P52

Pre-Leaving Certificate Examination, 2022

Computer Science

Section C

Higher Level

Time: 1 hour

80 marks

Instructions

There is one section of the examination paper in this booklet.

Section C	Programming	One question Answer all questions	80 marks

Answer all parts of the question on your digital device.

Calculators may be used during this section of the examination.

Ensure that you save your work regularly and when you complete each question part.

Save your files using the naming structure described at the beginning of each question part.

If you are unable to get some code to work correctly, you can comment out the code so that you can proceed. The code that has been commented out will be reviewed by the examiner.

Rough work pages are provided at the end of this booklet. Please note that this work will **not** be reviewed by an examiner.

At the end of the examination it is your responiibilty to ensure that you have saved all of your files onto your external media.

Section C Programming 80 marks

Answer all question parts.

Question 16

We can use the random library when building computer programs to build basic games. Many computer games use random features to keep gameplay fresh and exciting.

(a) Open the program called **Question16_A.py** from your device.
Enter your name on **line 2**.

```
1  # Question 16(a)
2  # Student name:
3  import random
4  your_name = input("Please enter your name: ")
5  lucky_number = 5
6  computer_die_roll = random.randint(1,6)
7  print("The computer rolled: ", computer_die_roll)
```

This program is designed to roll a six-sided dice and display the result, the user will then take a guess at what number the computer will roll. The objective of the game is for the user to correctly guess what number the computer will roll on its dice. The user has a variable called `lucky_number` whose value is 5.
A sample run of the program is displayed below – the user enters their name and the computer's dice roll is displayed.

```
Please enter your name: John
The computer rolled: 3
```

Modify the program to do the following:

(i) Insert a comment to say "initialize computer number" in the appropriate location in the program to show where the computer generates it's dice score.

(ii) Currently in the program the value of the variable **`lucky_number`** is hard-coded to 5. Modify the program so that it prompts the user to enter a value for **`lucky_number`**. The value should be converted to an integer.

When the program is run the output may look as follows:

```
Please enter your name: John
Please select a lucky number between 1 and 6: 4
The computer rolled: 1
```

(iii) Modify the program to display the user's lucky number before the computer's dice roll value. When the program is run, the output may look as follows:

```
Please enter your name: John
Please select a lucky number between 1 and 6: 4
John's lucky number was: 4
The computer rolled: 3
```

(iv) Incorporate the following function definition into your program and insert a line so that the function is called before the user enters any data. All your code should be incorporated into this **`dice_game()`** function.

```
def dice_game():
  print ("welcome to my dice game!!")
```

(v) As the program is now, it doesn't play a game between the user and the computer. A more meaningful output would be to compare the **`lucky_number`** of the user and the dice roll of the computer and see if the user guessed the correct number. Extend the program so that it displays if the user guessed the correct number, guessed too low or too high.

When the program is run the output may look as follows:

```
Welcome to my dice game!!
Please enter your name: John
Please select a lucky number between 1 and 6: 5
John's lucky number was: 5
The computer rolled: 2
You guessed too high!
```

```
Welcome to my dice game!!
Please enter your name: John
Please select a lucky number between 1 and 6: 2
John's lucky number was: 2
The computer rolled: 6
You guessed too low!
```

```
Welcome to my dice game!!
Please enter your name: John
Please select a lucky number between 1 and 6: 4
John's lucky number was: 4
The computer rolled: 4
You guessed correct, well done!
```

Save and close your file before moving on to the next part.

(b) Open the program called **Question16_B.py** from your device.
Another type of game that can be created using the random library in Python is a basic lotto game.

This program is designed to simulate a basic lottery game although some features are missing; the user picks 3 numbers to put on their ticket. The computer then randomly picks 3 numbers from a list of 10 numbers and displays the result.
A sample run of the program is displayed below – the user picks their numbers, and the computer takes random numbers from the drum list. Examine the code carefully and answer the questions below.

```
1   #Question 16 (b)
2   #Student name:
3   import random
4   ticket = []
5
6   user_number = int(input ("Please pick a number between 1 and 10: "))
7   ticket.append(user_number)
8   user_number = int(input("Please pick a number between 1 and 10: "))
9   ticket.append(user_number)
10  user_number = int(input ("Please pick a number between 1 and 10: ))
11  ticket.append (user_number)
12
13  print ("Your ticket is: ", ticket)
14  print ("The draw will start now, good luck!")
15  drum = [1,2,3,4,5,6,7,8,9,10]
16  result = []
17  def lotto (ticket):
18          for times in range (3):
19            draw = drum [random.randint (0,len (drum))-1]
20            result.append (draw)
21       print("The draw was: ", result)
22
23  lotto (ticket)
```

```
Please pick a number between 1 and 10: 4
Please pick a number between 1 and 10: 7
Please pick a number between 1 and 10: 2
Your ticket is: [4, 7, 2]
The draw will start now, good luck!
The draw was: [10, 10, 7]
```

(i) Modify the code so the line asking the user for their input is only used once but repeated 3 times to allow the user to enter their 3 numbers.

(ii) Using one of pythons inbuilt methods, functios or otherwise, modify the code above so that it will check if the ticket matches the result of the lotto (i.e., that the 3 numbers in each list are the same).
Note: The order that the user picks their numbers and the order in which numbers are drawn out of the drum does not matter, if they are the same, the user will win.
An appropriate message should be displayed in either scenario as below:

```
Please pick a number between 1 and 10: 2
Please pick a number between 1 and 10: 5
Please pick a number between 1 and 10: 3
Your ticket is: [2, 5, 3]
The draw will start now, good luck!
The draw was: [7, 9, 10]
You lose!
```

```
Please pick a number between 1 and 10: 4
Please pick a number between 1 and 10: 2
Please pick a number between 1 and 10: 7
Your ticket is: [4, 2, 7]
The draw will start now, good luck!
The draw was: [7, 4, 2]
You win!
```

(iii) Currently, it is possible for the computer to draw duplicates from the drum as shown below:

```
Please pick a number between 1 and 10: 4
Please pick a number between 1 and 10: 7
Please pick a number between 1 and 10: 2
Your ticket is: [4, 7, 2]
The draw will start now, good luck!
The draw was: [10, 10, 7]
```

This is not realistic as lotto games usually have only one copy of each number in the drum. Using one of pythons inbuilt methods, functios or otherwise, modify the code so that each number picked from the drum is unique for the draw list.

Save your file.

Ensure that you have saved and closed all files before you finish the examination.

Space for rough work.

This page will not be reviewed by an examiner.

Space for rough work.

This page will not be reviewed by an examiner.

Space for rough work.

This page will not be reviewed by an examiner.

Blank Page

Blank Page

Blank Page