

CS1010E – AY2020/21 S1

Assignment 2 – Writing Functions

程序代写代做 CS编程辅导

Problem 1: Drawing Functions with Turtle [30 marks]

Important:

1. You must import the turtle module as follows:
`from turtle import *`
2. Do not import the turtle module as `import turtle`.
3. You may assume the functions defined in the subparts have been correctly defined and made available to you. Simply call the function in your code. eg. if you wish to reuse the function `draw_polygon` from part b in part c, just call `draw_polygon` wherever you wish to draw a polygon in the code.



We started learning how to draw simple graphics with the Python Turtle package in Assignment 1. In this assignment, we will build upon those basic turtle functions to create more sophisticated graphics.

- a. Write a function `draw_square(d)` to draw a square of length `d`. The start and end point of the rectangle should be at the initial point of the turtle (*bottom-left corner of the square*). Check Coursemology for examples. [10 marks]
- b. Write a function `draw_polygon(d, n)` to draw a regular polygon with `n` sides where the length of each side is `d`. The start and end point of the polygon should be at the initial point of the turtle, in the bottom left corner of the image. Check Coursemology for examples. [10 marks]
- c. Write a function `draw_flower(d, n, p)` to draw a flower pattern where `d` is the length of the regular polygon, `n` is the number of sides of the polygon and `p` is the number of petals. You are required to draw the following graphic which consists of 10 petals. Each petal is a regular octagon with 100-unit long sides. [10 marks]

WeChat: estutorcs
Assignment Project Exam Help
Email: tutorc@163.com
QQ: 749389476
<https://tutorcs.com>

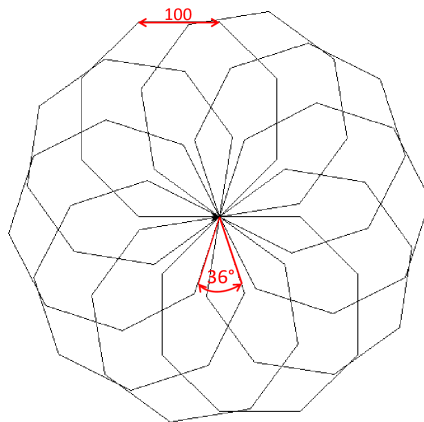


Figure 1: Flower with polygons

Problem 2: Valid Age [15 marks]

Problem statement:

程序代写代做 CS编程辅导

After learning repetition statements, we can do some simple data validation.

Suppose a program is to read in (via the function `input()`) an integer value representing the age of a person. It makes sense that the value should be in a reasonable range, say 1 to 100, both inclusive. If the user enters an invalid input outside this range, the program should ask for the age again.

Write a function `check_age` that takes an integer value as input and also to count and print out how many times the user has entered the input. The function should return the number of attempts. The input and output can be seen from the sample runs below.

Sample run #1:



```
Enter age: 17
Your age is 17
Number of attempts = 1
```

WeChat: cstutorcs

Sample run #2:

```
Enter age: -5
Enter age: 101
Enter age: 32
Your age is 32
Number of attempts = 3
```

Assignment Project Exam Help

Email: tutorcs@163.com

Problem 3: Who are the Winners? [20 marks]

QQ: 749389476

Note: In this problem, you should not need any “`print()`”.

Problem statement:

<https://tutorcs.com>

Citizens of Zakadaha hold an annual Gagalafa festival to celebrate the harvest of their prized produce, the well-sought after cocoa beans Kokomoko. A lucky draw is held during the festival. Every participant is given a lucky draw number. Each year, the organizer decides on two non-zero digits, the factor-digit and the must-have-digit. These two digits may be the same.

A winning lucky draw number is a number that is a multiple of *factor-digit* and also contains the *must-have-digit*.

In this exercise, you are to write a function `find_winners(f, m, n)` to read in the following three inputs:

1. The factor-digit, `f`, which is a non-zero digit (1 – 9).
2. The must-have-digit, `m`, which is also a non-zero digit (1 – 9).
3. The number of participants, `n`. Lucky draw numbers will be numbered from 1 to `n` inclusively.

You may assume that all inputs are valid.

For example, if factor-digit is 3, must-have-digit is 5, and the number of participants is 100, then the number of winners is 6 (the winning numbers are 15, 45, 51, 54, 57 and 75).

Your function is to count and return the number of winners whose lucky draw number is a multiple of factor-digit as well as contains the must-have-digit.

Sample runs:

```
>>> print(find_winners(3,5,100))
6
>>> print(find_winners(9,1,15))
0
>>> print(find_winners(7,7,200))
5
```



In order to completely solve this problem, we need to divide this problem into two parts.

Part 1: Check if a Number is a Winning Number

Before we rush into counting how many winning numbers there are, how do we tell if a person wins with his/her number x ? Or, how do we tell if a number x is a winning number that...

- Is divisible by the factor-digit. Let's call this f . And...
- Contains the must-have digit. Let's call this m .

Given a number x :

- How do you know if x is divisible by f ?
 - This should be easy to solve
- How do you know if x contains a digit m ?
 - This could be a bit harder
 - If we write a function `number_contains(x,m)`, a sample output for the function should look like this:

```
>>> number_contains(123456,3)
True
>>> number_contains(123456,9)
False
>>> number_contains(555555,5)
True
```

Part 2 Count How Many Numbers are "Winning" between 1 to n .

This should be easier than Part 1.