In [1]:

```
import random
import math
from faker import Faker
import numpy as np

'''
This is an example of a map of a ware house
    Data structure: 2D array
    Size: 6x6
    0 represents no shelf
    A capital letter represents the name of the shelf
'''



class WareHouse:
    def __init__(self, map, location=dict()):
        self.map = map
        self.location = location
        for i1, row in enumerate(map):
            for i2, shelf in enumerate(row):
                if shelf.isalpha():
                    self.location[shelf] = [i1, i2]


warehouse1 = WareHouse(map=np.array([[0, 0, 'D', 0, 0, 0], [0, 'A', 0, 0,
'G', 0], ['E', 0, 'B', 0, 'I', 0],
                                     [0, 'C', 0, 0, 0, 0], [0, 0, 'F', 0,
0, 'H'], [0, 0, 0, 'J', 0, 0]]))
warehouse2 = WareHouse(map=np.array([[0, 0, 'A', 0, 'P', 0], ['D', 0, 'B',
0, 'M', 0], ['E', 0, 'F', 0, 'K', 0],
                                     ['C', 0, 'H', 0, 0, 'O'], ['G', 0,
'J', 0, 0, 'Q'], ['I', 0, 0, 0, 'N', 0]]))
```

In [2]:

```
class Robot:
    """
    A class to represent a person.
    ...
    Attributes
    ----------
```

```
        warehouse: 2d array
            the ware house map
        order: list
            leftover items in the orders
        items: dict
            items that the robot has collected so far
        path: list
            the path the robot has followed,
        score: int
            if the robot goes to a grid square that does not have the
ordered item: -1 else + 3
        around: list
            an array of length 4 representing the shelf that has the
ordered item in corresponding
            to direction next to the robot in the pattern west / east /
north / south
    """

    @staticmethod
    def sensor(a_list: list):
        '''
        This function
        80% of time keeps the list unchanged
        10% changes element's value from 1 to 0 individually
        10% change element's value from 0 to 1 individually
        '''
        for i, v in enumerate(a_list):
            # If ran_num is from 2 to 9, the robot sensor works fine
            ran_num = random.randint(0, 9)
            if v == 0 and ran_num == 0:
                a_list[i] = 1
            elif v == 1 and ran_num == 1:
                a_list[i] = 0

    def __init__(self, a_ware_house: WareHouse, an_order: list,
items=dict(), around=[0, 0, 0, 0], path=[], score=0):
        self.warehouse = a_ware_house
        self.order = an_order
        self.items = items
        self.around = around
```

```
        self.path = path
        self.score = score
        self.rpos = 0
        self.cpos = 0

    def go_west(self):
        if self.cpos > 0:
            self.cpos -= 1

    def go_east(self):
        if self.cpos < len(self.warehouse.map[0]) - 1:
            self.cpos += 1

    def go_north(self):
        if self.rpos > 0:
            self.rpos -= 1

    def go_south(self):
        if self.rpos < len(self.warehouse.map) - 1:
            self.rpos += 1

    def peak_west(self):
        if self.cpos > 0:
            return self.cpos - 1

    def peak_east(self):
        if self.cpos < len(self.warehouse.map[0]) - 1:
            return self.cpos + 1

    def peak_north(self):
        if self.rpos > 0:
            return self.rpos - 1

    def peak_south(self):
        if self.rpos < len(self.warehouse.map) - 1:
            return self.rpos + 1

    def get_items(self, item, quantity):
        # Robot picks all items in a shelf that is included in an order
        self.items[item] = quantity
```

```
    def proceed_order(self):
        where_to_go = {
            0: self.go_west,
            1: self.go_east,
            2: self.go_north,
            3: self.go_south,
        }
        self.rpos = self.cpos = 0
        self.path.append([self.rpos, self.cpos])
        shelves_to_go = sorted(sorted(i[0] for i in self.order))
        while shelves_to_go:  # Until the robot picked up all items in an
order
            #             print(f'shelves_to_go: {shelves_to_go}')
            #             print(f'Move: {move}')
            #             print(f'Current position: {self.rpos,
self.cpos}')
            peak_all_directions = [[self.rpos, self.peak_west()],
[self.rpos, self.peak_east()],
                                   [self.peak_north(), self.cpos],
[self.peak_south(), self.cpos]]
            peak_all_directions = [i for i in peak_all_directions if None
not in i]
            #             print(f'peak_all_directions:
{peak_all_directions}')

            # Scan around to see how many surrounding shelves have the item
in the order
            if self.peak_west() and self.warehouse.map[self.rpos,
self.peak_west()] in shelves_to_go:
                #                 print(f'around 0 :
{self.warehouse.map[self.rpos, self.peak_west()]}')
                self.around[0] = 1
            if self.peak_east() and self.warehouse.map[self.rpos,
self.peak_east()] in shelves_to_go:
                #                 print(f'around 1 :
{self.warehouse.map[self.rpos, self.peak_east()]}')
                self.around[1] = 1
            if self.peak_north() and self.warehouse.map[self.peak_north(),
self.cpos] in shelves_to_go:
```

```
                #                 print(f'around 2 :
{self.warehouse.map[self.peak_north(), self.cpos]}')
                self.around[2] = 1
            if self.peak_south() and self.warehouse.map[self.peak_south(),
self.cpos] in shelves_to_go:
                #                 print(f'around 3 :
{self.warehouse.map[self.peak_south(), self.cpos]}')
                self.around[3] = 1

            # This is when the robot's sensor works
            self.sensor(self.around)

            # Determine the direction to move
            if sum(self.around) == 0:
                self.rpos, self.cpos = random.choice(peak_all_directions)
            #                 print(f'Current position after choice:
{self.rpos, self.cpos}')

            elif sum(self.around) == 1:
                # Find the index of the only grid square and move to that
only grid square
                where_to_go[self.around.index(1)]()
            #                 print(f'== 1 Current position after choice:
{self.rpos, self.cpos}')

            elif sum(self.around) > 1:
                # make a random choice between the positions involved
                index_of_directions = [i for i, v in enumerate(self.around)
if v == 1]
                where_to_go[random.choice(index_of_directions)]()
            #                 print(f'> 1 Current position after choice:
{self.rpos, self.cpos}')

            # Update the locations that the robot has followed so far
            self.path.append([self.rpos, self.cpos])

            # Determine what shelf it is
            name_of_shelf = self.warehouse.map[self.rpos, self.cpos]
            if name_of_shelf.isalpha() and name_of_shelf in shelves_to_go:
                #                 print(f'name of shelf: {name_of_shelf}')
```

```
                #                 print(f'order: {self.order}')
                # Pick up all items in the order that belong to a shelf
                self.score += 3
                for shelf, details in self.order:
                    #                     print(f'self.order:
{self.order}')
                    if shelf == name_of_shelf:
                        for code, quantity in details:
                            self.get_items(code, quantity)
                        shelves_to_go.remove(name_of_shelf)
            else:
                self.score -= 1
            # Reset around
            self.around = [0, 0, 0, 0]
#             print(f'score: {self.score}')
#             print()
```

In [3]:

```
def try_warehouses(warehouse, episodes=1000):
    avg_score = 0
    shortest_path, longest_path = [], []
    min_score = math.inf
    max_score = -math.inf
    for episode in range(episodes):
        # print(f'Episode {episode}')
        robot = Robot(warehouse, [('D', [('2166287989242', 3)])])
        robot.proceed_order()
        if min_score > robot.score:
            min_score = robot.score
            longest_path = robot.path[:]
        if max_score < robot.score:
            max_score = robot.score
            shortest_path = robot.path[:]
        avg_score += robot.score
    avg_score /= episodes
    print(f'Average score after {episodes} episodes is {avg_score}')
    print(f'Min score is : {min_score}')
    print(f'Max score is : {max_score}')
    print(f'The shortest path is {shortest_path}')
    print(f'The longest path is {longest_path}')
```

```
try_warehouses(warehouse1)
# try_warehouses(warehouse2)
```

```
Average score after 1000 episodes is -17.523
Min score is : -304
Max score is : 2
The shortest path is [[0, 0], [0, 1], [0, 2]]
The longest path is [[0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [0, 0], [1, 0], [2, 0], [1,
0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [0, 0], [1, 0], [2, 0], [2, 1], [1, 1], [1,
0], [1, 1], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [0, 1], [0, 2], [0,
0], [0, 0], [0, 1], [0, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [2, 0], [3,
0], [3, 1], [2, 1], [2, 0], [1, 0], [2, 0], [3, 0], [3, 1], [3, 2], [4, 2], [3, 2], [3,
3], [2, 3], [3, 3], [3, 4], [3, 5], [2, 5], [3, 5], [2, 5], [1, 5], [1, 4], [1, 3], [1,
4], [2, 4], [2, 3], [3, 3], [2, 3], [1, 3], [1, 4], [1, 3], [1, 4], [1, 3], [0, 3], [0,
2], [0, 0], [1, 0], [2, 0], [2, 1], [2, 2], [2, 3], [1, 3], [0, 3], [0, 4], [1, 4], [1,
5], [0, 5], [0, 4], [0, 5], [0, 4], [1, 4], [1, 5], [1, 4], [0, 4], [0, 5], [1, 5], [2,
5], [1, 5], [1, 4], [1, 3], [1, 4], [1, 5], [2, 5], [2, 4], [2, 5], [3, 5], [2, 5], [1,
5], [1, 4], [1, 5], [2, 5], [2, 5], [3, 5], [4, 5], [3, 5], [4, 5], [5, 5], [4, 5], [3,
5], [2, 5], [1, 5], [2, 5], [1, 5], [1, 4], [2, 4], [2, 3], [2, 4], [3, 4], [4, 4], [3,
4], [4, 4], [3, 4], [3, 5], [3, 4], [3, 5], [3, 5], [3, 4], [3, 5], [4, 5], [3, 5], [3,
4], [2, 4], [2, 3], [2, 2], [2, 1], [1, 1], [0, 1], [0, 2], [0, 0], [1, 0], [0, 0], [1,
0], [0, 0], [1, 0], [2, 0], [2, 1], [2, 0], [2, 1], [2, 0], [2, 1], [3, 1], [2, 1], [3,
1], [2, 1], [2, 0], [1, 0], [2, 0], [1, 0], [1, 1], [1, 2], [1, 3], [1, 2], [1, 1], [1,
0], [1, 1], [1, 0], [2, 0], [1, 0], [1, 1], [2, 1], [2, 2], [3, 2], [3, 1], [4, 1], [4,
0], [5, 0], [4, 0], [4, 1], [3, 1], [3, 2], [4, 2], [4, 3], [5, 3], [5, 2], [4, 2], [3,
2], [4, 2], [4, 3], [3, 3], [2, 3], [2, 4], [2, 3], [1, 3], [0, 3], [0, 2], [0, 0], [0,
0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [0, 0], [0, 0], [1, 0], [2, 0], [2, 0], [1,
0], [0, 0], [0, 0], [1, 0], [0, 0], [1, 0], [0, 0], [1, 0], [2, 0], [1, 0], [2, 0], [2,
1], [2, 2], [1, 2], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0,
2], [0, 0], [1, 0], [2, 0], [3, 0], [2, 0], [3, 0], [2, 0], [1, 0], [1, 1], [2, 1], [2,
0], [3, 0], [3, 1], [3, 2], [3, 1], [3, 0], [4, 0], [4, 1], [4, 2], [3, 2], [3, 3], [4,
3], [5, 3], [5, 3], [5, 4], [5, 5], [5, 4], [5, 3], [5, 4], [4, 4], [4, 5], [4, 4], [4,
3], [5, 3], [5, 2], [5, 2], [5, 1], [5, 2], [5, 1], [5, 0], [4, 0], [3, 0], [2, 0], [3,
0], [3, 1], [3, 2], [3, 1], [3, 2], [3, 3], [4, 3], [5, 3], [4, 3], [4, 4], [4, 5], [4,
5], [4, 4], [5, 4], [5, 4], [5, 5], [5, 5], [4, 5], [4, 4], [5, 4], [4, 4], [5, 4], [5,
3], [5, 4], [5, 5], [5, 4], [5, 4], [4, 4], [5, 4], [5, 5], [4, 5], [5, 5], [5, 4], [5,
5], [5, 5], [4, 5], [5, 5], [5, 4], [4, 4], [5, 4], [5, 3], [5, 4], [5, 5], [5, 5], [4,
5], [4, 4], [3, 4], [2, 4], [3, 4], [3, 5], [4, 5], [3, 5], [3, 4], [2, 4], [2, 3], [2,
4], [2, 3], [1, 3], [1, 2], [1, 1], [1, 0], [1, 1], [1, 0], [0, 0], [0, 1], [0, 2], [0,
0], [1, 0], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [1, 0], [1, 0], [2, 0], [3, 0], [3,
1], [2, 1], [2, 2], [1, 2], [1, 3], [2, 3], [1, 3], [2, 3], [2, 4], [2, 3], [2, 4], [2,
3], [1, 3], [2, 3], [3, 3], [4, 3], [4, 4], [4, 3], [4, 2], [5, 2], [5, 1], [4, 1], [4,
2], [4, 3], [4, 2], [3, 2], [2, 2], [2, 3], [2, 2], [3, 2], [4, 2], [4, 1], [3, 1], [3,
2], [4, 2], [5, 2], [5, 1], [5, 2], [5, 1], [4, 1], [4, 2], [5, 2], [4, 2], [4, 1], [4,
2], [4, 3], [3, 3], [4, 3], [4, 4], [4, 5], [4, 5], [4, 4], [4, 3], [5, 3], [5, 2], [5,
1], [5, 2], [4, 2], [4, 3], [4, 2], [3, 2], [3, 1], [4, 1], [5, 1], [5, 0], [5, 1], [5,
2], [5, 3], [5, 3], [5, 4], [5, 5], [5, 4], [5, 4], [5, 3], [5, 4], [5, 5], [4, 5], [3,
5], [3, 4], [4, 4], [4, 5], [4, 4], [5, 4], [5, 3], [5, 4], [5, 3], [4, 3], [4, 4], [4,
5], [4, 4], [5, 4], [5, 3], [5, 2], [4, 2], [5, 2], [5, 1], [5, 2], [5, 1], [5, 2], [5,
1], [5, 2], [4, 2], [5, 2], [5, 1], [5, 1], [5, 0], [5, 1], [5, 2], [5, 3], [5, 2], [5,
3], [5, 2], [5, 1], [5, 1], [5, 2], [5, 1], [5, 2], [5, 2], [4, 2], [4, 3], [4, 4], [4,
5], [4, 4], [4, 3], [5, 3], [4, 3], [5, 3], [5, 4], [5, 3], [4, 3], [5, 3], [5, 4], [5,
```

3], [5, 4], [5, 3], [5, 2], [4, 2], [4, 1], [4, 0], [4, 0], [5, 0], [5, 1], [5, 1], [5,
2], [5, 1], [5, 2], [4, 2], [5, 2], [5, 1], [5, 0], [4, 0], [4, 0], [4, 1], [4, 2], [5,
2], [4, 2], [3, 2], [4, 2], [5, 2], [5, 1], [5, 0], [4, 0], [4, 1], [3, 1], [3, 2], [4,
2], [4, 3], [4, 4], [4, 5], [4, 4], [4, 5], [5, 5], [5, 4], [4, 4], [4, 5], [4, 4], [4,
5], [3, 5], [3, 5], [3, 4], [2, 4], [2, 3], [2, 2], [2, 3], [2, 2], [2, 1], [2, 0], [3,
0], [2, 0], [2, 0], [2, 1], [1, 1], [1, 0], [2, 0], [3, 0], [2, 0], [3, 0], [3, 1], [2,
1], [2, 2], [2, 1], [3, 1], [4, 1], [3, 1], [4, 1], [4, 0], [4, 1], [4, 2], [4, 3], [4,
2], [3, 2], [4, 2], [4, 3], [4, 4], [4, 3], [4, 2], [5, 2], [5, 3], [5, 4], [5, 5], [5,
4], [4, 4], [5, 4], [4, 4], [4, 3], [4, 4], [4, 3], [4, 2], [4, 3], [5, 3], [5, 4], [5,
3], [5, 3], [4, 3], [4, 4], [3, 4], [2, 4], [1, 4], [1, 5], [2, 5], [3, 5], [3, 4], [3,
3], [3, 2], [3, 3], [3, 4], [2, 4], [2, 5], [3, 5], [2, 5], [2, 5], [2, 4], [3, 4], [2,
4], [1, 4], [2, 4], [2, 3], [1, 3], [1, 4], [0, 4], [0, 5], [1, 5], [1, 5], [1, 5], [1,
4], [1, 3], [0, 3], [0, 2], [0, 0], [1, 0], [2, 0], [2, 0], [1, 0], [2, 0], [1, 0], [0,
0], [1, 0], [2, 0], [2, 1], [2, 2], [2, 1], [2, 2], [2, 1], [1, 1], [2, 1], [2, 2], [3,
2], [2, 2], [2, 3], [1, 3], [2, 3], [2, 4], [1, 4], [2, 4], [3, 4], [3, 5], [3, 4], [3,
3], [2, 3], [3, 3], [4, 3], [4, 2], [4, 1], [4, 2], [5, 2], [5, 3], [5, 4], [4, 4], [3,
4], [3, 5], [3, 4], [3, 5], [4, 5], [5, 5], [5, 5], [5, 5], [4, 5], [3, 5], [3, 4], [3,
3], [3, 4], [2, 4], [2, 3], [1, 3], [0, 3], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0,
0], [1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [5, 0], [5, 1], [5, 2], [4, 2], [4, 3], [4,
2], [4, 1], [4, 2], [4, 3], [4, 4], [4, 5], [4, 5], [5, 5], [5, 4], [5, 3], [5, 2], [5,
3], [5, 4], [5, 5], [5, 4], [5, 3], [4, 3], [5, 3], [5, 2], [5, 1], [4, 1], [4, 2], [5,
2], [4, 2], [3, 2], [4, 2], [3, 2], [2, 2], [2, 1], [3, 1], [3, 0], [2, 0], [1, 0], [0,
0], [1, 0], [1, 1], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [1, 0], [2, 0], [2, 1], [3,
1], [2, 1], [2, 2], [1, 2], [2, 2], [2, 1], [1, 1], [1, 0], [2, 0], [3, 0], [3, 0], [3,
1], [4, 1], [3, 1], [3, 2], [2, 2], [2, 3], [2, 2], [2, 3], [2, 4], [1, 4], [1, 3], [2,
3], [1, 3], [1, 2], [1, 1], [2, 1], [2, 0], [1, 0], [0, 0], [1, 0], [2, 0], [2, 1], [1,
1], [1, 2], [1, 1], [1, 0], [1, 0], [1, 1], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 0], [0, 0], [0,
0], [0, 0], [1, 0], [2, 0], [2, 1], [2, 2], [3, 2], [3, 3], [3, 2], [2, 2], [1, 2], [0,
2], [0, 0], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [2,
0], [3, 0], [2, 0], [1, 0], [2, 0], [3, 0], [4, 0], [4, 1], [5, 1], [4, 1], [5, 1], [4,
1], [4, 0], [5, 0], [4, 0], [3, 0], [3, 1], [3, 0], [3, 1], [2, 1], [3, 1], [4, 1], [3,
1], [3, 2], [4, 2], [4, 1], [4, 0], [5, 0], [4, 0], [5, 0], [4, 0], [3, 0], [3, 1], [2,
1], [2, 0], [2, 1], [3, 1], [4, 1], [3, 1], [4, 1], [4, 2], [4, 3], [3, 3], [4, 3], [4,
2], [4, 3], [3, 3], [3, 2], [2, 2], [3, 2], [3, 1], [3, 2], [4, 2], [5, 2], [5, 3], [5,
3], [5, 4], [4, 4], [4, 3], [3, 3], [4, 3], [5, 3], [4, 3], [3, 3], [2, 3], [2, 2], [3,
2], [3, 1], [2, 1], [3, 1], [3, 0], [4, 0], [4, 1], [4, 2], [4, 3], [4, 2], [4, 3], [5,
3], [5, 2], [5, 1], [5, 0], [4, 0], [3, 0], [3, 1], [3, 2], [3, 1], [3, 2], [4, 2], [4,
1], [3, 1], [2, 1], [2, 0], [2, 1], [2, 0], [2, 0], [2, 1], [1, 1], [0, 1], [0, 2], [0,
0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [1, 1], [1, 2], [0, 2], [0, 0], [0,
0], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [1, 2], [0, 2], [0, 0], [0, 1], [0, 2], [0,
0], [1, 0], [2, 0], [1, 0], [1, 1], [2, 1], [3, 1], [4, 1], [3, 1], [3, 2], [2, 2], [1,
2], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
0], [1, 0], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0,
1], [1, 1], [2, 1], [2, 2], [2, 3], [1, 3], [1, 4], [0, 4], [1, 4], [0, 4], [0, 5], [0,
5], [0, 5], [0, 4], [0, 5], [1, 5], [0, 5], [0, 4], [0, 3], [0, 2], [0, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 0], [1, 0], [2, 0], [2, 1], [2, 0], [2, 1], [1, 1], [1, 2], [0,
2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [0, 0], [0, 1], [0, 0], [0, 1], [0,
2], [0, 0], [1, 0], [2, 0], [3, 0], [2, 0], [2, 1], [2, 2], [2, 3], [2, 2], [3, 2], [3,
1], [3, 2], [3, 1], [4, 1], [4, 0], [3, 0], [4, 0], [3, 0], [3, 0], [3, 1], [4, 1], [3,
1], [2, 1], [2, 0], [2, 0], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [2,
1], [3, 1], [3, 0], [3, 0], [2, 0], [3, 0], [3, 1], [4, 1], [4, 0], [4, 1], [4, 2], [4,

1], [5, 1], [4, 1], [4, 0], [3, 0], [4, 0], [4, 1], [4, 0], [3, 0], [3, 1], [4, 1], [4,
2], [3, 2], [4, 2], [4, 3], [3, 3], [4, 3], [3, 3], [3, 2], [3, 3], [4, 3], [4, 2], [4,
3], [4, 4], [4, 3], [4, 2], [3, 2], [4, 2], [5, 2], [4, 2], [3, 2], [3, 3], [2, 3], [2,
4], [3, 4], [3, 3], [3, 2], [4, 2], [5, 2], [5, 3], [5, 4], [5, 5], [5, 4], [5, 3], [5,
2], [5, 3], [5, 3], [5, 2], [5, 1], [5, 0], [5, 1], [5, 2], [5, 3], [5, 4], [5, 3], [4,
3], [4, 4], [4, 3], [5, 3], [5, 4], [5, 4], [5, 5], [5, 4], [5, 3], [5, 4], [4, 4], [3,
4], [2, 4], [3, 4], [2, 4], [2, 3], [3, 3], [4, 3], [5, 3], [5, 4], [5, 3], [4, 3], [3,
3], [2, 3], [2, 2], [3, 2], [3, 3], [2, 3], [3, 3], [3, 2], [3, 1], [3, 2], [3, 1], [2,
1], [3, 1], [2, 1], [3, 1], [3, 2], [3, 3], [2, 3], [2, 2], [2, 3], [1, 3], [1, 4], [1,
3], [1, 2], [1, 1], [2, 1], [2, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0,
0], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [1, 0], [2, 0], [3,
0], [2, 0], [2, 1], [2, 0], [3, 0], [2, 0], [1, 0], [2, 0], [2, 1], [3, 1], [3, 2], [3,
3], [2, 3], [2, 2], [2, 3], [1, 3], [1, 2], [2, 2], [2, 1], [2, 2], [2, 1], [1, 1], [2,
1], [1, 1], [1, 2], [1, 1], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0,
0], [0, 0], [1, 0], [1, 0], [0, 0], [0, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [2, 1], [1, 1], [1, 2], [1, 1], [1,
0], [2, 0], [3, 0], [2, 0], [3, 0], [3, 0], [4, 0], [4, 1], [4, 0], [4, 0], [5, 0], [5,
1], [5, 0], [5, 1], [5, 2], [4, 2], [5, 2], [5, 3], [5, 2], [5, 3], [4, 3], [4, 2], [5,
2], [5, 3], [4, 3], [4, 4], [4, 3], [3, 3], [3, 4], [3, 3], [2, 3], [2, 2], [2, 1], [1,
1], [0, 1], [0, 2], [0, 0], [1, 0], [2, 0], [2, 0], [1, 0], [1, 0], [0, 0], [0, 1], [0,
2], [0, 0], [0, 0], [1, 0], [1, 0], [1, 1], [2, 1], [2, 0], [1, 0], [2, 0], [2, 1], [3,
1], [4, 1], [4, 2], [5, 2], [5, 2], [4, 2], [4, 3], [4, 2], [4, 1], [3, 1], [4, 1], [5,
1], [5, 2], [5, 1], [5, 0], [5, 0], [4, 0], [4, 1], [4, 0], [5, 0], [4, 0], [4, 1], [3,
1], [3, 2], [2, 2], [3, 2], [4, 2], [5, 2], [5, 3], [5, 4], [5, 5], [5, 4], [5, 4], [5,
3], [4, 3], [4, 2], [4, 1], [5, 1], [4, 1], [5, 1], [5, 2], [5, 3], [4, 3], [4, 2], [4,
3], [5, 3], [5, 3], [5, 3], [5, 3], [4, 3], [4, 4], [3, 4], [3, 5], [3, 4], [4, 4], [5,
4], [4, 4], [4, 3], [4, 4], [4, 5], [5, 5], [4, 5], [5, 5], [4, 5], [5, 5], [5, 5], [4,
5], [4, 5], [5, 5], [4, 5], [4, 4], [3, 4], [3, 5], [3, 4], [2, 4], [3, 4], [3, 3], [2,
3], [2, 4], [2, 5], [1, 5], [2, 5], [3, 5], [2, 5], [1, 5], [1, 4], [1, 3], [1, 4], [0,
4], [0, 5], [1, 5], [0, 5], [0, 4], [0, 5], [0, 5], [1, 5], [0, 5], [1, 5], [1, 4], [1,
3], [1, 2], [0, 2], [0, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 1], [1, 1], [1, 2], [1, 3], [1, 4], [2, 4], [3, 4], [2, 4], [3,
4], [3, 3], [3, 4], [2, 4], [2, 5], [3, 5], [4, 5], [4, 4], [3, 4], [3, 3], [3, 4], [3,
3], [4, 3], [5, 3], [4, 3], [4, 4], [3, 4], [3, 3], [2, 3], [2, 2], [2, 1], [2, 0], [1,
0], [2, 0], [2, 0], [2, 0], [3, 0], [4, 0], [4, 1], [5, 1], [5, 0], [4, 0], [4, 1], [4,
0], [3, 0], [2, 0], [2, 1], [2, 0], [2, 1], [2, 0], [1, 0], [2, 0], [1, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0,
0], [0, 1], [1, 1], [1, 0], [1, 0], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0,
2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [2,
0], [2, 1], [3, 1], [4, 1], [3, 1], [3, 0], [3, 1], [3, 2], [3, 1], [3, 0], [2, 0], [2,
1], [3, 1], [3, 2], [3, 1], [3, 2], [3, 1], [2, 1], [3, 1], [2, 1], [3, 1], [3, 0], [2,
0], [1, 0], [0, 0], [1, 0], [1, 1], [1, 0], [2, 0], [3, 0], [3, 0], [4, 0], [5, 0], [5,
1], [5, 2], [4, 2], [4, 1], [4, 0], [5, 0], [5, 1], [5, 2], [4, 2], [4, 3], [3, 3], [2,
3], [3, 3], [2, 3], [3, 3], [4, 3], [3, 3], [3, 2], [2, 2], [3, 2], [2, 2], [2, 3], [2,
4], [3, 4], [2, 4], [2, 5], [3, 5], [2, 5], [2, 4], [1, 4], [0, 4], [1, 4], [2, 4], [3,
4], [4, 4], [4, 3], [3, 3], [3, 2], [3, 1], [2, 1], [1, 1], [1, 2], [0, 2], [0, 0], [0,
0], [1, 0], [1, 1], [0, 1], [1, 1], [1, 0], [1, 0], [0, 0], [1, 0], [2, 0], [3, 0], [2,
0], [3, 0], [4, 0], [4, 1], [4, 0], [4, 0], [4, 1], [4, 2], [4, 3], [4, 4], [3, 4], [3,
3], [2, 3], [2, 2], [3, 2], [2, 2], [2, 1], [1, 1], [1, 2], [2, 2], [2, 3], [1, 3], [0,
3], [0, 4], [0, 5], [0, 4], [0, 4], [1, 4], [1, 3], [0, 3], [0, 2], [0, 0], [0, 0], [1,
0], [2, 0], [3, 0], [2, 0], [3, 0], [2, 0], [1, 0], [1, 1], [1, 0], [0, 0], [1, 0], [1,
0], [2, 0], [2, 0], [1, 0], [1, 1], [0, 1], [1, 1], [1, 2], [0, 2], [0, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 0], [1, 0], [0, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
0], [1, 0], [1, 0], [2, 0], [2, 1], [2, 2], [2, 1], [3, 1], [3, 2], [3, 3], [3, 2], [3,

3], [4, 3], [3, 3], [2, 3], [2, 2], [2, 3], [2, 2], [2, 1], [1, 1], [1, 0], [2, 0], [2,
0], [2, 1], [2, 0], [2, 1], [2, 0], [2, 0], [1, 0], [0, 0], [1, 0], [1, 0], [0, 0], [0,
1], [0, 2], [0, 0], [0, 0], [1, 0], [2, 0], [1, 0], [1, 1], [2, 1], [2, 2], [1, 2], [0,
2], [0, 0], [0, 0], [0, 1], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [2,
0], [1, 0], [1, 1], [1, 0], [1, 1], [2, 1], [1, 1], [0, 1], [0, 2], [0, 0], [0, 0], [1,
0], [2, 0], [2, 1], [3, 1], [3, 2], [3, 3], [3, 4], [3, 3], [3, 2], [3, 1], [2, 1], [2,
2], [2, 3], [1, 3], [2, 3], [3, 3], [2, 3], [3, 3], [2, 3], [1, 3], [1, 4], [1, 3], [0,
3], [0, 4], [1, 4], [2, 4], [2, 5], [3, 5], [3, 4], [2, 4], [2, 5], [1, 5], [0, 5], [0,
4], [1, 4], [2, 4], [2, 5], [1, 5], [0, 5], [0, 4], [0, 3], [1, 3], [0, 3], [0, 2], [0,
0], [1, 0], [0, 0], [1, 0], [1, 1], [2, 1], [1, 1], [1, 0], [2, 0], [2, 1], [2, 0], [2,
1], [3, 1], [4, 1], [5, 1], [4, 1], [5, 1], [5, 0], [4, 0], [3, 0], [4, 0], [4, 1], [5,
1], [4, 1], [3, 1], [4, 1], [4, 2], [4, 1], [4, 2], [4, 1], [5, 1], [5, 2], [4, 2], [5,
2], [5, 3], [5, 2], [5, 3], [5, 4], [4, 4], [4, 5], [5, 5], [4, 5], [4, 4], [5, 4], [5,
3], [4, 3], [3, 3], [4, 3], [4, 4], [5, 4], [4, 4], [5, 4], [5, 5], [5, 4], [5, 5], [4,
5], [4, 4], [4, 3], [4, 2], [4, 1], [5, 1], [4, 1], [3, 1], [4, 1], [5, 1], [4, 1], [3,
1], [3, 0], [2, 0], [2, 0], [3, 0], [3, 1], [3, 2], [4, 2], [4, 3], [4, 2], [4, 3], [4,
2], [4, 3], [3, 3], [3, 4], [3, 3], [2, 3], [1, 3], [1, 4], [1, 3], [1, 2], [1, 3], [2,
3], [2, 4], [2, 5], [3, 5], [4, 5], [4, 4], [4, 3], [4, 4], [5, 4], [5, 3], [5, 2], [5,
1], [5, 2], [5, 1], [4, 1], [3, 1], [2, 1], [1, 1], [0, 1], [0, 2], [0, 0], [0, 0], [0,
0], [0, 0], [0, 0], [0, 0], [0, 0], [1, 0], [0, 0], [1, 0], [1, 1], [1, 2], [2, 2], [1,
2], [0, 2], [0, 0], [0, 0], [1, 0], [1, 1], [2, 1], [1, 1], [0, 1], [0, 2], [0, 0], [0,
0], [0, 1], [0, 2], [0, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0,
0], [1, 0], [1, 1], [1, 2], [1, 3], [0, 3], [0, 2], [0, 0], [0, 0], [1, 0], [2, 0], [3,
0], [4, 0], [5, 0], [5, 0], [5, 1], [5, 1], [4, 1], [4, 0], [4, 1], [5, 1], [4, 1], [3,
1], [2, 1], [1, 1], [2, 1], [3, 1], [4, 1], [4, 0], [3, 0], [4, 0], [5, 0], [5, 0], [4,
0], [4, 1], [5, 1], [5, 2], [4, 2], [5, 2], [5, 1], [5, 0], [5, 0], [4, 0], [4, 1], [3,
1], [3, 0], [4, 0], [4, 1], [4, 2], [3, 2], [3, 1], [4, 1], [5, 1], [4, 1], [4, 0], [5,
0], [4, 0], [5, 0], [5, 0], [4, 0], [3, 0], [4, 0], [5, 0], [4, 0], [3, 0], [2, 0], [1,
0], [1, 1], [1, 2], [2, 2], [1, 2], [2, 2], [1, 2], [1, 3], [1, 2], [1, 1], [0, 1], [0,
2], [0, 0], [1, 0], [0, 0], [0, 0], [0, 0], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
1], [1, 1], [2, 1], [3, 1], [4, 1], [3, 1], [2, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5,
1], [5, 2], [4, 2], [5, 2], [4, 2], [4, 1], [4, 0], [3, 0], [2, 0], [3, 0], [2, 0], [3,
0], [2, 0], [2, 0], [2, 1], [2, 2], [2, 3], [2, 2], [2, 1], [3, 1], [4, 1], [5, 1], [5,
0], [4, 0], [5, 0], [4, 0], [5, 0], [5, 1], [4, 1], [4, 2], [5, 2], [4, 2], [4, 1], [3,
1], [3, 0], [3, 1], [3, 0], [3, 1], [3, 2], [3, 3], [3, 4], [3, 5], [3, 4], [4, 4], [4,
5], [4, 4], [5, 4], [4, 4], [4, 5], [4, 4], [5, 4], [5, 4], [5, 4], [5, 5], [4, 5], [4,
5], [4, 4], [4, 3], [4, 2], [5, 2], [5, 1], [5, 2], [5, 3], [5, 3], [4, 3], [5, 3], [4,
3], [4, 4], [5, 4], [4, 4], [4, 3], [4, 2], [3, 2], [2, 2], [3, 2], [3, 1], [3, 2], [3,
3], [4, 3], [4, 4], [5, 4], [5, 3], [5, 4], [4, 4], [3, 4], [2, 4], [1, 4], [1, 3], [1,
4], [0, 4], [0, 5], [0, 5], [0, 4], [0, 3], [0, 4], [0, 3], [0, 2], [0, 0], [0, 1], [0,
2], [0, 0], [0, 1], [0, 0], [0, 0], [1, 0], [1, 1], [1, 0], [0, 0], [1, 0], [2, 0], [1,
0], [2, 0], [2, 1], [1, 1], [1, 2], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0,
1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1,
0], [2, 0], [3, 0], [3, 1], [2, 1], [1, 1], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0,
0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [1, 0], [0, 0], [0, 1], [0,
2], [0, 0], [0, 0], [1, 0], [2, 0], [3, 0], [2, 0], [1, 0], [0, 0], [0, 1], [0, 2], [0,
0], [0, 0], [0, 1], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [1,
1], [2, 1], [2, 2], [1, 2], [1, 1], [1, 0], [1, 1], [0, 1], [0, 2], [0, 0], [0, 1], [0,
2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [0, 0], [0, 1], [0, 2], [0,
0], [0, 1], [0, 2], [0, 0], [0, 1], [0, 2], [0, 0], [1, 0], [1, 1], [1, 2], [1, 3], [1,
4], [1, 5], [2, 5], [3, 5], [4, 5], [4, 4], [5, 4], [5, 4], [5, 4], [5, 5], [4, 5], [3,
5], [3, 4], [3, 3], [3, 4], [3, 3], [2, 3], [2, 2], [1, 2], [1, 1], [1, 2], [2, 2], [1,
2], [2, 2], [2, 3], [3, 3], [4, 3], [5, 3], [5, 4], [5, 4], [5, 4], [5, 5], [4, 5], [5,
5], [5, 4], [5, 3], [4, 3], [5, 3], [5, 4], [5, 3], [5, 3], [4, 3], [5, 3], [5, 4], [5,

3], [5, 4], [5, 4], [5, 3], [5, 2], [4, 2], [3, 2], [4, 2], [3, 2], [4, 2], [5, 2], [4,
2], [5, 2], [5, 3], [5, 2], [5, 3], [4, 3], [3, 3], [2, 3], [2, 4], [2, 3], [3, 3], [3,
4], [4, 4], [3, 4], [3, 5], [4, 5], [5, 5], [5, 4], [5, 4], [5, 5], [4, 5], [5, 5], [5,
4], [5, 3], [5, 4], [4, 4], [3, 4], [3, 5], [4, 5], [4, 5], [4, 4], [4, 3], [4, 2], [4,
3], [5, 3], [4, 3], [5, 3], [5, 4], [4, 4], [4, 3], [5, 3], [4, 3], [5, 3], [5, 4], [5,
5], [5, 4], [5, 5], [5, 4], [4, 4], [5, 4], [5, 3], [4, 3], [4, 2], [4, 1], [4, 0], [3,
0], [3, 1], [4, 1], [4, 2], [4, 3], [3, 3], [2, 3], [3, 3], [3, 4], [3, 3], [3, 4], [2,
4], [3, 4], [3, 5], [3, 4], [3, 3], [4, 3], [5, 3], [5, 2], [4, 2], [5, 2], [5, 3], [4,
3], [4, 4], [3, 4], [3, 3], [4, 3], [4, 2], [4, 3], [4, 4], [4, 3], [5, 3], [4, 3], [4,
4], [4, 5], [4, 4], [5, 4], [5, 5], [4, 5], [5, 5], [4, 5], [4, 4], [3, 4], [3, 3], [4,
3], [4, 2], [4, 1], [5, 1], [4, 1], [4, 2], [4, 3], [4, 4], [5, 4], [5, 4], [4, 4], [4,
5], [3, 5], [2, 5], [1, 5], [0, 5], [1, 5], [1, 5], [0, 5], [0, 5], [1, 5], [0, 5], [0,
4], [1, 4], [1, 5], [2, 5], [2, 4], [3, 4], [3, 3], [3, 4], [4, 4], [5, 4], [5, 3], [4,
3], [4, 4], [4, 3], [3, 3], [2, 3], [2, 4], [1, 4], [0, 4], [1, 4], [1, 3], [1, 2], [2,
2], [3, 2], [4, 2], [4, 3], [4, 2], [4, 3], [4, 4], [4, 5], [4, 5], [3, 5], [3, 4], [3,
3], [4, 3], [4, 4], [4, 3], [4, 4], [4, 3], [5, 3], [5, 4], [5, 5], [5, 5], [4, 5], [4,
4], [4, 3], [3, 3], [2, 3], [3, 3], [4, 3], [4, 2], [3, 2], [4, 2], [4, 1], [3, 1], [4,
1], [3, 1], [4, 1], [4, 0], [5, 0], [5, 0], [5, 0], [4, 0], [5, 0], [5, 1], [5, 0], [4,
0], [4, 1], [3, 1], [3, 0], [4, 0], [3, 0], [4, 0], [3, 0], [3, 1], [3, 2], [4, 2], [4,
3], [4, 4], [4, 5], [3, 5], [3, 4], [3, 5], [4, 5], [3, 5], [4, 5], [4, 4], [5, 4], [4,
4], [3, 4], [3, 3], [4, 3], [5, 3], [5, 4], [5, 5], [5, 5], [4, 5], [5, 5], [5, 4], [5,
3], [4, 3], [5, 3], [5, 2], [5, 1], [4, 1], [4, 2], [4, 3], [4, 2], [4, 1], [4, 0], [4,
1], [4, 2], [4, 3], [4, 4], [3, 4], [2, 4], [2, 5], [2, 5], [3, 5], [2, 5], [1, 5], [1,
5], [2, 5], [1, 5], [1, 4], [0, 4], [0, 3], [0, 2]]