

Test3Q

Time Limit: 2:00:00

Time Left:1:41:23

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 ---	3 ---
4 ✓	5 ✓	6 ---
7 ✓		

Question 1 (2 points) ✓ Saved

Consider that this program is running and is about to print "bye dragonfly".

```
def dragonfly(y, b):  
    y[1]=0  
    b=30  
    z=y  
    z[0]=1000  
    print("bye dragonfly")
```

```
x=[1,2,3]  
a=22  
dragonfly(x,a)
```

Which one of the following statement is true about variables x, y and z.

- ☐ x and y refer to a same object and z to a different object
- ☒ All three variables x, y and z refer to a same object
- ☐ x and z refer to a same object and y to a different object
- ☐ The 3 variables refer to three distinct objects
- ☐ y and z refer to a same object and x to a different object

Question 2 (10 points)

Suppose you need to make teams out of a list of players. Here is one way to do it that you will need to implement in this question. If, for example, we want 4 teams, we need to go through the list of people, assigning them, in order, to teams 0, 1, 2, 3, 0, 1, 2, 3, etc. You will need to write a function make_teams(players, num_teams) that forms teams in this way. For example, if the list of people is: ["pavle", "filip", "ana", "severin", "sanja", "ari", "tvrtko", "denis", "nela", "maja", "ziko", "dijana", "aca", "dubrayka", "senka"]

Test3Q

Time Limit: 2:00:00

Time Left:1:41:20

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 ---	3 ---
4 ✓	5 ✓	6 ---
7 ✓		

Question 2 (10 points)

Suppose you need to make teams out of a list of players. Here is one way to do it that you will need to implement in this question. If, for example, we want 4 teams, we need to go through the list of people, assigning them, in order, to teams 0, 1, 2, 3, 0, 1, 2, 3, etc. You will need to write a function `make_teams(players, num_teams)` that forms teams in this way. For example, if the list of people is: `["pavle", "filip", "ana", "severin", "sanja", "ari", "tvrtko", "denis", "nela", "maja", "ziko", "dijana", "aca", "dubravka", "senka"]`

and the number of teams desired is four, it should return:

```
[['pavle', 'sanja', 'nela', 'aca'], ['filip', 'ari', 'maja', 'dubravka'], ['ana', 'tvrtko', 'ziko', 'senka'], ['severin', 'denis', 'dijana']]
```

Write the body of the function (with the docstrings that follows) in a `make_teams_yourID.py` file and **attach** it in the response window. Do not assume that `len(player)` is divisible by the number of teams. Do not assume that `len(player)` is at least as large as the number of teams. If there aren't enough players to cover the teams, some (or even all) teams will simply be empty.

The following docstrings provide you a better understanding of the process:

```
def make_teams(players, num_teams):  
    """(list of str, int)->2D list  
    Make num_teams teams out of the players in list players by counting off.  
    Players is a list of players' names and num_teams is the desired number of teams  
    Return a 2D list where each sublist is representing a team.  
    Preconditions: num_teams >= 1
```

```
>>> make_teams(["pele", "maradona", "serena", "venus", "fed", "rafa", "lionel"], 3)  
[['pele', 'venus', 'lionel'], ['maradona', 'fed'], ['serena', 'rafa']]  
>>> make_teams(["pele", "maradona", "serena", "venus", "fed", "rafa", "lionel"], 2)  
[['pele', 'serena', 'fed', 'lionel'], ['maradona', 'venus', 'rafa']]
```

Test3Q



Time Limit: 2:00:00

Time Left:1:41:19

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 --	3 --
4 ✓	5 ✓	6 --
7 ✓		

def make_teams(players, num_teams):

"""(list of str, int)->2D list

Make num_teams teams out of the players in list players by counting off.

Players is a list of players' names and num_teams is the desired number of teams

Return a 2D list where each sublist is representing a team.

Preconditions: num_teams >= 1

```
>>> make_teams(["pele", "maradona", "serena", "venus", "fed", "rafa", "lionel"], 3)
[['pele', 'venus', 'lionel'], ['maradona', 'fed'], ['serena', 'rafa']]
>>> make_teams(["pele", "maradona", "serena", "venus", "fed", "rafa", "lionel"], 2)
[['pele', 'serena', 'fed', 'lionel'], ['maradona', 'venus', 'rafa']]
>>> make_teams(["1", "2", "3", "4", "5", "6", "7", "8", "9"], 3)
[['1', '4', '7'], ['2', '5', '8'], ['3', '6', '9']]
>>> make_teams(["1", "2", "3", "4", "5", "6", "7", "8", "9"], 1)
[['1', '2', '3', '4', '5', '6', '7', '8', '9']]
>>> make_teams(["1", "2", "3", "4", "5", "6", "7", "8", "9"], 4)
[['1', '5', '9'], ['2', '6'], ['3', '7'], ['4', '8']]
>>> make_teams(["1", "2", "3", "4", "5", "6", "7", "8", "9"], 7)
[['1', '8'], ['2', '9'], ['3'], ['4'], ['5'], ['6'], ['7']]
>>> make_teams(["1", "2", "3", "4", "5", "6", "7", "8", "9"], 11)
[['1'], ['2'], ['3'], ['4'], ['5'], ['6'], ['7'], ['8'], ['9'], [], []]
>>> make_teams( [], 3)
[[], [], []]
"""
```

#YOUR CODE GOES HERE

pass



Test3Q

Time Limit: 2:00:00

Time Left:1:41:16

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 ---	3 ---
4 ✓	5 ✓	6 ---
7 ✓		

Question 3 (10 points)

Write a function called `diff_start_end(l, q)` as described in the docstrings below:

```
def diff_start_end(l, q):  
    """(list of int, list of int) -> (int, int)
```

`l` and `q` are two parallel lists i.e. `len(l)==len(q)`

The function returns the following two integers (i.e. a tuple with the following two integers).

- the index where `l` and `q` first differ, and
- the index where `l` and `q` last differ. This index should be **NEGATIVE**

Preconditions: `len(l)==len(q)>=1`, and `l != q` *******(The 2 lists are NOT the same)*******

```
>>> diff_start_end([0,1,6,0,7,2,3,4], [0,1,8,0,9,2,3,4])  
(2, -4)  
>>> diff_start_end([6],[8])  
(0, -1)  
>>> diff_start_end([0,0,0,1,0],[0,0,0,22,0])  
(3, -2)  
>>> diff_start_end([0,1,1],[1,1,1])  
(0, -3)  
>>> diff_start_end([0,1,1,0],[0,0,0,0])  
(1, -2)  
"""
```

#YOUR CODE GOES HERE

pass

Copy and paste the above docstrings into a `diff_start_end_yourID.py` file, add your code to it and **attach** it in the response window.

Test3Q

Time Limit: 2:00:00

Time Left:1:41:14

Kian Zahrai: Attempt 1

Page 1:

Add a File

Record Audio

Record Video

1 ✓	2 ---	3 ---
4 ✓	5 ✓	6 ---
7 ✓		

Question 4 (2 points) ✓ Saved

What is the running-time (i.e. a rough number of operations) of the following function, called *mantis*, on a list with n elements?

```
def mantis(L):  
    '''(list of int)->None'''  
    n=len(L)  
    for i in range(n):  
        L[i]=L[i]**2  
    print("Elements of L are:")  
    for i in range(n):  
        print(L[i])
```

- ☐ $O(n \log n)$
- ☐ $O(\log n)$
- ☒ $O(n)$
- ☐ $O(n^2)$
- ☐ $O(1)$

Question 5 (2 points) ✓ Saved

Consider that this program is running and is about to print "bye dragonfly".

```
def dragonfly(y, b):  
    y[1]=0  
    b=30  
    z=y
```

Kian Zahrai: Attempt 1

○ $O(1)$

1 ✓	2 000 000	3 000 000
4 ✓	5 ✓	6 000 000
7 ✓		

Consider that this program is running and is about to print "bye dragonfly".

```
x=[1,2,3]
a=22
dragonfly(x,a)
```

Which one of the following statement is true about variables a and b.

- ☒ Both a and b refer to a same object
- ☐ only b is an object
- ☐ a and b refer to two distinct objects
- ☐ only a is an object

Write a function `draw_w_stars(n)` that draws the images below where `n` is the number of the stars in on the top and bottom of the drawing. You may assume that `n` is a positive odd integer.

```
>>> draw_w_stars(9)      >>> draw_w_stars(7)      >>> draw_w_stars(5)
```

Test3Q

Time Limit: 2:00:00

Time Left:1:41:10

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 ---	3 ---
4 ✓	5 ✓	6 ---
7 ✓		

Question 6 (10 points)

Write a function `draw_w_stars(n)` that draws the images below where `n` is the number of the stars in on the top and bottom of the drawing. You may assume that `n` is a positive odd integer.

```
>>> draw_w_stars(9)      >>> draw_w_stars(7)      >>> draw_w_stars(5)
*****
*   *
*   *
*   *
*   *
*   *
*   *
*   *
*   *
*****

*****
*   *
*   *
*   *
*   *
*   *
*   *
*****

*****
*   *
*   *
*   *
*****

>>> draw_w_stars(3)      >>> draw_w_stars(1)
***
*
***

*

```

Copy and paste the following into a `draw_w_stars_YourID.py` file add your code and attach it to the response window.

```
def draw_w_stars(n):
    """(int)->None
    Preconditions: n is positive odd integer
    Draws a figure as depicted in Question 5 with n stars in top and bottom rows
    """

    #YOUR CODE GOES HERE

    pass
```

Test3Q

Time Limit: 2:00:00

Time Left:1:41:07

Kian Zahrai: Attempt 1

Page 1:

1 ✓	2 --	3 --
4 ✓	5 ✓	6 --
7 ✓		

Add a File

Record Audio

Record Video

Question 7 (2 points) ✓ Saved

If we replace n in the second range function, in the function *mantis*, with the number 10.

What is the running-time of function *mantis* with that change?

```
def mantis(L):  
    '''(list of int)->None'''  
    n=len(L)  
    for i in range(n):  
        L[i]=L[i]**2  
    print("Elements of L are:")  
    for i in range(n):  
        print(L[i])
```

☐ $O(1)$

☒ $O(n)$

☐ $O(n^2)$

☐ $O(n \log n)$

☐ $O(\log n)$

Submit Quiz

4 of 7 questions saved