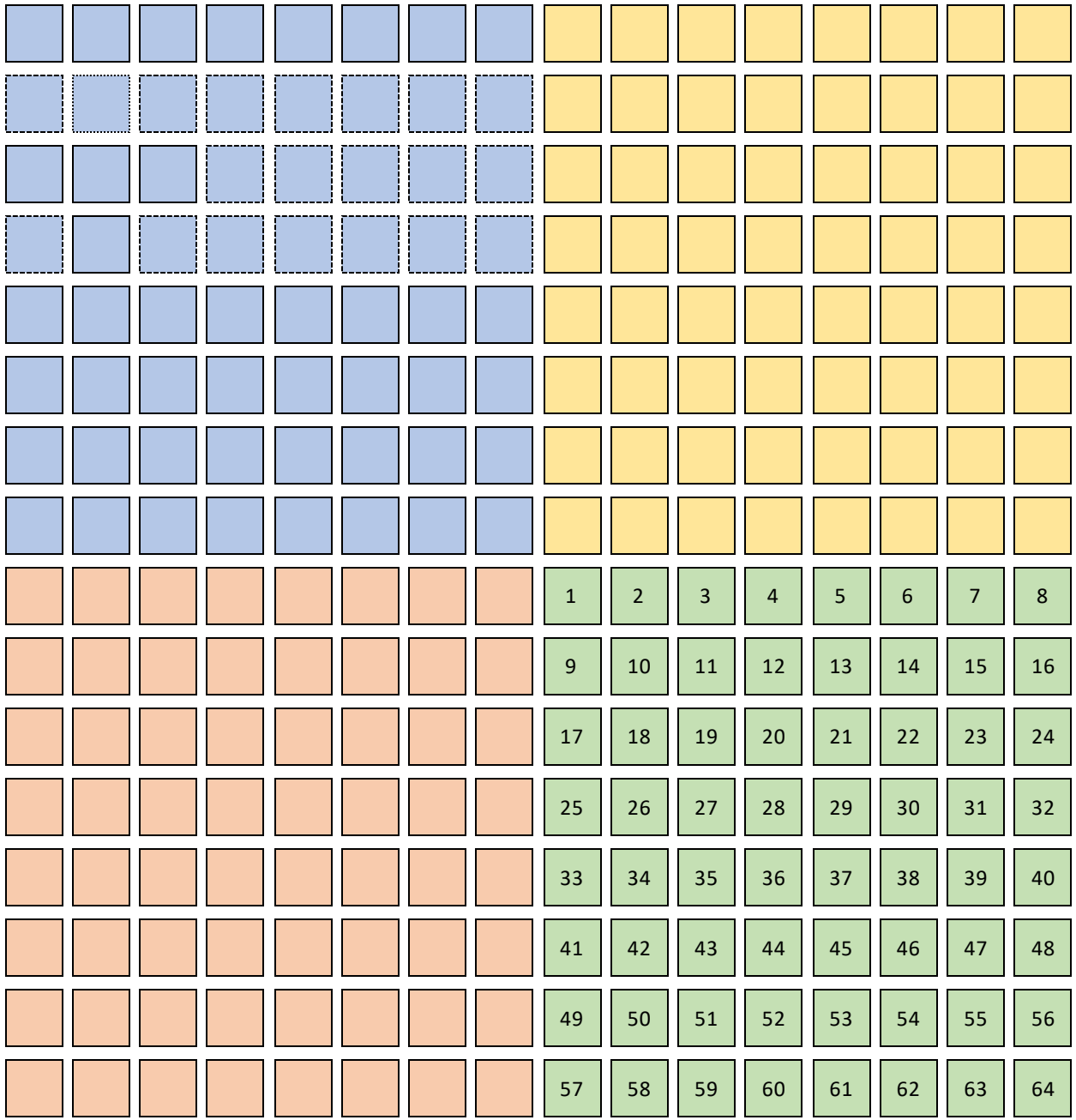

13
14
15
16
9
10
11
12
5
6
7
8
1
2
3
4

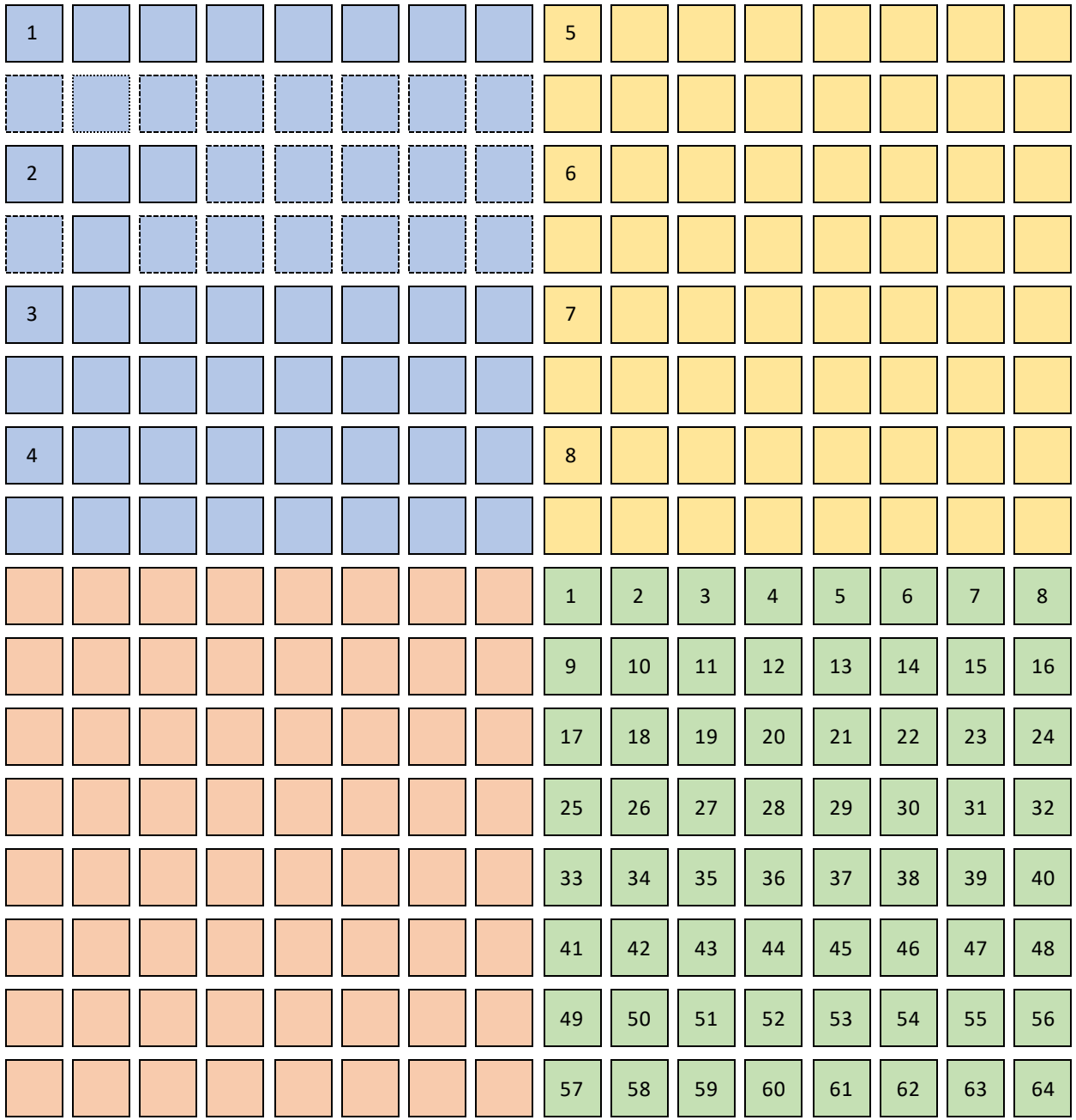

13
14
15
16

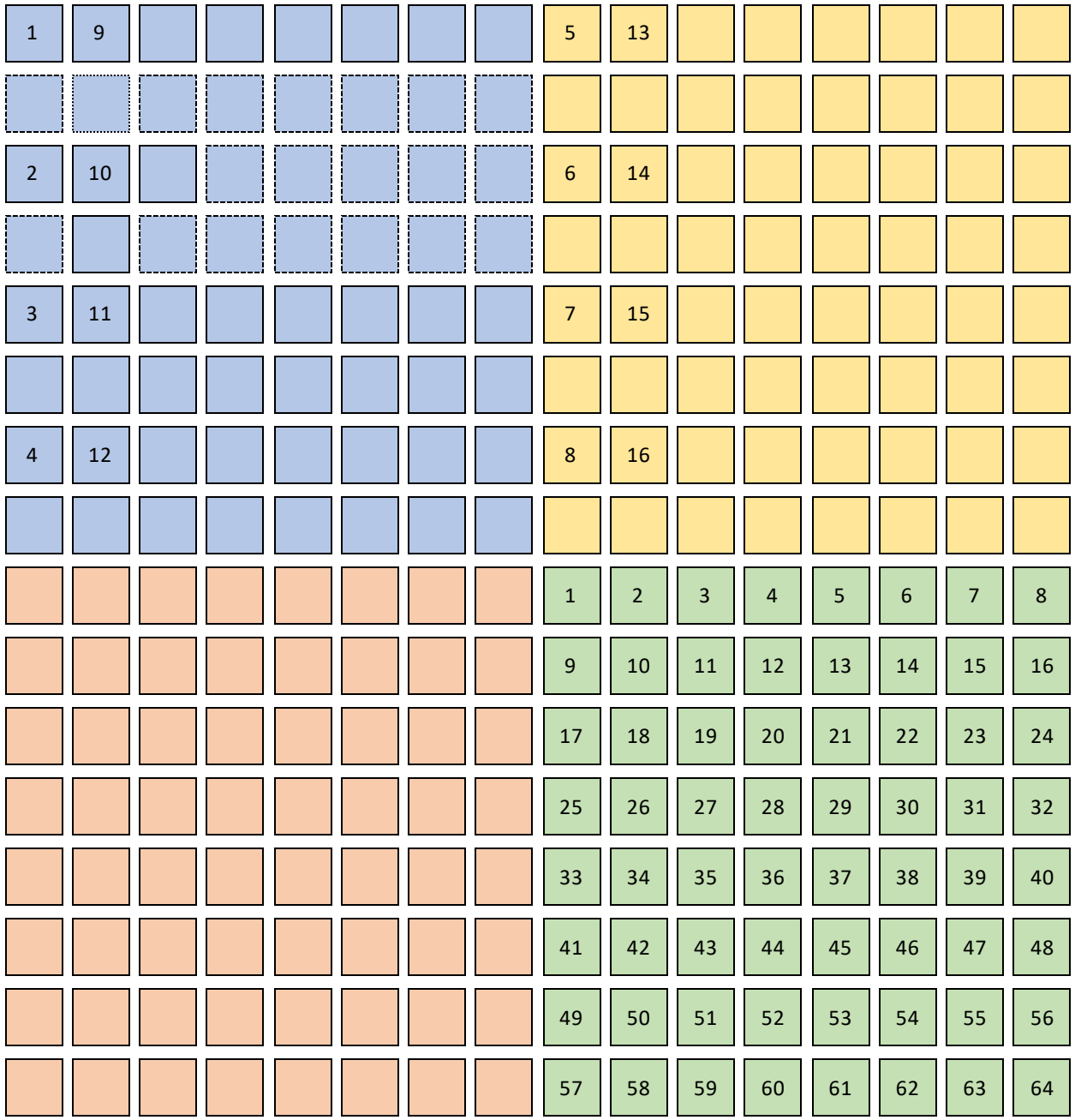

15
16
13
14

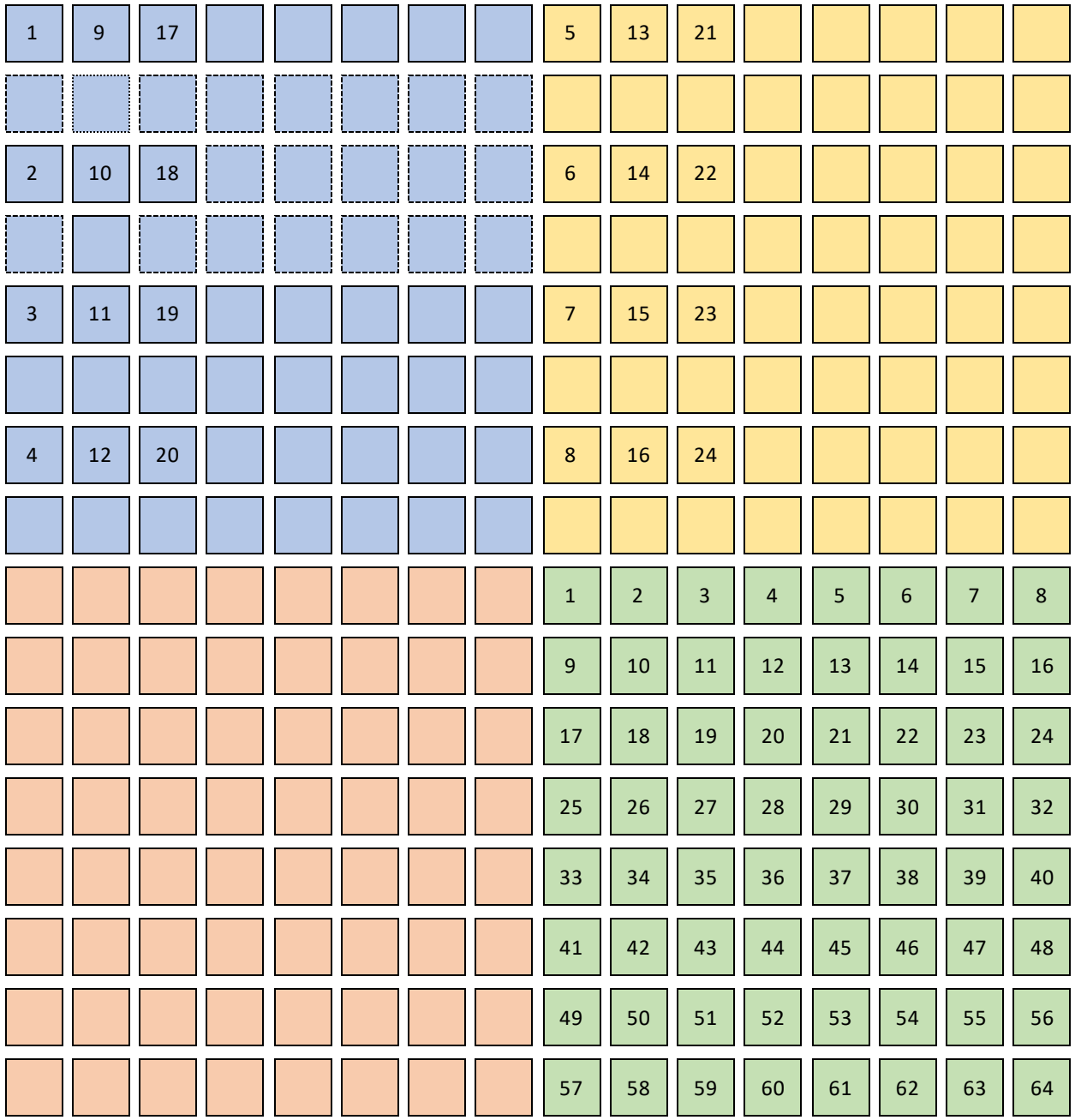

15
16
13
14

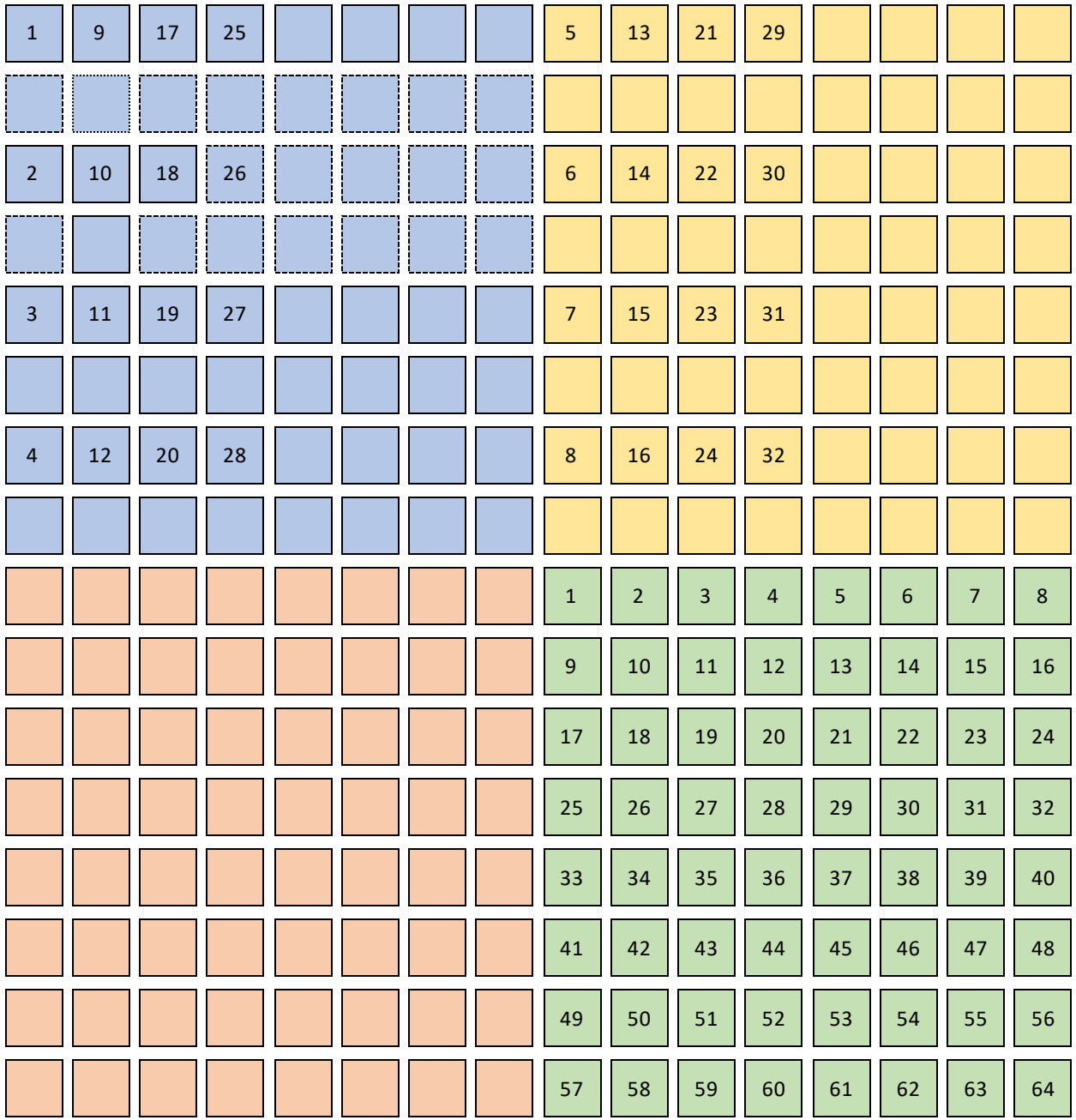

13
14
16
15

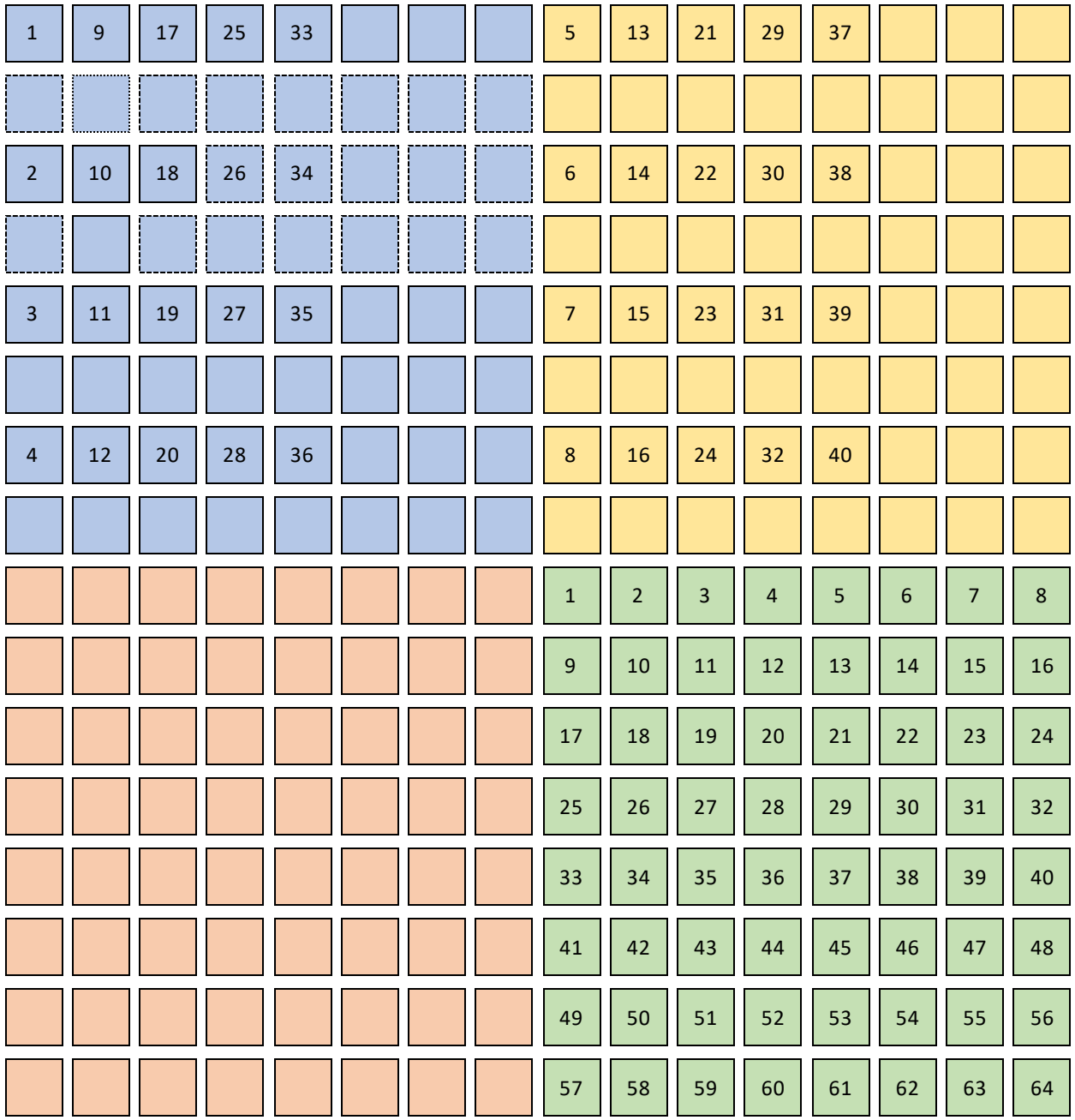

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

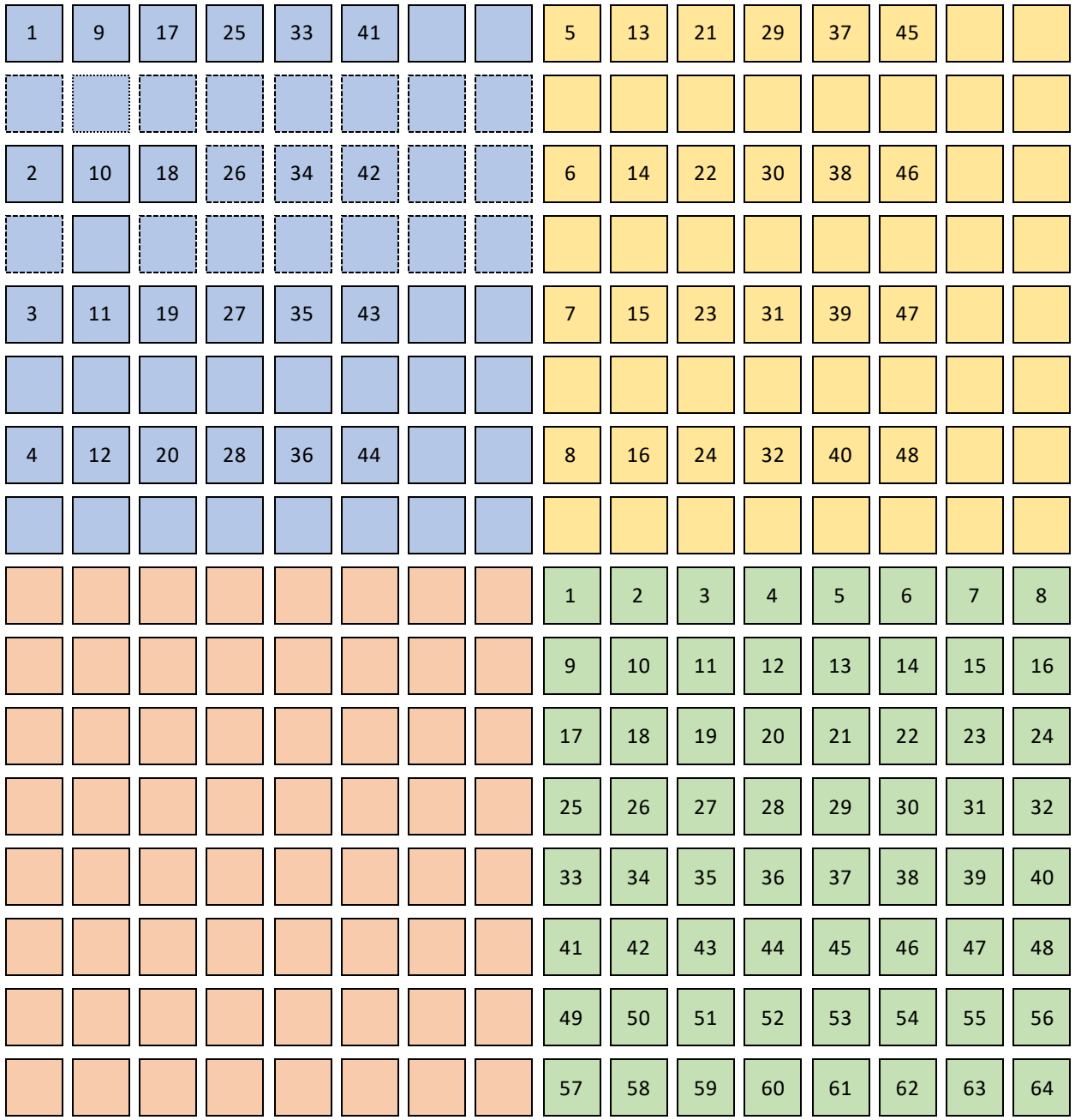

1
5
2
6
3
7
4
8
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

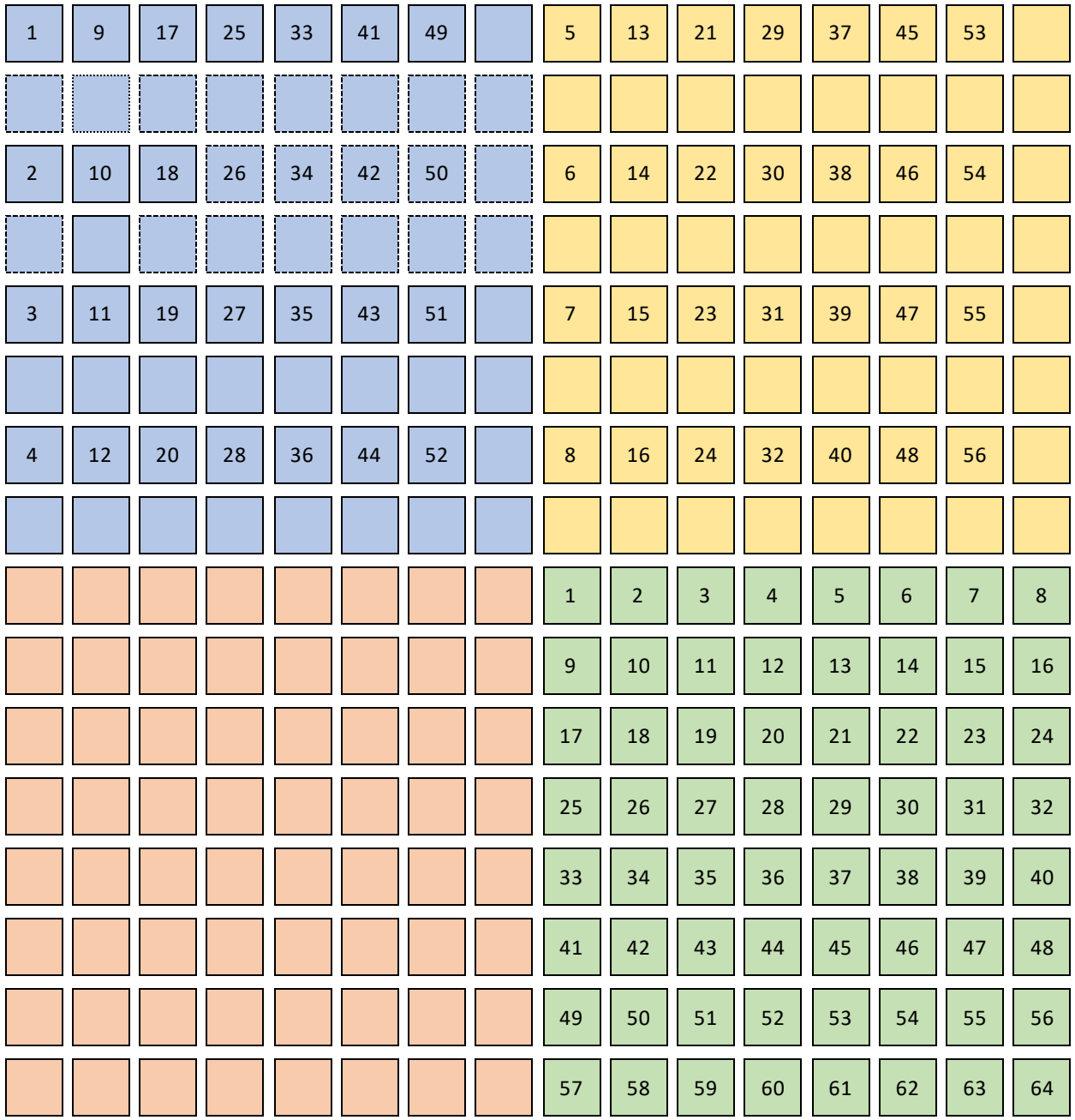

1 9 5 13
2 10 6 14
3 11 7 15
4 12 8 16
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

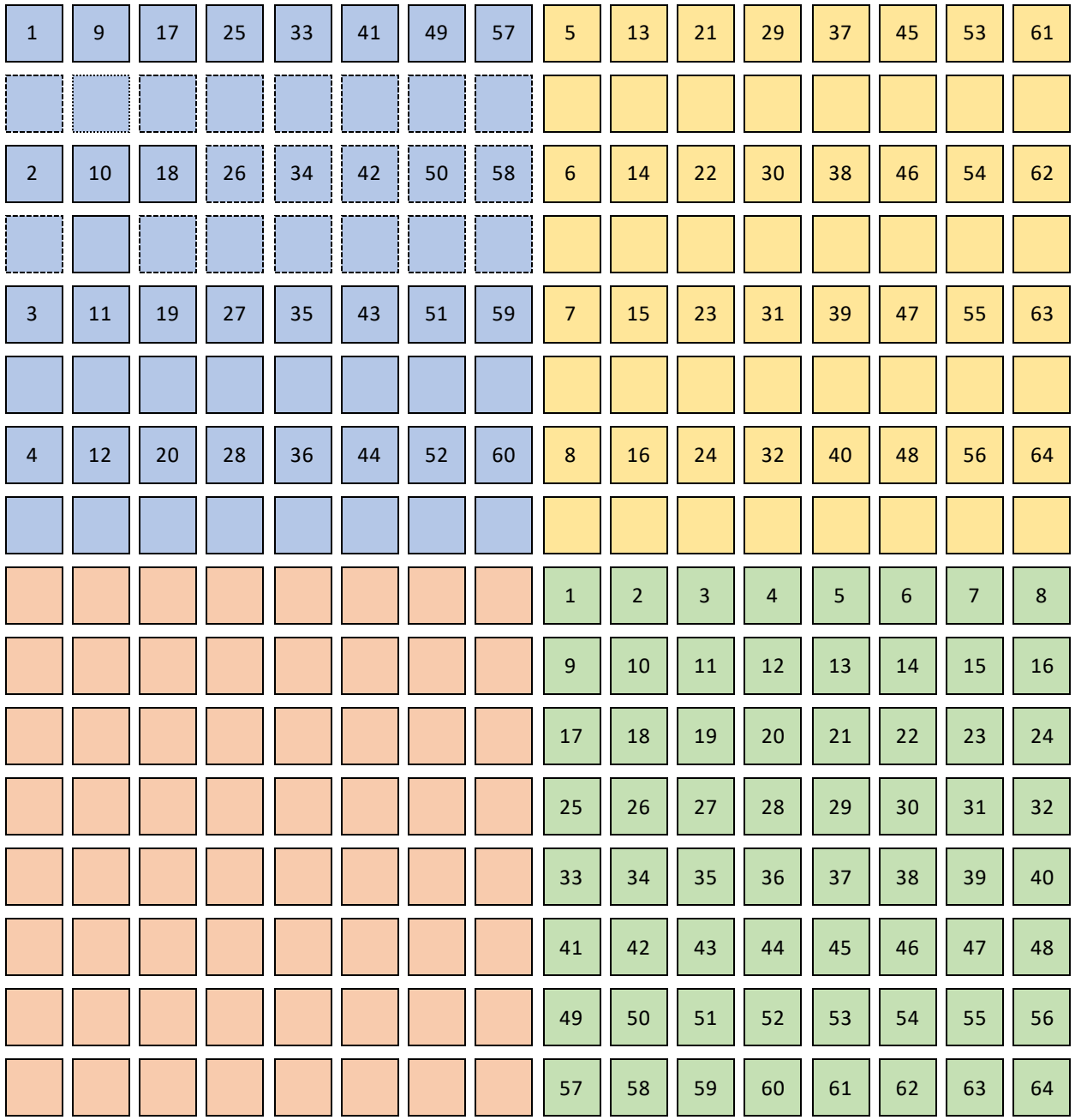

1 9 17 5 13 21
2 10 18 6 14 22
3 11 19 7 15 23
4 12 20 8 16 24
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

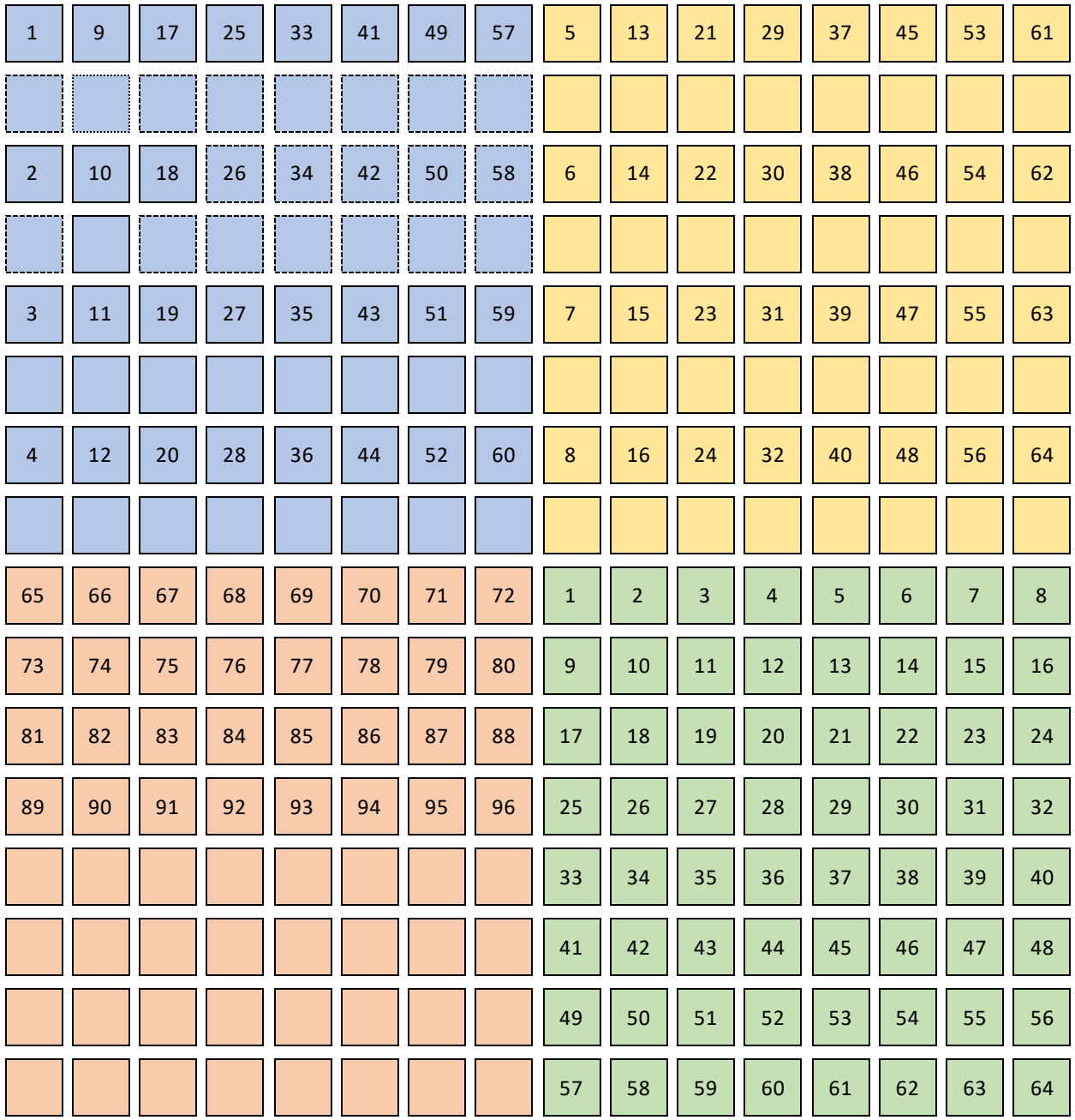

1 9 17 25 5 13 21 29
2 10 18 26 6 14 22 30
3 11 19 27 7 15 23 31
4 12 20 28 8 16 24 32
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

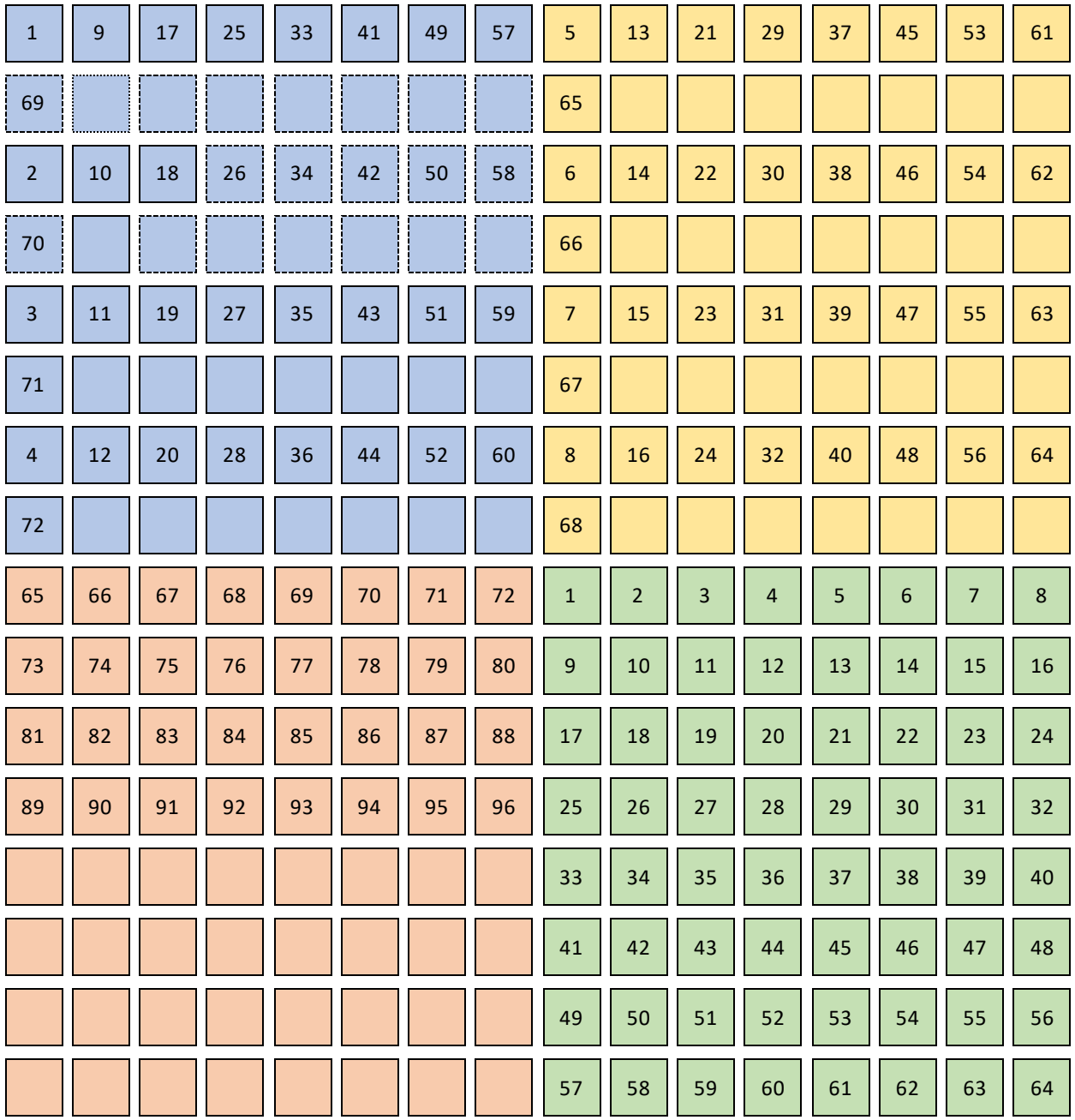

1 9 17 25 33 5 13 21 29 37
2 10 18 26 34 6 14 22 30 38
3 11 19 27 35 7 15 23 31 39
4 12 20 28 36 8 16 24 32 40
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

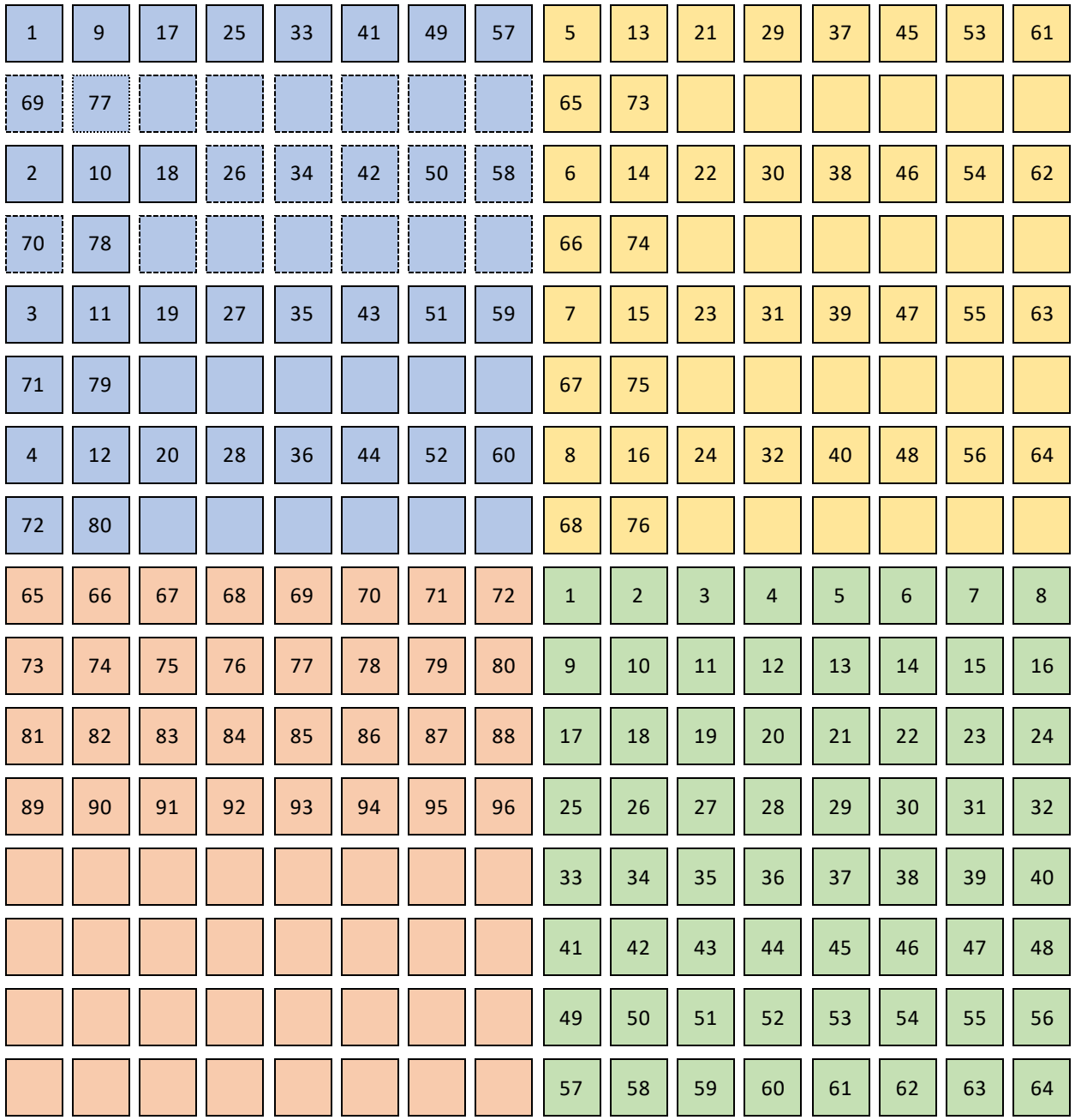

1 9 17 25 33 41 5 13 21 29 37 45
2 10 18 26 34 42 6 14 22 30 38 46
3 11 19 27 35 43 7 15 23 31 39 47
4 12 20 28 36 44 8 16 24 32 40 48
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

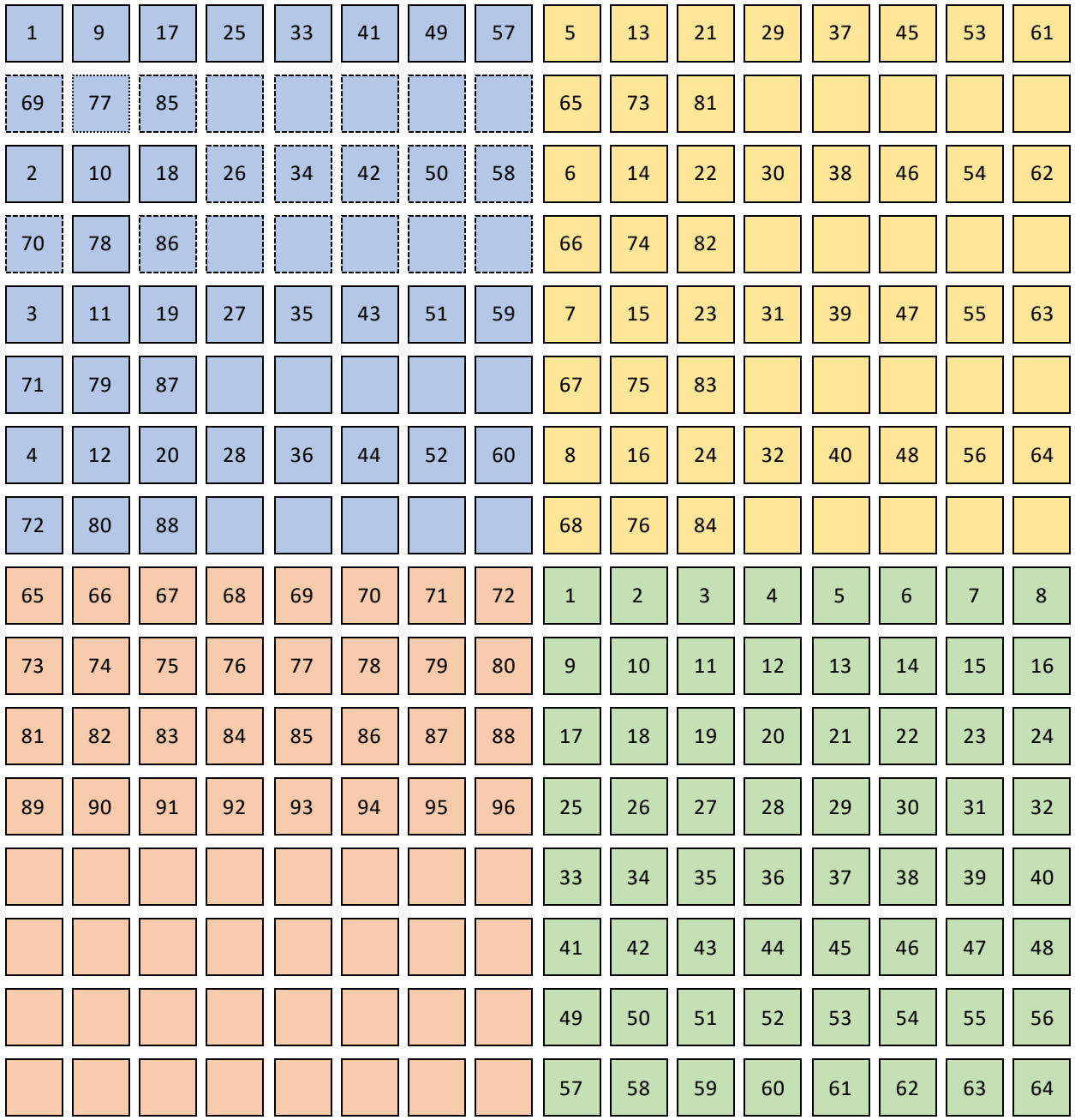

1 9 17 25 33 41 49 5 13 21 29 37 45 53
2 10 18 26 34 42 50 6 14 22 30 38 46 54
3 11 19 27 35 43 51 7 15 23 31 39 47 55
4 12 20 28 36 44 52 8 16 24 32 40 48 56
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

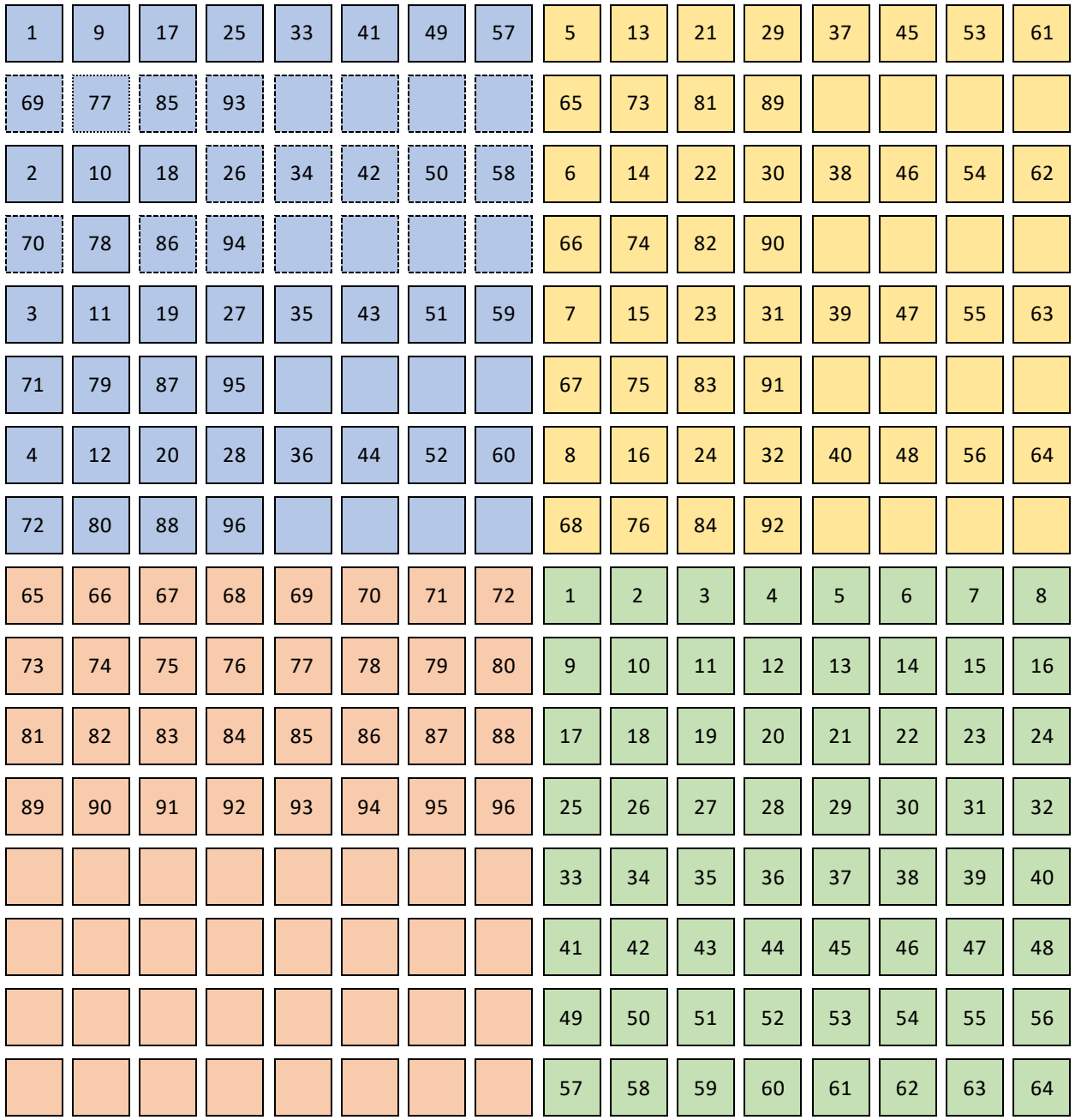

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64

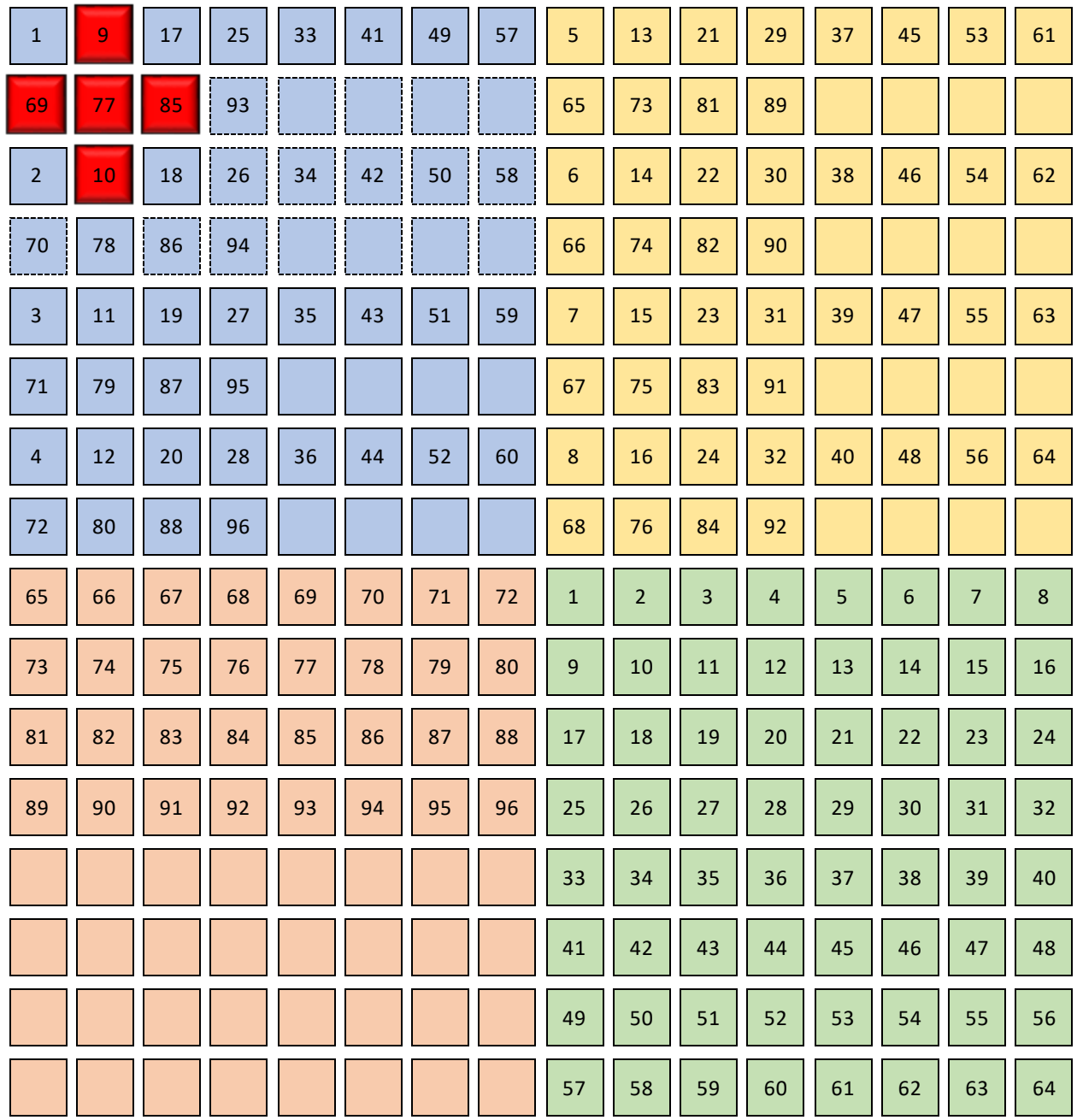

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
65 66 67 68 69 70 71 72 1 2 3 4 5 6 7 8
73 74 75 76 77 78 79 80 9 10 11 12 13 14 15 16
81 82 83 84 85 86 87 88 17 18 19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

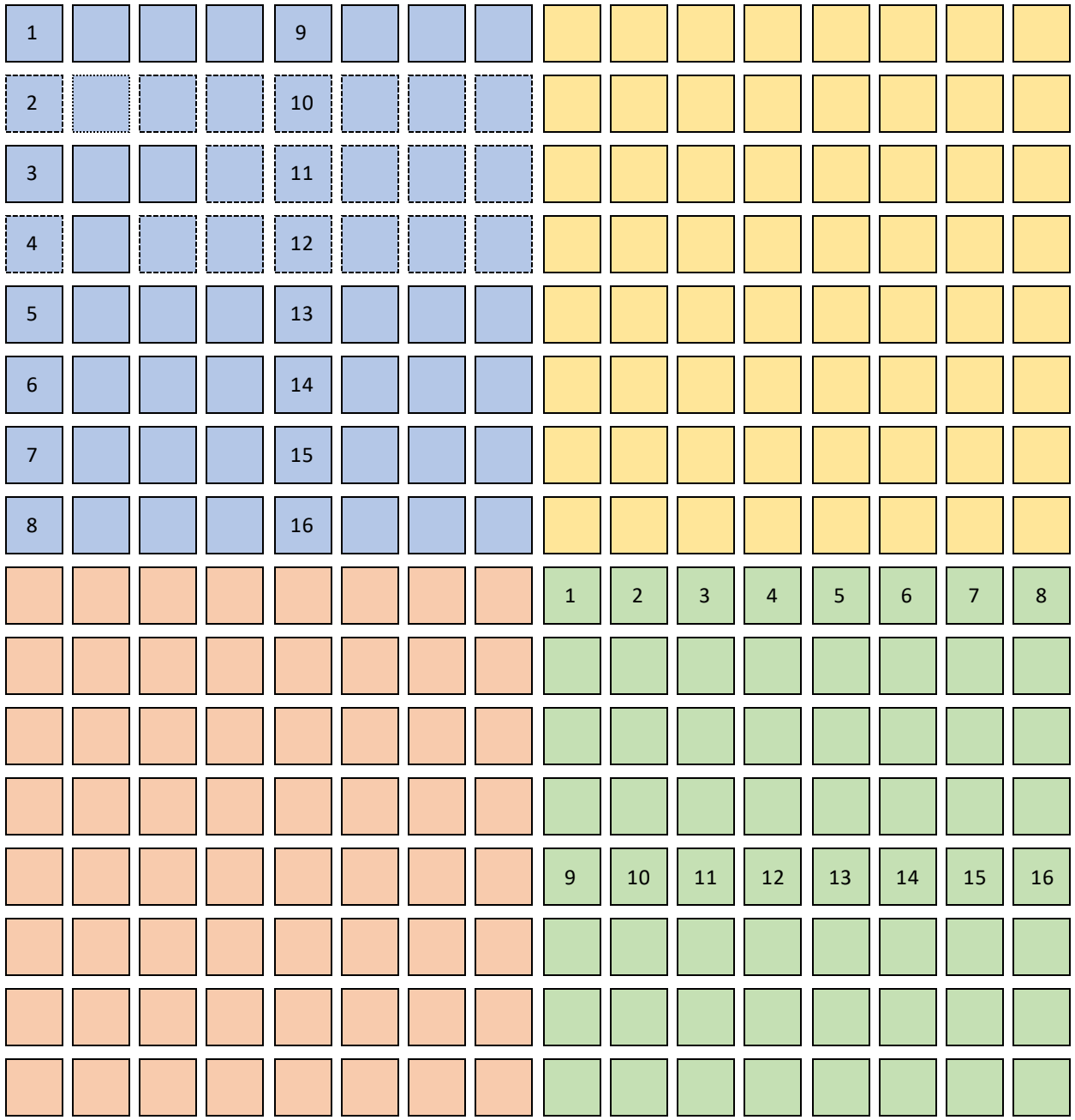

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
69 65
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
70 66
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
71 67
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
72 68
65 66 67 68 69 70 71 72 1 2 3 4 5 6 7 8
73 74 75 76 77 78 79 80 9 10 11 12 13 14 15 16
81 82 83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

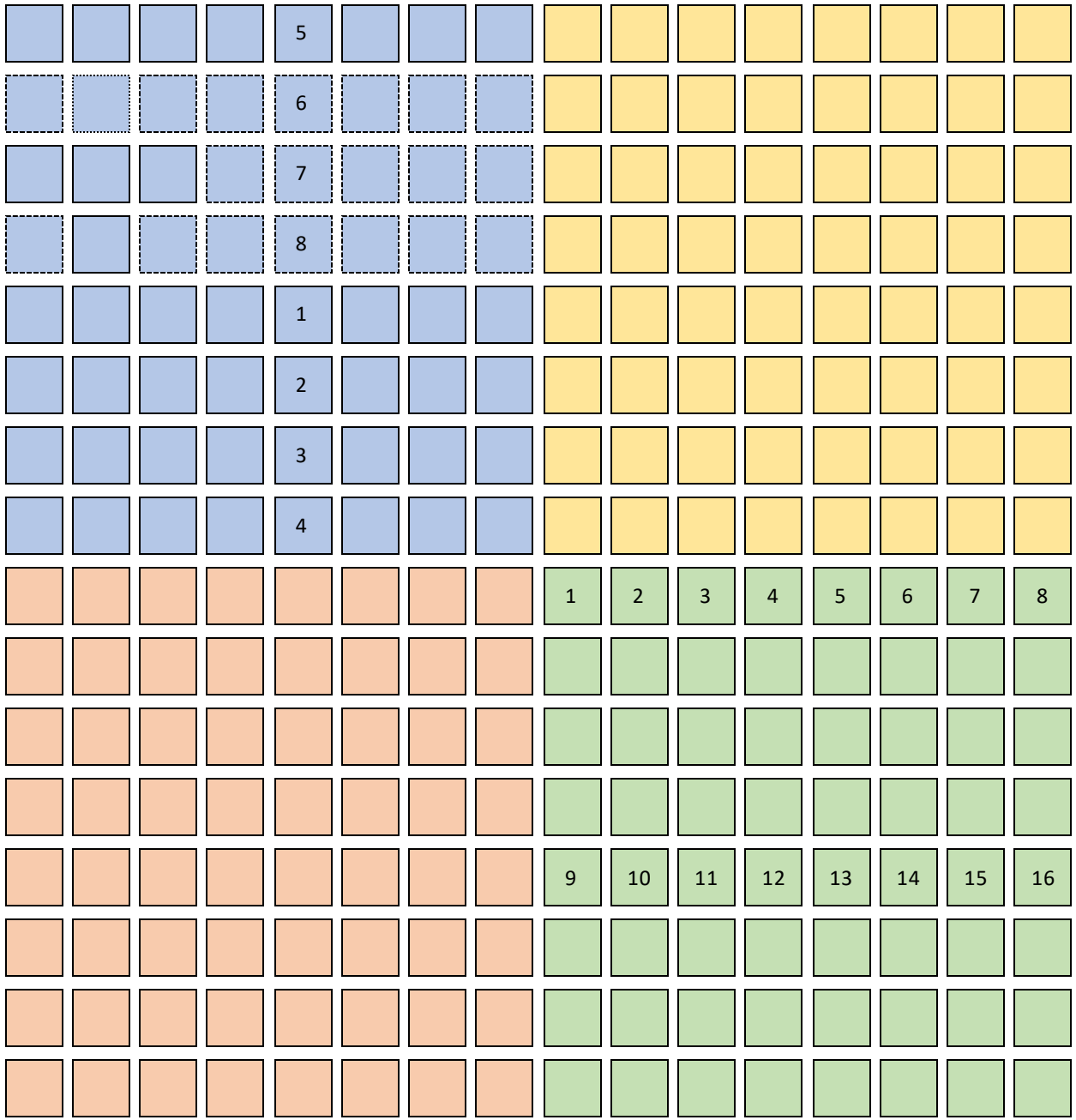

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
69 77 65 73
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
70 78 66 74
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
71 79 67 75
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
72 80 68 76
65 66 67 68 69 70 71 72 1 2 3 4 5 6 7 8
73 74 75 76 77 78 79 80 9 10 11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

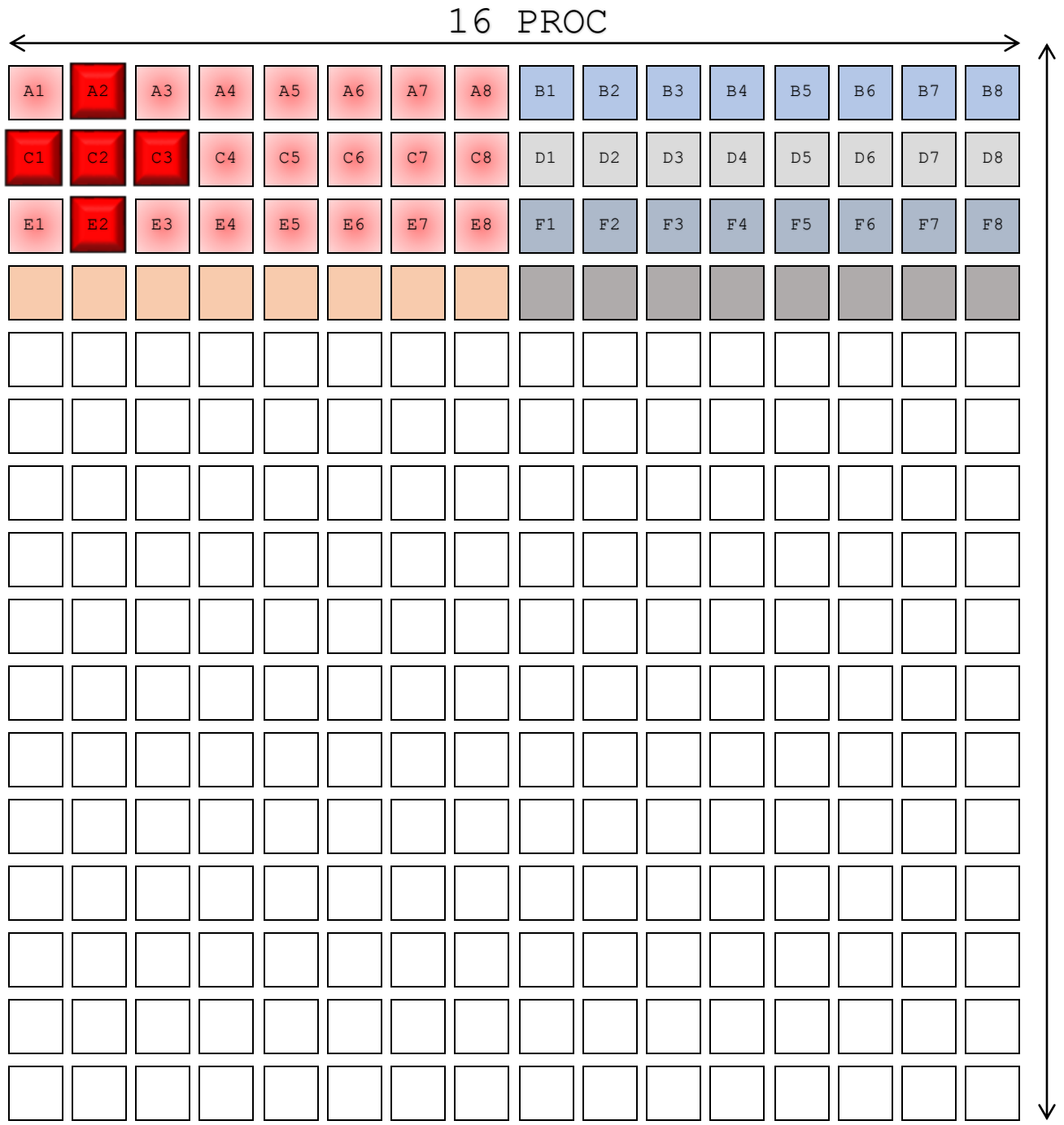

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
69 77 85 65 73 81
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
70 78 86 66 74 82
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
71 79 87 67 75 83
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
72 80 88 68 76 84
65 66 67 68 69 70 71 72 1 2 3 4 5 6 7 8
73 74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


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

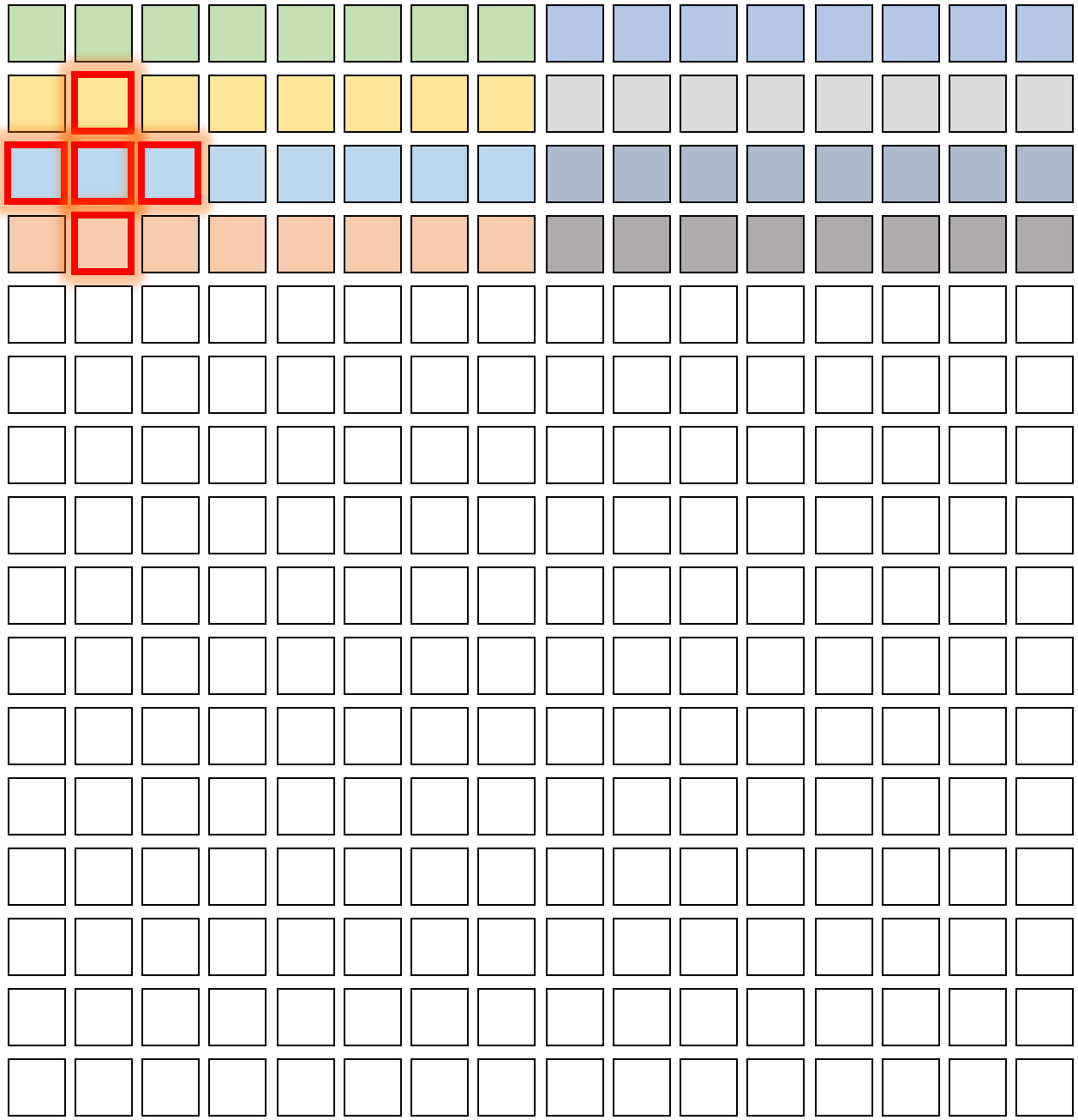

1 9 17 25 33 41 49 57 5 13 21 29 37 45 53 61
69 77 85 93 65 73 81 89
2 10 18 26 34 42 50 58 6 14 22 30 38 46 54 62
70 78 86 94 66 74 82 90
3 11 19 27 35 43 51 59 7 15 23 31 39 47 55 63
71 79 87 95 67 75 83 91
4 12 20 28 36 44 52 60 8 16 24 32 40 48 56 64
72 80 88 96 68 76 84 92
65 66 67 68 69 70 71 72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

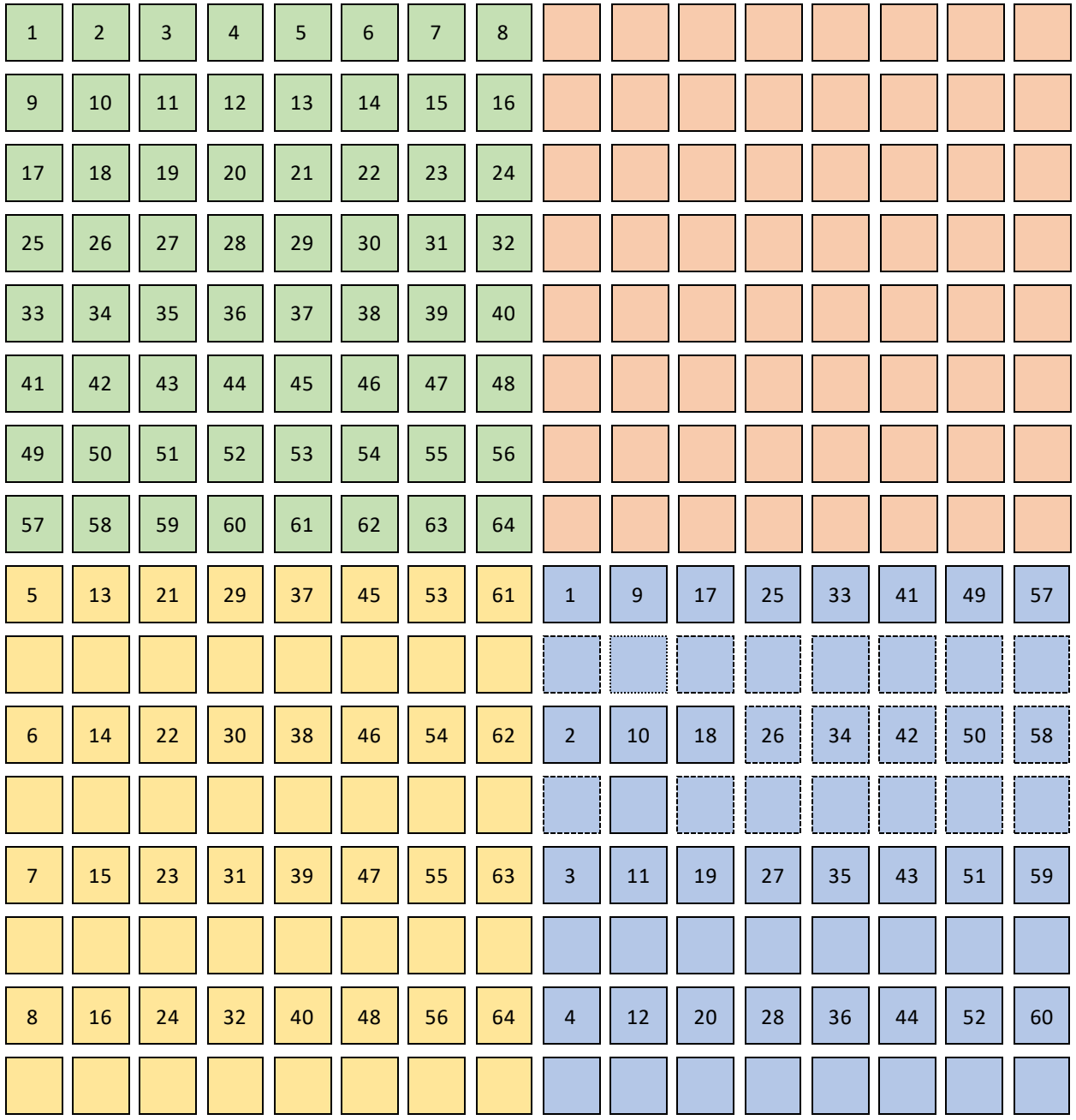

1
9
2
10
3
11
4
12
5
13
6
14
7
15
8
16
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16


5
6
7
8
1
2
3
4
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16




```
#!/bin/bash
#SBATCH --nodes=32 --ntasks-per-node=8
mpiexec ./heat_2d 8192 8192
```




1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
5 13 21 29 37 45 53 61 1 9 17 25 33 41 49 57
6 14 22 30 38 46 54 62 2 10 18 26 34 42 50 58
7 15 23 31 39 47 55 63 3 11 19 27 35 43 51 59
8 16 24 32 40 48 56 64 4 12 20 28 36 44 52 60