QUESTION 1 (3 MARKS)
A regular dodecahedron is a 3-dimensional shape that has 1 key dimension:

1. The edge length – $a$

An illustration of a regular dodecahedron is shown below:

The formula to calculate the surface area of a regular dodecahedron is:

$$\text{Surface Area} = 3\sqrt{25 + 10\sqrt{5}} \, a^2$$

The formula to calculate the volume of a dodecahedron is:

$$\text{Volume} = \frac{15 + 7\sqrt{5}}{4} \, a^3$$
Write a program that calculates the surface area and volume of a regular dodecahedron. Name your program file ‘YourUserNameA1Q1.py’, e.g. dazh001A1Q1.py

Your program will need to:

- Display a banner.
- Ask the user to enter the value for the edge length. You can assume that the user will always enter integer values.
- Calculate the surface area and volume of the dodecahedron using the value for the edge length entered by the user.
- Display these calculated values rounded to 3 decimal places.

Two example outputs using the completed program are shown below. The user input is shown in red font. Your program must give the output in the same format as the outputs in the two examples below.

```
# Regular Dodecahedron Surface Area and Volume Calculator

Edge Length: 25
Surface area: 12903.581
Volume: 119736.234

# Regular Dodecahedron Surface Area and Volume Calculator

Edge Length: 37
Surface area: 28264.003
Volume: 388159.965
```

**QUESTION 2 (4 MARKS)**

Write a program which generates a password that is 8 characters long. Name your program ‘YourUsernameA1Q2.py’, e.g. dazh001A1Q2.py. Your program must use the following variable initialized to the value specified below:

```
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
```

The password you generate will consist of 4 alphabetical characters followed by 4 numerical characters. Each of the 4 alphabetical characters will be selected from the string assigned to the variable alphabet at random.

In terms of the numerical characters, you will need to generate a random number between 1 and 999 inclusive. You will then zero extend the number you generate so that it is 4 characters long. For example, a single digit number like 1 would result in a numerical component of "0001". A two digit random number like 56 would result in a numerical component of "0056". A three digit random number like 989 would result in a numerical component of "0989".

Three example outputs using the completed program follow. Your program must give the output in the same format as the outputs in these three examples.
QUESTION 3 (4 MARKS)
Write a program which solves the change-making problem for a particular currency. Name your program ‘YourUserNameA1Q3.py’, e.g. dazh001A1Q3.py

The change-making problem involves selecting the fewest number of coins from a given set of coin denominations to represent a given value. The currency in question has the following coin denominations: 1, 5, 10, 25, 50 and 100. For example, given this currency and a value of 76 cents, the solution to the change making problem would be to select one 50 cent coin, one 25 cent coin, and one 1 cent coin.

The value to represent will be entered by the user. You can assume that the user will always enter a positive integer value. Your program must give the output in the same format as the outputs in the following three examples. The user input is shown in red font. The lines of “=” symbols contain 19 characters.

<table>
<thead>
<tr>
<th>Value: 76</th>
<th>Value: 254</th>
<th>Value: 83</th>
</tr>
</thead>
<tbody>
<tr>
<td>You will need:</td>
<td>You will need:</td>
<td>You will need:</td>
</tr>
<tr>
<td>100s: 0</td>
<td>100s: 2</td>
<td>100s: 0</td>
</tr>
<tr>
<td>50s: 1</td>
<td>50s: 1</td>
<td>50s: 1</td>
</tr>
<tr>
<td>25s: 1</td>
<td>25s: 0</td>
<td>25s: 1</td>
</tr>
<tr>
<td>10s: 0</td>
<td>10s: 0</td>
<td>10s: 0</td>
</tr>
<tr>
<td>5s: 0</td>
<td>5s: 0</td>
<td>5s: 1</td>
</tr>
<tr>
<td>1s: 1</td>
<td>1s: 4</td>
<td>1s: 3</td>
</tr>
</tbody>
</table>

QUESTION 4 (4 MARKS)
Write a program which implements a dice game. Name your program file ‘YourUserNameA1Q4.py’, e.g. dazh001A1Q4.py.

The aim of the game is to reach a score as close as possible to 100 in three rounds. Each round consists of throwing five random dice. The user then chooses two of the dice values, where the two dice values chosen form a two digit score which is added to the user’s current total. For example, if the user first chooses a dice
with the value 3 and then a dice with the value 5, 35 is added to the user’s total. The first dice chosen is therefore the tens digit and the second dice chosen is the units digit.

The five random dice are displayed with one space between each dice:

```
Your dice: 3 5 3 4 1
```

To choose the dice the user enters a number 1, 2, 3, 4, 5 indicating which of the five dice they wish to choose (i.e. the position of the dice and **not** the value of the dice). This process is repeated three times. At the end of the program you will need to inform the user of their score and how far off they are from the goal of 100.

When writing your program, you **must** use the following statement to initialize the user’s current score:

```
current_score = 0
```

Below are two example outputs using the completed program. The user input is shown in red font. You can assume the user will always enter valid input (i.e. an integer between 1 and 5). Your program **must** give the output in the **same format** as the outputs in the following two examples. The top string of “*” symbols has a length of 45 and the bottom string of “*” symbols has a length of 29.

```
*********************************************
REACH 100 IN THREE ROUNDS!   Initial total: 0
*********************************************

Round 1:
Your dice: 2 4 1 1 2
  Tens? 2
  Units? 3
Dice value: 41
Your current total: 41

Round 2:
Your dice: 6 1 2 1 6
  Tens? 3
  Units? 5
Dice value: 26
Your current total: 67

Round 3:
Your dice: 5 6 3 2 1
  Tens? 3
  Units? 4
Dice value: 32

*********************************************
Your final score: 99
You are 1 away from the goal
*********************************************
```
REACH 100 IN THREE ROUNDS! Initial total: 0

Round 1:
Your dice: 5 2 1 6 2
  Tens? 1
  Units? 4
Dice value: 56
Your current total: 56

Round 2:
Your dice: 3 1 4 5 4
  Tens? 2
  Units? 3
Dice value: 14
Your current total: 70

Round 3:
Your dice: 5 6 1 4 1
  Tens? 4
  Units? 5
Dice value: 41

Your final score: 111
You are 11 away from the goal