2020. M110AB

2020L219A1EL

Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate Examination 2020

Computer Science

Sections A & B

Higher Level

1 hour 30 minutes

130 marks

Examination number					

Centre stamp

For Examiner use only	
Section	**Mark**
A	
B	
C	
Total	

Instructions

There are **three** sections in this examination. Section A and B appear in this booklet. Section C is in a separate booklet that will be provided for the computer-based element.

Section A	Short Answer Questions	60 marks	12 questions
Section B	Long Questions	70 marks	3 questions
Section C	Programming	80 marks	1 question

Answer all questions.

Calculators may **not** be used during this section of the examination.

The superintendent will give you a copy of page 78 (Logic Gates) of the *Formulae and Tables* booklet on request. You are not allowed to bring your own copy into the examination.

Write your answers for Section A and Section B in the spaces provided in this booklet. There is space for extra work at the end of the booklet. Label any such extra work clearly with the question number and part.

Section A Short Answer Questions 60 marks

Answer all twelve questions.

Question 1

Given the following JavaScript variable declarations, state the result of each of the JavaScript expressions in the table below.

```
let a = 5;
let b = 2;
```

Expression	Result
`a * b`	
`a ** b`	
`a / b`	
`b % a`	
`++a`	

Question 2

Many modern laptops have a hard-disk drive (HDD) which can exceed 1TB in capacity. If you are buying a laptop you may also have the option of a solid-state drive (SSD) with a capacity of 512GB.

(a) In terms of storage capacity, what do the letters GB and TB stand for?

GB:
TB:

(b) Assuming that neither cost nor capacity were issues, explain why you might opt for the SSD rather than the HDD.

Question 3

The program counter is a special register in the processor of a computer.
Outline the purpose of the program counter.

Question 4

The World Wide Web (WWW) and the network infrastructure that supports it allow for seamless transmission of data such as web pages, voice applications and streaming services. This is all possible due to the protocols that it uses.

(a) When referring to World Wide Web infrastructure, what is meant by the term *protocol*?

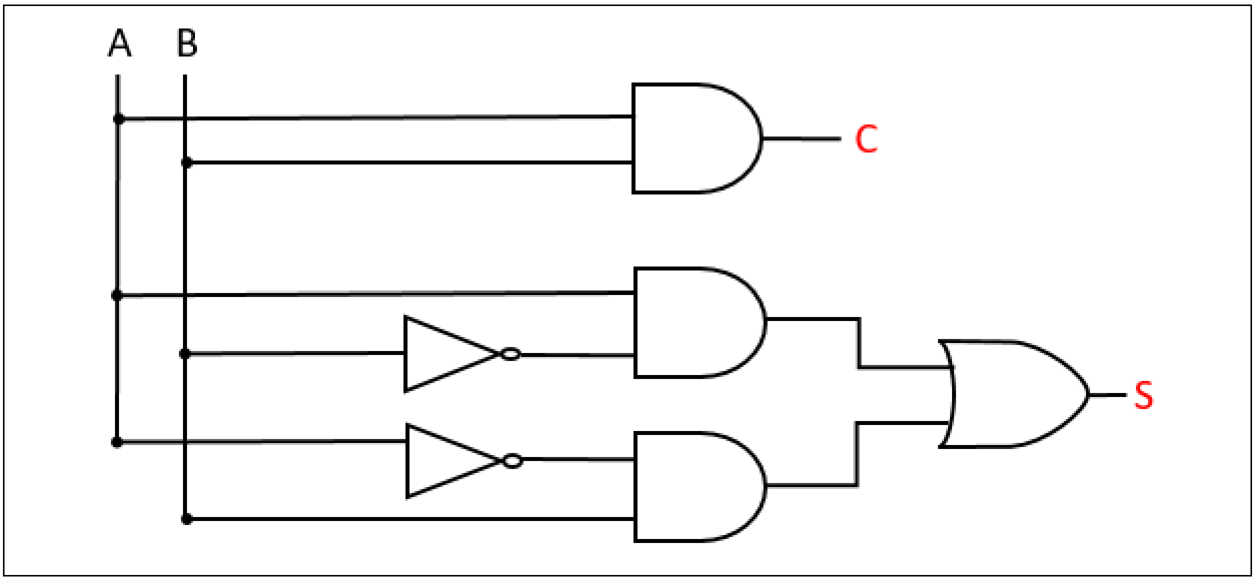
(b) State the name and purpose of **one** such protocol.

Name:

Purpose:

Question 5

The half-adder logic circuit shown below generates two outputs, S and C, from two inputs, A and B.



(a) What is the value of **C** when the inputs A and B are both 0?

(b) What is the value of **S** when the inputs A and B are both 1?

Question 6

The American Standard Code for Information Interchange (ASCII) is a character encoding standard adopted by the Institute of Electrical and Electronics Engineers (IEEE) in 1963.

(a) Why are encoding standards such as ASCII important?

(b) State **one** limitation of ASCII.

Question 7

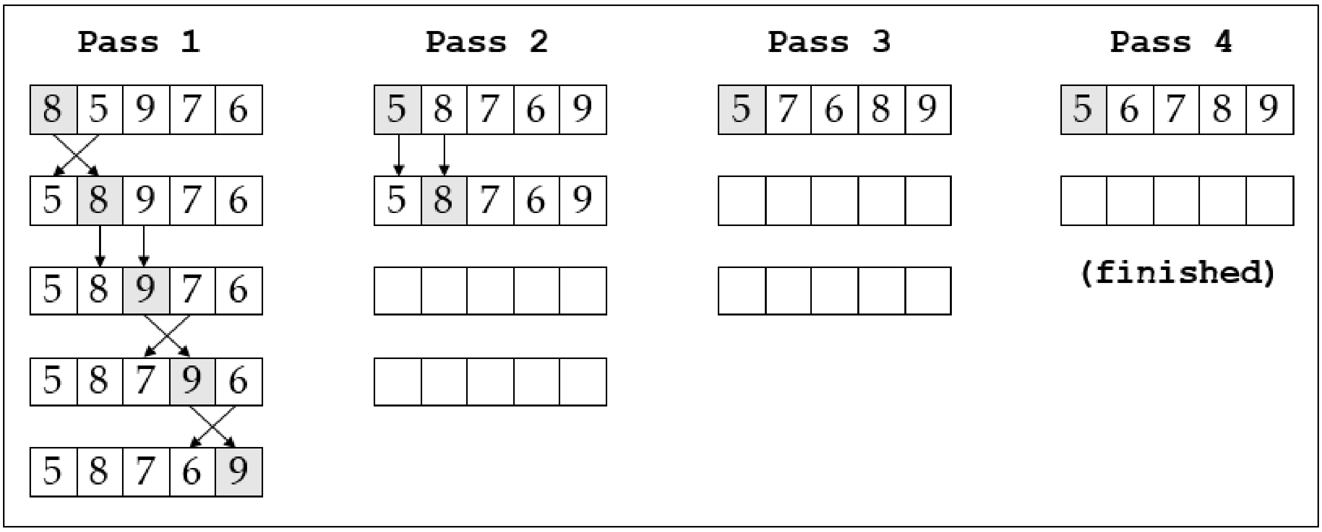
The 8-bit binary representation for the ASCII character *K* is shown below.
Convert this binary number to hexadecimal notation.

```
0100 1011
```

Question 8

The diagram below sets out the operation of the bubble sort algorithm to sort the list of integers [8, 5, 9, 7, 6]. The algorithm works by scanning over the data in four passes. The diagram is complete for pass 1 and started for pass 2.

Complete the diagram for passes 2, 3 and 4. You only need to fill in the numbers.



Question 9

The data set below shows the raw data collected from the result of a 100m school race.

Surname	Gender	Age	Time
Murphy	M	17	13,12
Ogene	M	16	12.14
Ogene	M	16	12.14
Mc Intyre	F.	17	12.87
Lopez	F	-18	14.01
	F	17	1 329
McCarthy	M	77	13.65
Ó Brádaigh	f	16	13.09

List **three** problems with the data in the data set.

1.

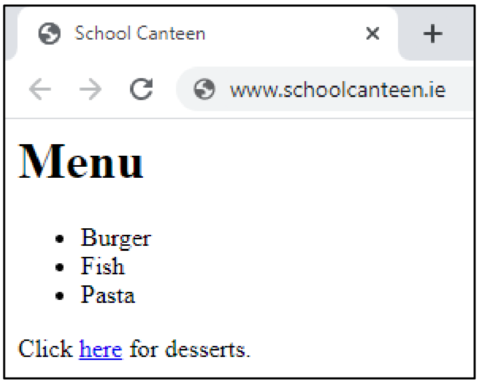
2.

3.

Question 10

The illustrations below show HTML code and the resulting web page as it would be displayed in a typical web browser. Some of the code has been replaced with the numbers 1 – 4.

```
 < ❶ >
  <head>
    <title>School Canteen</title>
  </head>
  <body>
    <h1> ❷ </h1>
    < ❸ >
      <li>Burger</li>
      <li>Fish</li>
      <li>Pasta</li>
    </ ❸ >
    <p>Click <a ❹ ="desserts.html">here</a> for desserts.</p>
  </body>
</html>
```



School Canteen ×

www.schoolcanteen.ie

Menu

- Burger
- Fish
- Pasta

Click here for desserts.

Complete the table below with the missing code.

Number	Missing Code
❶	
❷	
❸	
❹	

Question 11

The intention of the JavaScript function below is to return a student grade descriptor based on a percentage grade as shown in the table to the right.

Percentage Grade	Descriptor
80 or over	Distinction
From 40 to 79	Pass
Less than 40	Unsuccessful

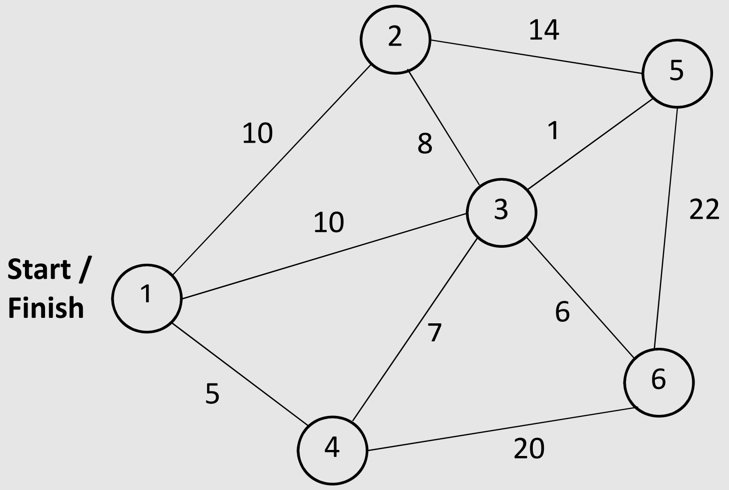
```javascript
function getGradeDescription(percentageGrade) {
    let gradeDescription = "Unsuccessful";

    if (percentageGrade >= 80)
        gradeDescription = "Distinction";

    if (percentageGrade >= 40)
        gradeDescription = "Pass";

    return gradeDescription;

 }
```

The code runs without any syntax errors but it does not always return the correct grade descriptor. Outline **one** way in which the function could be modified so that it works as intended.

Question 12

The travelling salesperson problem is commonly used in the study of algorithms and appears in formats similar to the problem below.

> "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city once and returns to the original city?"

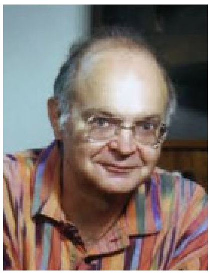


Why would *heuristics* be considered a good approach to solving a problem such as this?

Section B Long Questions 70 marks

Answer all three questions.

Question 13



In his book *The Art of Computer Programming*, Donald Knuth states that “searching is the most time-consuming part of many programs, and the substitution of a good search method for a bad one often leads to a substantial increase in speed.”

(a) The Python code below shows an implementation of a search algorithm. Examine the code and answer the questions that follow:

```python
names = ["John", "Mary", "Zoe", "Alex", "Séamas"]
name = input("Enter lookup name: ")

found = False
index = 0

while (not found) and (index != len(names)):
    if name == names[index]:
        found = True
    else:
        index = index + 1

print("Result:", index)
```

(i) State the name of the above search algorithm.

(ii) What is the data type of the variable called **`found`**?

(iii) Step through how the algorithm finds the name *Zoe* in the list called **`names`**.

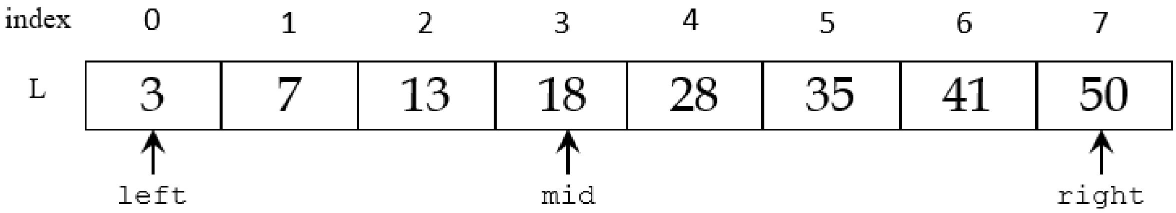
(iv) What would be the value of **`index`** after running the algorithm if the user entered a name that was not present in the list called **`names`**?

(v) What is the worst-case time complexity of this search algorithm? Explain your answer.

(b) Binary search is generally regarded as a highly efficient search algorithm.

(i) State **one** disadvantage of the binary search algorithm.

(ii) Step through how the binary search algorithm finds the integer 28 in the list **`L`** shown below.



index	0	1	2	3	4	5	6	7
L	3	7	13	18	28	35	41	50

(iii) What is the maximum number of comparisons the binary search algorithm would need to make if the value to be searched for did **not** exist in a list of 8 items?

Answer:

Space for rough work:

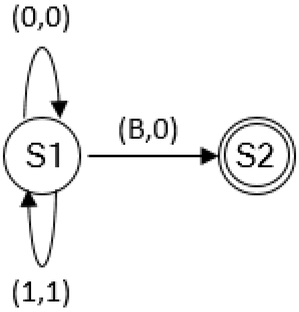
(c) Why is the study of algorithmic efficiency considered to be important in computer science?

Question 14

Alan Turing is widely regarded as one of the founders of computer science because of his work on the development of the Turing Machine and the Turing Test.

(a) Explain the importance of Turing Machines as a computational tool.

(b) The illustration below depicts a Finite State Machine which, for a particular Turing Machine, defines two states, S1 and S2, and three transitions.
Study the illustration and answer the questions that follow.



B = blank cell

(i) What is the significance of state S2?

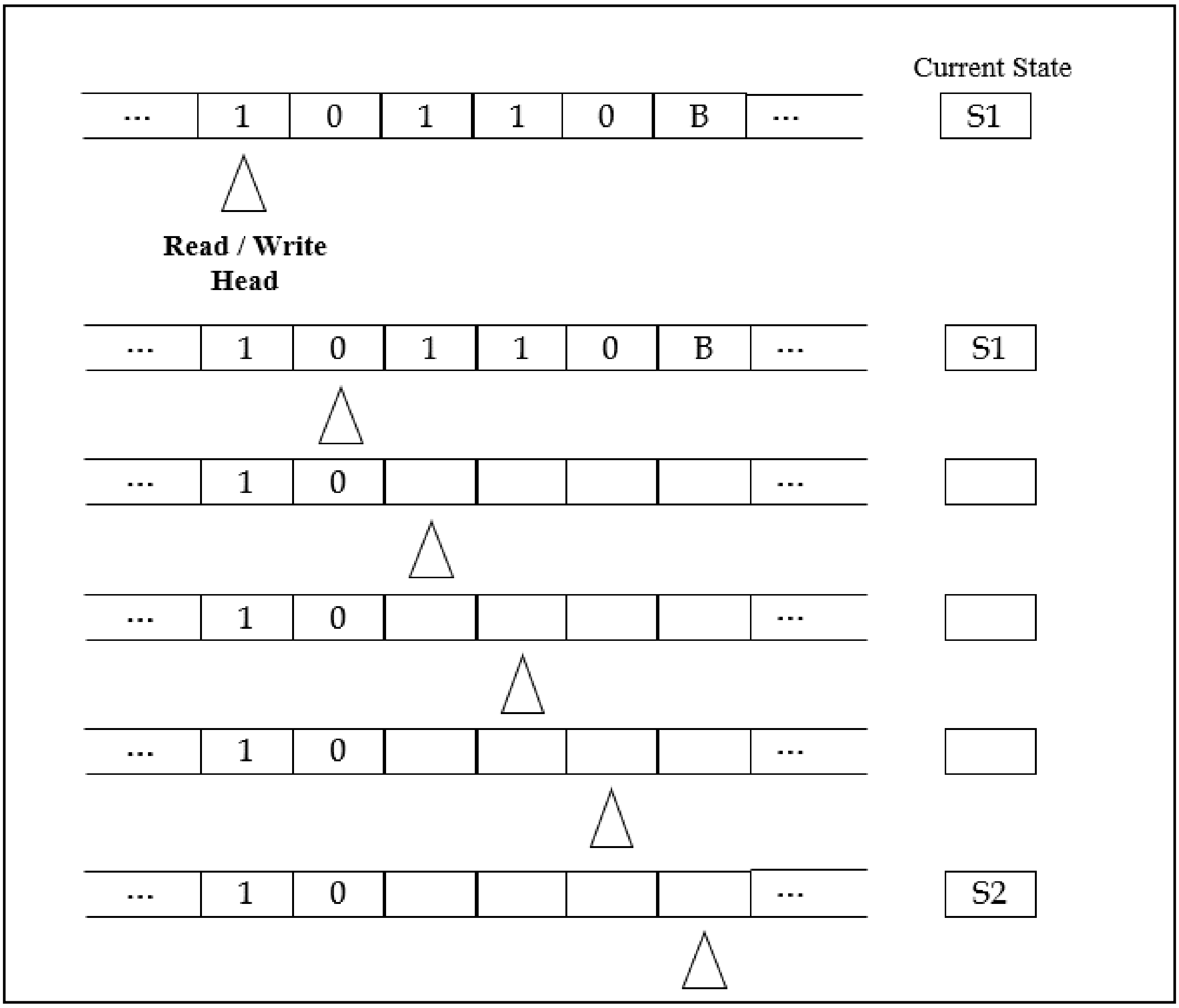
(ii) Complete the state transition table below based on the above Finite State Machine.
The first row has already been completed.

Current State	Input	Output	Next State
S1	0	0	S1
S1			
S1			

(iii) Starting from an initial state of S1 and an input of 10110B, as shown on the tape diagram below, show how the Turing Machine produces an output of 101100.

In your answer you should trace the computation clearly by showing the contents of the tape at the end of each state transition. You should complete the tape and the current state for each row. You can assume that the read/write head is moved one place to the right at the end of each step.

The first two rows have been completed for you.



(c) In the mid-20th century, Turing carried out what is now generally regarded as the earliest substantial work in the field of Artificial Intelligence (AI). There is a lot of controversy and discussion about the area of AI in recent times.

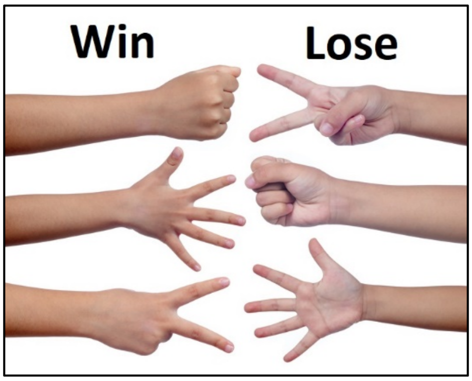
Bill Gates stated that “humans should be worried about the threat posed by artificial intelligence”.

Discuss the quotation above, focusing on the potential positive and negative effects of AI on society.

Question 15

Rock Paper Scissors is a popular two player game in which each player simultaneously picks one of three objects – rock, paper or scissors. The rules to determine the winner are relatively straightforward:

- Rock beats Scissors
- Paper beats Rock
- Scissors beats Paper



If both players choose the same object, the result is a draw (tie).

You have been asked to design and develop an interactive computerised *Rock Paper Scissors* game.

(a) State whether you would use a staged or iterative development process for the project. Justify your answer by giving **two** reasons for your decision.

Staged or Iterative:

Reason 1:

Reason 2:

(b) As project manager one of your main tasks will be to form a project team and assign roles and responsibilities to the team members.

Identify **two** roles that would be appropriate for this project and for each role outline **two** responsibilities associated with that role.

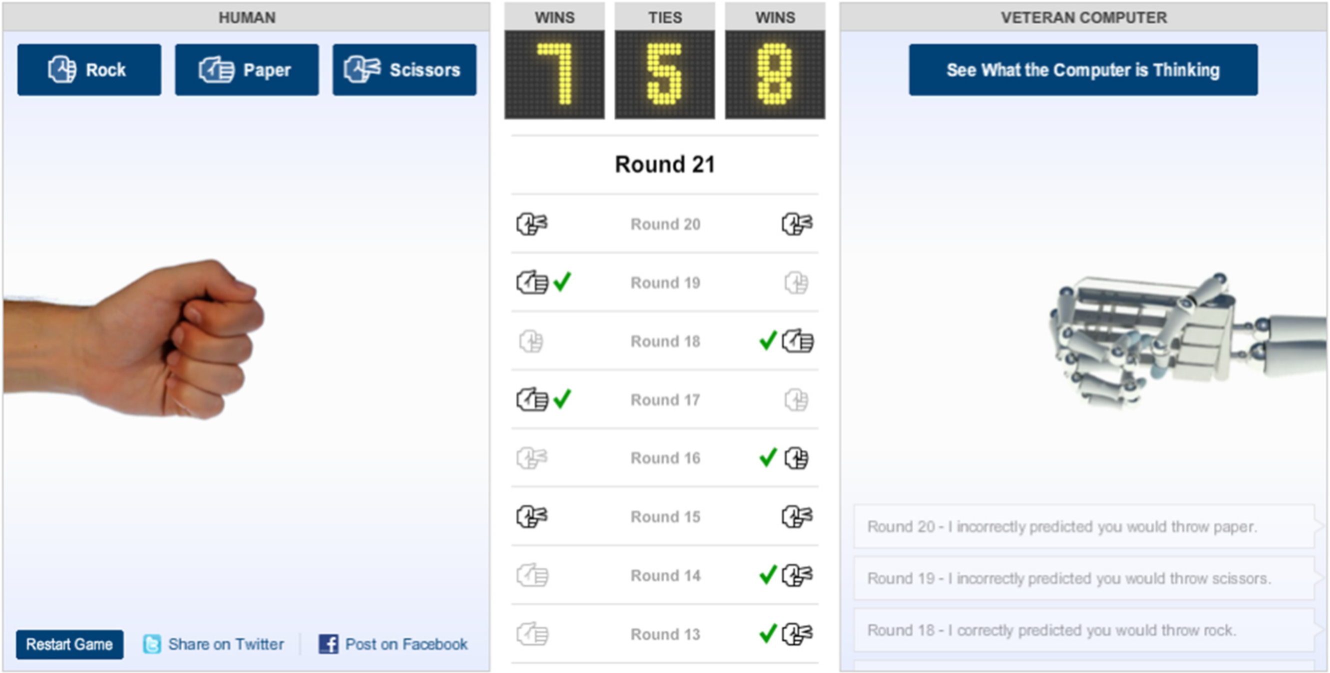
Role 1:
Responsibility 1:
Responsibility 2:

Role 2:
Responsibility 1:
Responsibility 2:

(c) Your project team has identified two user interfaces that they observed in other *Rock Paper Scissors* games. Both of these interfaces are shown on the next page.

(i) Compare the two interfaces in terms of the types of applications and the functionality they offer.

Rock Paper Scissors – User Interface 1

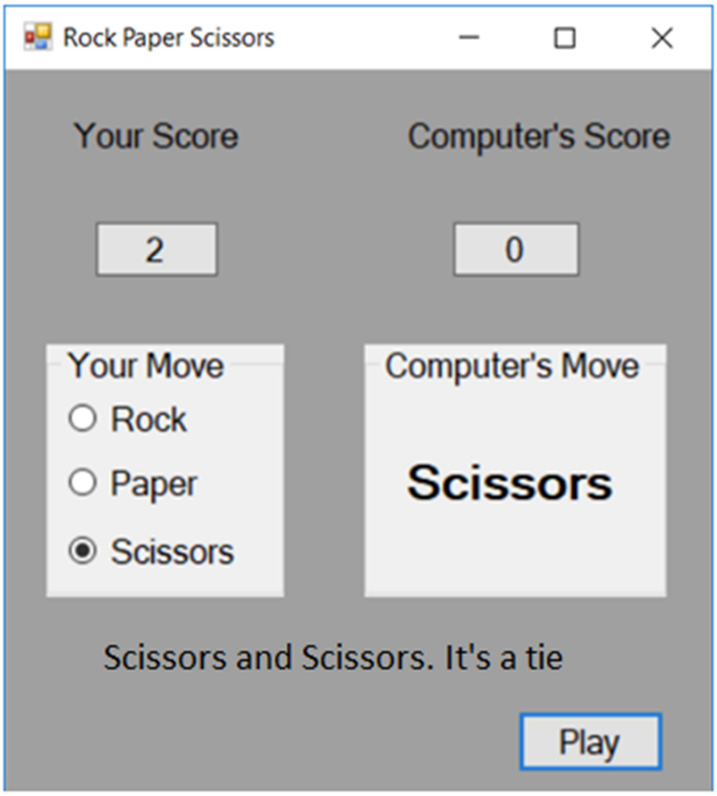


Round 20 - I incorrectly predicted you would throw paper.

Round 19 - I incorrectly predicted you would throw scissors.

Round 18 - I correctly predicted you would throw rock.

Rock Paper Scissors – User Interface 2



Scissors and Scissors. It's a tie

(ii) Provide **two** examples of how adaptive technology could be incorporated into a game of *Rock Paper Scissors*.

1.

2.

Space for extra work.

Indicate clearly the number and part of the question(s) you are answering.

Space for extra work.

Indicate clearly the number and part of the question(s) you are answering.

Space for extra work.

Indicate clearly the number and part of the question(s) you are answering.

Acknowledgements

Images

Image on page 11: www.computerhistory.org/fellowawards/hall/donald-knuth/
Image on page 16: www.britannica.com/biography/Alan-Turing
Image on page 20: www.portablepress.com/blog/2017/11/how-to-win-at-rock-paper-scissors/
Image 1 on page 23: archive.nytimes.com/www.nytimes.com/interactive/science/rock-paper-scissors.html
Image 2 on page 23: www.chegg.com/homework-help/questions-and-answers/rock-paper-scissors-game-application-write-c-program-allows-one-user-play-rock-paper-sciss-q32554392

Texts

Quote on page 11: Donald E. Knuth, The Art of Computer Programming (Vol III Searching and Sorting, 2nd ed., Pgs. 392-393)

Leaving Certificate – Higher Level

Computer Science – Sections A & B

1 hour 30 minutes

2020. M110C

2020L219A2EL

Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate Examination 2020

Computer Science

Section C

Higher Level

1 hour

80 marks

Instructions

There is one section in this paper.

Section C	Programming	80 marks	1 question

Answer all parts of the question on your digital device.

Calculators may be used during this section of the examination.

The *Formulae and Tables* booklet cannot be used for this section of the examination.

The Superintendent will give you a copy of the *Python Reference Guide*.

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 booklet is not to be handed up and will **not** be reviewed by an examiner.

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

You will be provided with a brown envelope for your external media. Write your examination number on this envelope and place your external media into it before sealing. Place this envelope in the pouch at the front of the red envelope that contains your examination booklet from Section A and B.

Do not hand this paper up

Section C Programming 80 marks

Answer all question parts.

Question 16

A password strength meter is a mechanism that can be used to safeguard against setting weak passwords. When a user is creating a password for the first time or changing an existing password, a password strength meter can be used to show how resistant the password is to attack.

Meters have rules they use to assign points for password strengthening measures such as including combinations of uppercase and lowercase letters as well as numbers and special symbols.

(a) Open the program called **Question16_A.py** from your device. The source code is shown on the next page and described briefly below.

Before making any changes, you should use the format **CandidateNumberQuestion16_A.py** to save your file. For example, if your candidate number was 123456 you would save the file as **123456Question16_A.py**.

Enter your Examination Number in the space provided on **line 2.**

This program is designed to calculate and display a score that indicates the strength of a password entered by the user.

The variable `score` is used to store the password strength. This variable is initially set to zero and additional points are added based on the following rules:

1. The password contains more than seven characters: +5 points
2. The password contains at least one lowercase letter: +1 point
3. The password contains a mix of lowercase and uppercase letters. +5 points

A sample run of the program is shown below:

```
Enter a password: sunshine
6
```

Here the user enters the password *sunshine* and the program calculates and displays a score of 6. This is because the password contains more than seven characters (5 points) and contains lowercase letters (1 point).

```python
# Question 16(a)
# Examination Number:

# Prompt the user to enter a password and store the ...
# value entered in the variable password
password = input("Enter a password: ")

# A variable to store all the lowercase letters in the alphabet
LOWER_CASE_LETTERS = "abcdefghijklmnopqrstuvwxyz"

# The variables lowercase and uppercase indicate the presence or ...
# absence of lowercase and uppercase characters in the password
lowercase = False # True if password contains a lowercase letter
uppercase = False # True if password contains an uppercase letter

# Loop through each character in the password and ...
# check the password for specific characters
for character in password:
  if character in LOWER_CASE_LETTERS:
    lowercase = True
  if character in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
    uppercase = True

# Calculate the score based on the rules

score = 0

# Rule 1
if len(password) > 7:
    score = score + 5

# Rule 2
if lowercase:
    score = score + 1

# Rule 3
if lowercase and uppercase:
    score = score + 5

# Display the score
print(score)

```

Make the following changes to the program:

(i) Insert a comment to say *'initialise score'* in an appropriate location in the program.

(ii) Amend the program so that is displays two lines of output as follows:

- the first line will display the word *Password:* followed by the password that was entered by the user, and
- the second line will display the word *Score:* followed by the calculated score for that password.

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

```
Enter a password: sunshine
Password: sunshine
Score: 6
```

(iii) Currently in the program, the uppercase letters are hard-coded as the string: `"ABCDEFGHIJKLMNOPQRSTUVWXYZ"`

Replace the use of this string with a variable, in a manner similar to that used to represent the lowercase letters. The output of the program should not be changed. You should name the variable **`UPPER_CASE_LETTERS`**.

(iv) Implement a new rule (rule 4) so that the score is increased by 2 points if the password contains at least one uppercase letter.

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

```
Enter a password: Sunshine
Password: Sunshine
Score: 13
```

(v) Implement a new rule (rule 5) so that the score is increased by 5 points if the password contains at least one digit (any integer in the range 0 to 9: +5 points).

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

```
Enter a password: 3Sunshine
Password: 3Sunshine
Score: 18
```

(vi) Implement a new rule (rule 6) so that the score is increased by:

- 1 point if the first character of the password is a digit
- 1 point if the last character of the password is a digit
- 2 extra points if both the first and the last characters of the password are digits

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

```
Enter a password: 3Sunshine7
Password: 3Sunshine7
Score: 22
```

(vii) Implement a new rule (rule 7) so that the score is reduced by 10 points if the password contains only digits.

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

```
Enter a password: 1234
Password: 1234
Score: -1
```

(viii) Change rule 1 so that the score is adjusted according to the password lengths as shown in the following table.

Password Length	**Score**
Greater than 7 characters	+5 points
From 4 to 7 characters	+2 points
Less than 4 characters	−2 points

The table below shows the scores that would be awarded for a variety of passwords. You could use this information to test your program.

Password	**Score**
sun	−1
Sun	6
sun2	9
2sun3	12
3Sunshine	19
3Sunshine7	22

Use the format **CandidateNumberQuestion16_A.py** to save your file. For example, if your candidate number was 123456 you would save the file as **123456Question16_A.py**.

(b) Open the program called **Question16_B.py** from your device. The source code is shown on the next page and described briefly below.

Before making any changes, you should use the format **CandidateNumberQuestion16_B.py** to save your file. For example, if your candidate number was 123456 you would save the file as **123456Question16_B.py**.

Enter your Examination Number in the space provided on l**ine 2.**

This program is very similar to that provided for part (a) with two main differences:

– The code to calculate the password score is contained in a function definition called **`calculate_score`**. This function accepts a parameter called **`password`** and returns the calculated score.
– Instead of prompting the user to enter a single password this program, uses a list of hard-coded passwords called **`test_passwords`**.

When the program is run it loops through each password in the list **`test_passwords`**. As it does so, it calculates and displays the score of each password.

A sample run of the program is shown below:

```
1
6
11
0
5
```

```python
# Question 16(b)
# Examination Number:

# A variable to store all the lower case letters in the alphabet
LOWER_CASE_LETTERS = "abcdefghijklmnopqrstuvwxyz"

def calculate_score(password):

    # The variables lowercase and uppercase indicate the presence or
    # absence of lowercase and uppercase characters in the password
    lowercase = False #True if password contains a lowercase letter
    uppercase = False #True if password contains an uppercase letter

    # Loop through each character in the password and ...
    # ... check the password for specific characters
    for character in password:
      if character in LOWER_CASE_LETTERS:
        lowercase = True
      if character in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
        uppercase = True

    # Calculate the score based on the rules

    score = 0

    # Rule 1
    if len(password) > 7:
        score = score + 5

    # Rule 2
    if lowercase:
        score = score + 1

    # Rule 3
    if lowercase and uppercase:
        score = score + 5

    return score

# Test driver
test_passwords = ["sun", "Sun", "Sunshine", "12345", "123456789"]
for password in test_passwords:
    pass_score = calculate_score(password)
    print(pass_score)
```

Make the following changes to the program:

(i) Amend the program so that the output is displayed in the following format:

```
Score   Password
-----   --------
1       sun
6       Sun
11      Sunshine
0       12345
5       123456789
```

(ii) Insert a line of code to change the password contained at index 4 of the list **`test_passwords`** from *123456789* to *Moonlight*.

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

```
Score   Password
-----   --------
1       sun
6       Sun
11      Sunshine
0       12345
11      Moonlight
```

(iii) Amend the program so that it determines and displays the weakest password in the list along with its score.

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

```
Score   Password
-----   --------
1       sun
6       Sun
11      Sunshine
0       12345
11      Moonlight

The weakest password is: 12345
Score: 0
```

(iv) Write a function definition called **`is_strong`** which accepts a password as a parameter and returns **`True`** if the password is strong; **`False`** otherwise.

A password is deemed strong if it contains more than seven characters and both lowercase and uppercase letters.

The first line of the function definition will look like this:

```
def is_strong(password):
```

(v) Modify the program so that it calls the function **`is_strong`** for each password in the list **`test_passwords`** and displays all the strong passwords.

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

```
Score   Password
-----   --------
1       sun
6       Sun
11      Sunshine
0       12345
11      Moonlight

The weakest password is: 12345
Score: 0

The strong passwords are:
Sunshine
Moonlight
```

Use the format **CandidateNumberQuestion16_B.py** to save your file. For example, if your candidate number was 123456 you would save the file as **123456Question16_B.py**.

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.

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.

Leaving Certificate – Higher Level

Computer Science – Section C

1 hour