1
a)

0 - 0
1 - 1
9 - 2
0 - 9
1 - 12
0 - 0

b)

```
def is_perfect(n):
        divisor_sum = 0
        for i in range(1, n):
                if n % i == 0:
                        divisor_sum += i
        return divisor_sum == n


for n in range(1, 5000):
        if is_perfect(n):
                print(n)
```

c)

```
def max_column_values(m):
        num_cols = len(m[0])
        num_rows = len(m)

        a_list = []

        for col_i in range(num_cols):
                highest = m[0][col_i]
                for row_i in range(1, num_rows):
                        value = m[row_i][col_i]
                        if value > highest:
                                highest = value
                # highest is highest
                a_list.append(highest)

        return a_list
```

d)

3 4
2 2
2 4
2 7
2 9
2 11

2)

a)

```
class FootballTeam:
  def __init__(self):
    self.players = []


  def add(self,player) :
    self.players.append(player)
```

b) Add to class FootballPlayer:

```
TOP_SCORER_GOALS_PER_GAME = 0.4

def average_goals_per_game(self):
   return self.number_of_goals / self.number_of_games

def is_top_scorer(self):
  return self.average_goals_per_game() >= TOP_SCORER_GOALS_PER_GAME
```

Add to class FootballTeam:

```
def select_top_scorers(self):
  res = FootballTeam()
  for player in self.players:
    if player.is_top_scorer() :
      res.add(player)
  return res
```

c) Add to FootballPlayer:

```
def is_inactive(self) :
  return  self.injured or self.red_card
```

Add to FootballTeam:

```
def remove_inactive(self):
  # Go from right to left, as del self.player[i] moves all elements > i
  # one to the left
  for i in range(len(self.players), -1, -1) :
    if self.players[i].is_inactive():
      del self.players[i]
```

d)

Add functions:

```
def get_best_in_post(sorted_players, position, amount) :
  selection = []
  for player in sorted_players:
    if player.position == position:
      selection.append(player)
  return selection[:amount]

def select_formation(team):
  sorted = sort_players(team.players)
  result = []
  result += get_best_in_pos(sorted, "keeper",1)
  result += get_best_in_pos(sorted, "defender", 4)
  result += get_best_in_pos(sorted, "mindfielder", 4)
  result += get_best_in_pos(sorted, "striker", 2)
  return result
```

3)

global variables:

```
thing = ui.WALL



def update():
        global thing
```

```
# your code goes here
for y in range(0, HEIGHT, 2):
        for x in range(0, WIDTH, 2):
                ui.place(x, y, thing)
for y in range(1, HEIGHT, 2):
        for x in range(1, WIDTH, 2):
                ui.place(x, y, thing)

if thing == ui.WALL:
        thing = ui.SNAKE
else:
        thing = ui.WALL
```