Coimisiún na Scrúduithe Stáit
State Examinations Commission

Leaving Certificate 2020

Marking Scheme

Computer Science

Higher Level

Note to teachers and students on the use of published marking schemes

Marking schemes published by the State Examinations Commission are not intended to be standalone documents. They are an essential resource for examiners who receive training in the correct interpretation and application of the scheme. This training involves, among other things, marking samples of student work and discussing the marks awarded, so as to clarify the correct application of the scheme. The work of examiners is subsequently monitored by Advising Examiners to ensure consistent and accurate application of the marking scheme. This process is overseen by the Chief Examiner, usually assisted by a Chief Advising Examiner. The Chief Examiner is the final authority regarding whether or not the marking scheme has been correctly applied to any piece of candidate work.

Marking schemes are working documents. While a draft marking scheme is prepared in advance of the examination, the scheme is not finalised until examiners have applied it to candidates' work and the feedback from all examiners has been collated and considered in light of the full range of responses of candidates, the overall level of difficulty of the examination and the need to maintain consistency in standards from year to year. This published document contains the finalised scheme, as it was applied to all candidates' work.

In the case of marking schemes that include model solutions or answers, it should be noted that these are not intended to be exhaustive. Variations and alternatives may also be acceptable. Examiners must consider all answers on their merits, and will have consulted with their Advising Examiners when in doubt.

Future Marking Schemes

Assumptions about future marking schemes on the basis of past schemes should be avoided. While the underlying assessment principles remain the same, the details of the marking of a particular type of question may change in the context of the contribution of that question to the overall examination in a given year. The Chief Examiner in any given year has the responsibility to determine how best to ensure the fair and accurate assessment of candidates' work and to ensure consistency in the standard of the assessment from year to year. Accordingly, aspects of the structure, detail and application of the marking scheme for a particular examination are subject to change from one year to the next without notice.

Marking Scheme – Section C

Structure of the marking scheme for Section C (Programming)

Candidate responses are marked according to different scales, depending on the types of response anticipated. Scales labelled A divide candidate responses into four categories (correct response, almost correct response, partially correct response, and response of no substantial merit), and so on. The scales and the marks that they generate are summarised in this table:

Scale Label	A	B	C
No. of categories	4	5	6
5 mark scale	0, 2, 3, 5	0, 2, 3, 4, 5	
10 mark scale	0, 3, 7, 10	0, 3, 5, 8, 10	0, 2, 4, 6, 8, 10

A general descriptor of each point on each scale is given below. More specific directions in relation to interpreting the scales in the context of each question are given in the scheme, where necessary.

Marking scales – level descriptors

A-scales (4 categories)

- response of no substantial merit
- response with some merit
- almost correct response
- correct response

B-scales (5 categories)

- response of no substantial merit
- response with some merit
- response about half-right
- almost correct response
- correct response

C-scales (6 categories)

- response of no substantial merit
- response with some merit
- response about half-right
- response more than half-right
- almost correct response
- correct response

Section A — Short Answer Questions — 60 marks

Answer all twelve questions.

Question 1 **5 marks**

Expression	Result
`a * b`	**10**
`a ** b`	**25**
`a / b`	**2.5**
`b % a`	**2**
`++a`	**6**

Each correct item 1 mark

Question 2 **2 + 3 marks**

(a)

GB: Gigabyte
TB: Terabyte

Each correct item 1 mark

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

- **SSD is smaller, faster, quieter**
- **SSD has no moving parts - more reliable, durable, better for transportation, etc.**
- **SSD is completely electronic - consumes less power**
- **SSD is safe from magnets**

Very good description - clear understanding demonstrated 3 marks
Good description - clear information, lacking full understanding 2 marks
Fair description - limited understanding 1 mark

Question 3 **5 marks**

Contains address of the instruction to be executed in memory or similar

Very good description - clear understanding demonstrated	5 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	1 mark

Question 4 **2 + 3 marks**

(a)

- **Rules and procedures for network communication or similar**
- **An agreed way to communicate between two computers/devices/processes**

Any one of the above points:	
Good description - clear information	2 marks
Fair description - limited understanding	1 mark

(b)

HTTP, HTTPS, TCP, IP, VOIP, or similar (Ethernet, SMTP, FTP, Telnet)

Others: Echo, finger, Gopher, POP3, DNS, UDP

Protocol	Purpose
HTTP	**High level protocol used to support the exchange of WWW documents (typically HTML)**
HTTPS	**HTTP + added security layer / encryption-decryption**
TCP	**Low level protocol – sender breaks messages into packets hands over to IP (for routing) and re-assembles and orders at the destination**
IP	**Responsible for the routing of packets from source/sending process/device to destination process/device via a network of inter-connected computers/devices**
VOIP	**The transmission of voice over IP**

Name	1 mark
Purpose:	
Good description - clear information	2 marks
Fair description - limited understanding	1 mark

Question 5 **3 + 2 marks**

(a) 0 / False
(b) 0 / False

First correct	3 marks
Second correct	2 marks
Incorrect	0 mark

Question 6 **3 + 2 marks**

(a)

- **Allows data to be encoded in a standard way**
- **Provides a means of representing data as information**
- **Allows for worldwide communication in electronic form or similar**

Any one of the above points:

Good description - clear understanding demonstrated	3 marks
Fair description - limited understanding	2 mark

(b)

Limitation:

7 bits so only max of $2^7 = 128$ characters – 94 printable (accept 8 bits max 256 characters)
Does not allow for emojis, etc
Does not support symbols from different languages
Not enough symbols to support certain languages with large character sets
And similar

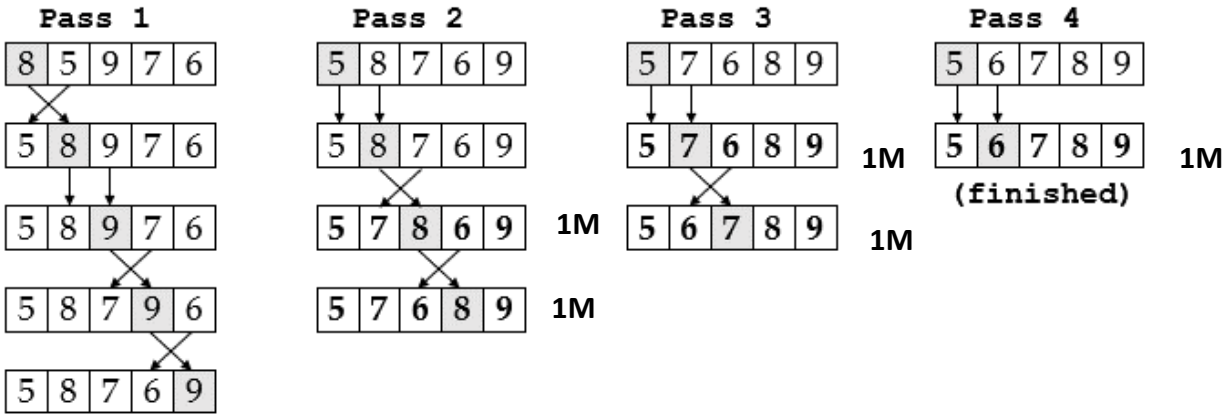
Good description - clear information	2 marks
Fair description - limited understanding	1 mark

Question 7 **5 marks**

4B

Correct	5 marks
Half correct	3 marks
Response with some merit	1 mark

Question 8 **5 marks**



1M 1M 1M 1M

Each correct step 1 mark

Question 9 **5 marks**

Repeated record Ogene
Blank surname
Dot after F
Small f
Negative age
77 age anomaly
Comma in time
Space in time, no decimal point

First two problems 2 marks (each)
Third problem 1 mark

Question 10 **5 marks**

Number	Missing Code
❶	**html**
❷	**Menu**
❸	**ul**
❹	**href**

First item correct	2 marks
Additional item correct	1 mark

Question 11 **5 marks**

Change 2nd `if` to `else`
Rearrange code with correct example.
Similar suitable answer.

Very good description - clear understanding demonstrated	5 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	1 mark

Question 12

Heuristics will find the best approximate solution when it would take too long to calculate.
The best actual solution or similar answer.
Heuristics is a guess which can be based on previous experience or rule of thumb.

Very good description - clear understanding demonstrated	5 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	1 mark

Section B	Long Questions	70 marks

Question 13 **30 marks**

(a) **14 (2, 2, 5, 2, 3) marks**

(i) **2 marks**

Linear search

(ii) **2 marks**

Boole / Boolean (2 marks)
True / False (1 mark)

(iii) **5 marks**

Starts at index 0 – checks/compares first name. Name is John. Found stays as false.
Index increased by 1 – checks/compares second name. Name is Mary. Found stays as false.
Index increased by 1 – checks/compares third name. Name is Zoe. Found changes to true. Exits while loop and prints Zoe, index 2.
Or any similar relevant explanation.

Very good description - clear understanding demonstrated	5 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	1 mark

(iv) **2 marks**

Index = 5

(v) **3 marks**

Worst case: O(n) / length of list

Worst case would mean that the item is not in the list so each item is searched before being found or not found or similar.
In a list of n items there would be at most n comparisons -> worst case.

Answer	1 mark
Explanation:	
Good description - clear information	2 marks
Fair description - limited understanding	1 mark

(b) **9 (2, 5, 2) marks**

(i) **2 marks**

The data set being searched must be sorted and remain sorted.
Can be slower than linear search (average performance is O(log n) vs. average of linear search is O(n/2).
More complicated than linear search and unnecessary for small data sets.
Only works on data with a "less-than" relationship.
Data has to be of the same type.
Or similar answer.

Any suitable answer 2 marks

(ii) **5 marks**

1. **Compares 28 to the value in mid (18). It is greater so subarray to the left is dismissed. Now left is = 28, mid = 41, right = 50 (Note: mid may also be 35 depending on code implementation – do not deduct marks)**
2. **Compares 28 to the value in mid(41 or 35). It is smaller so subarray to the right is dismissed. Now left = 28, mid = 35, right = 41**
3. **Compares 28 to the value in mid (35). It is smaller so subarray to the right is dismissed. Now left = 28, mid = 28, right =28. Returns index 4**
   Similar explanations acceptable.

Alternative

28 > 18 => left <- (mid+1) = 4; mid <- (left+right)/2 = (4+7)/2 = 11/2 = 5
28 < 35 => right <- (mid-1) = 4; mid <- (left+right)/2 = (4+4)/2 = 8/2 = 4
28 = 28 => found

left	mid	right	L[mid]
0	3	7	18
4	5	7	35
4	4	4	28

Very good description - clear understanding demonstrated	5 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	1 mark

(iii) **2 marks**

4

Correct answer	2 marks
Any relevant roughwork	1 mark

(c) **7 marks**

It is important to understand the time and efficiency so that different algorithms can be compared, particularly as data sets grow.
It allows for the best algorithm to be selected for the specific problem.
The best algorithm should be the fastest and/or use least amount of memory.

Very good description - clear understanding demonstrated	7 marks
Good description - clear information, lacking full understanding	5 marks
Fair description - limited understanding	3 marks

Question 14 **20 (4, 8, 8) marks**

(a) **4 marks**

Can compute anything that is possible to be computed or any similar explanation.

Very good description - clear understanding demonstrated	4 marks
Good description - clear information, lacking full understanding	3 marks
Fair description - limited understanding	2 mark

(b) **8 (2, 2, 4) marks**

(i) **2 marks**

S2 is the end/final/halt state.
The FSM is finished computing.
The problem is computable if it reaches this state (deterministic).
Any similar explanation.

Good explanation - clear information	2 marks
Fair explanation - limited understanding	1 mark

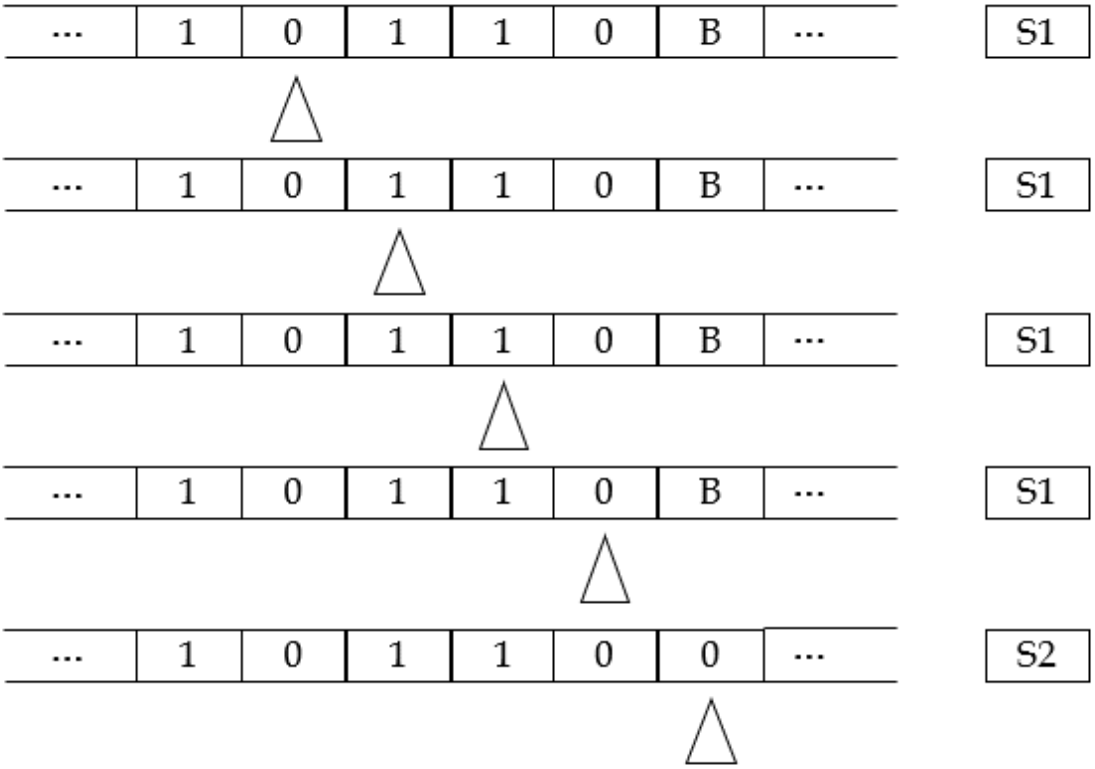
(ii) **2 marks**

Current State	Input (Read)	Output (Write)	Next State
S1	0	0	S1
S1	**1**	**1**	**S1**
S1	**B**	**0**	**S2**

Each correct row 1 mark

(rows can be in any order)

(iii) **4 marks**



Each correct row 1 mark

(c) **8 marks**

Positives:

- **Positive impact on human life e.g. identifying cancer, predicting weather, etc.**
- **Improving daily life – traffic management, shopping suggestions, etc.**
- **Analysing large datasets, self-learning (machine learning) that humans could not do.**
- **Always available, no emotional decisions, faster decisions, more accuracy.**
- **New inventions, solutions or ideas that humans may not think of.**
- **Any similar suitable answer.**

Negatives:

- **Data privacy issues and potential misuse with the use of existing datasets.**
- **Ethical issues – no emotion in decision e.g. law, drones in war.**
- **Unemployment – replace humans in jobs.**
- **No innovation or creativity – AI is limited by the algorithms on which it is based.**
- **Risk of creating sentient machines that develop a conscience.**
- **Any similar suitable answer.**

Mark 2 best points for positive and 2 best points for negative.

For each of the 4 points (2 positive/2 negative):	
Good description - clear information	2 marks
Fair description - limited understanding	1 marks

Question 15 **20 (4, 8, 8 (6,2)) marks**

(a) **4 marks**

Any suitable reasons that demonstrate understanding. Example reasons for a staged approach include simplicity and clarity. Example reasons for an iterative approach include flexibility and focus on end-user. It is also easy to understand and use.

For each reason:

Good description - clear understanding demonstrated	2 marks
Fair description - limited understanding	1 mark

(b) **8 marks**

Role	Responsibilities
Project Manager (Leader/Team Leader)	**Planning, monitoring, resourcing, communication**
Analyst	**Understands the problem and define the system scope**
Designer	**Creates a representation of the system using tools and algorithms**
Programmer / Developer / Coder / Software Engineer	**Implements design – writes the programs – unit test**
Testers/Testing	**Determines whether the system works as intended**
Any similar role	

For each role:

Any relevant role	2 marks
Role description rather than name	1 mark
Any relevant responsibility related to role	1 mark for each

(c) **8 (6, 2) marks**

(i) **6 (2, 4) marks**

Type of application:
The game shown in UI1 looks like a web application probably running on a web browser but possibly running on a mobile device such as a tablet. The game shown in UI2 looks a lot more traditional – most likely a windows application running off a desktop or laptop device.

Good comparison - clear information, good understanding	2 marks
Fair comparison - limited understanding	1 mark

Functionality:
The basic functionality offered by both examples a game of RPS in which the user plays against the computer/bot. UI1 offers the following additional functionality:

- **Social media integration (Facebook and Twitter).**
- **Ability to set and play at different levels (the screen shows the computer is at veteran level).**
- **A log of the result history (UI2 just shows the result of the most recent game).**
- **A display of the number of draws (ties).**
- **UI 1 is more user friendly because it just requires one click to play whereas in UI2 the user is required to choose their move and then click the play button (possibly twice as many clicks).**

Mark 2 best points for comparison of functionality. For each point:	
Good comparison - clear information, good understanding	2 marks
Fair comparison - limited understanding	1 mark

(ii) **2 marks**

Possible answers – voice recognition, hand gestures, touch screen interface, VR/AR (virtual reality/augmented reality).

For each relevant example	1 mark

Section C Programming 80 marks

Question 16

(a) **50 (5, 5, 5, 5, 10, 5, 5, 10) marks**

Possible solution:

```
1
2  # Examination Number:
3
4  # Prompt the user to enter a password and store the ...
5  # value entered in the variable password
6  password = input("Enter a password: ")
7
8  # A variable to store all the lower case letters in the alphabet
9  LOWER_CASE_LETTERS = "abcdefghijklmnopqrstuvwxyz"
10 UPPER_CASE_LETTERS = "ABCDEFGHIJKLMNOPQRSTUVWXYZ" # Ans (iii):
11 DIGITS = "0123456789" # Ans (v)
12
13 # The variables lowercase and uppercase indicate the presence of ...
14 # ... lowercase and uppercase characters in the password
15 lowercase = False # True if password contains at least 1 lowercase letter
16 uppercase = False # True if password contains at least 1 uppercase letter
17 digits = False
18
19 # Loop through each character in the password and ...
20 # ... check the password for specific characters
21 for character in password:
22   if character in LOWER_CASE_LETTERS:
23     lowercase = True
24   if character in UPPER_CASE_LETTERS: # Ans (iii):
25     uppercase = True
26   if character in DIGITS: # Ans (v)
27     digits = True
28
29
30 # Calculate the score based on the rules
31
32 score = 0 # Ans (i): initialise score
33
34 # Rule 1
35 # Ans (viii)
36 if len(password) > 7:
37     score = score + 5
38 elif len(password) >= 4 and len(password) <= 7:
39     score = score + 2
40 else:
```

```
    score = score - 2

# Rule 2
if lowercase:
    score = score + 1

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

# Ans (iv): Rule 4
if uppercase:
    score = score + 2

# Ans (v): Rule 5
if digits:
    score = score + 5

# Ans (vi): Rule 6
if password[0] in DIGITS:
    score = score + 1
if password[-1] in DIGITS:
    score = score + 1
if password[0] in DIGITS and password[-1] in DIGITS:
    score = score + 2

# Ans (vii): Rule 7
if digits and not lowercase and not uppercase:
    score = score - 10


# Display the score
#print(score)
# Ans (ii):
print("Password:", password)
print("Score:", score)
```

(i) **5 marks (A-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
3 marks	**Almost correct response** Comment in inappropriate location.
2 mark	**Response with some merit** Any reasonable attempt at inserting a comment.

(ii) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct implementation of only one output using included solution or similar. Minor string concatenation error.
3 marks	**Response about half-right** Correct implementation using included solution or similar but with one syntax error. Only prints both variables (without text). String and variable printed on separate lines. Any other similar half-right response.
2 marks	**Response with some merit** Attempted use of `print` or similar.

(iii) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation.
3 marks	**Response about half-right** Correct implementation of variable declaration but not used in `if` statement. Variable name changed in `if` statement but not declared. Any other similar half-right response.
2 mark	**Response with some merit** Any attempt at creating variable or any relevant attempt at changing `if` statement.

(iv) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation.
3 marks	**Response about half-right** Attempted use of conditional statement or similar.
2 marks	**Response with some merit** Value of the score changed correctly but not inside conditional statement or similar.

(v) **10 marks (C-10 scale)**

10 marks	**Correct response** Correct implementation using included solution or similar.
8 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation. Correct design of solution but flag not initialised.
6 marks	**Response more than half-right** Correct implementation of solution to determine that the password contains at least one digit (e.g. flag is set correctly by searching through the string for digits, but not tested).
4 marks	**Response about half-right** Attempted use of conditional statement to test flag. Any reasonable attempt to create digits variable, list or flag or use of `isdigit()` function on whole password.
2 marks	**Response with some merit** Value of the score changed correctly but no attempt to set or test flag.

(vi) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar (optimal). Do not award full marks if solution is inside the `for` loop.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation. Correct implementation but the `if` statements are inside the `for` loop (suboptimal).
3 marks	**Response about half-right** Implements any part of the rule correctly using the included solution or similar.
2 marks	**Response with some merit** Attempted use of any combination of conditional statement and string index. Use of conditional statements and attempt at finding a specific character in the string. Value of the score changed correctly.

(vii) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation.
3 marks	**Response about half-right** Relevant attempt at including a condition to test any of digits, lowercase and uppercase.
2 marks	**Response with some merit** Value of the score changed correctly.

(viii) **10 marks (C-10 scale)**

10 marks	**Correct response** Correct implementation using included solution or similar (optimal).
8 marks	**Almost correct response** Correct implementation using multiple `if` statements instead of using `elif` (suboptimal).
6 marks	**Response more than half-right** Code runs but solution is incorrect due to one semantic error e.g. error in conditionals. Any two parts of the solution correctly implemented. Correct implementation using included solution or similar but one syntax error (e.g. relational operators correctly identified and combined using the logical and operator or similar).
4 marks	**Response about half-right** Attempted use of conditional statement **and** relational operators (correctly identified) or similar.
2 marks	**Response with some merit** Value of the score changed correctly.

(b) **30 (5, 5, 5, 10, 5) marks**

Possible solution (note 3 options for function definition):

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

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

# Ans (iv)
def is_strong(password):
    return (calculate_score(password) == 11)

# Ans (iv) - version 1
def is_strong_v1(password):
    strong = False

    if calculate_score(password) == 11:
        strong = True

    return strong

# Ans (iv) - version 2
def is_strong_v2(password):
    strong = False

    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

    if len(password) > 7 and lowercase and uppercase:
        strong = True

    return strong

def calculate_score(password):

    # The variables lowercase and uppercase indicate the presence of ...
    # ... lowercase and uppercase characters in the password
    lowercase = False # True if password contains at least 1 lowercase
letter
```

```
45      uppercase = False # True if password contains at least 1 uppercase
    letter
46
47      # Loop through each character in the password and ...
48      # ... check the password for specific characters
49      for character in password:
50        if character in LOWER_CASE_LETTERS:
51          lowercase = True
52        if character in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
53          uppercase = True
54
55      # Calculate the score based on the rules
56
57      score = 0
58
59      # Rule 1
60      if len(password) > 7:
61          score = score + 5
62
63      # Rule 2
64      if lowercase:
65          score = score + 1
66
67      # Rule 3
68      if lowercase and uppercase:
69          score = score + 5
70
71
72      return score
73
74  # Test driver ...
75  test_passwords = ["sun", "Sun", "Sunshine", "12345", "123456789"]
76  test_passwords[4] = "Moonlight" # Ans (ii)
77
78  print("Score\tPassword") # Ans (i)
79  print("-----\t--------") # Ans (i)
80  lowest_score = 999 # Ans (iii)
81  weakest_password = "" # Ans (iii)
82  for password in test_passwords:
83      pass_score = calculate_score(password)
84      if pass_score < lowest_score: # Ans (iii)
85          lowest_score = pass_score # Ans (iii)
86          weakest_password = password # Ans (iii)
87      print(pass_score, "\t", password) # Ans (i)
88
89  print()# Ans (iii)
90  print("The weakest password is:", weakest_password) # Ans (iii)
91  print("Score:", lowest_score) # Ans (iii)
```

```
92
93  # Ans (v)
94  # Modify the program so that it calls the function is_strong for each
    password in the list, test_passwords, and
95  # ... displays the password if it is strong
96  print()
97  print("The strong passwords are:")
98  for password in test_passwords:
99      if is_strong(password):
100         print(password)
101
```

(i) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation. Correct implementation but display incorrect (e.g. did not use `\t`).
3 marks	**Response about half-right** Correct use of only one `print` statement or similar. Variables are displayed on separate lines.
2 marks	**Response with some merit** Attempt at using `print` statement or similar.

(ii) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation. Changes the wrong password (e.g. index 5) or similar.
3 marks	**Response about half-right** Good attempt at changing the password.
2 marks	**Response with some merit** Hard-coded change of the password in the list.

(iii) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar.
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation. Evidence of attempt to use loop to find the weakest password.
3 marks	**Response about half-right** Program runs but incorrect implementation – identifies the wrong password but has used variables and conditional statement.
2 marks	**Response with some merit** Not working but creates variable(s) and uses conditional statement. Only uses conditional statement and does not store the password in a variable. Creates a variable to store the password but no conditional. Evidence of link between weakest password and the score. Hard-coded password displayed in the correct message. Attempt at sorting list of scores.

(iv) **10 marks (C-10 scale)**

10 marks	**Correct response** Correct implementation using included solution or similar.
8 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation.
6 marks	**Response more than half-right** Correct implementation of test for any two of length, lowercase and uppercase inside the function definition. Function returns Boolean value.
4 marks	**Response about half-right** Correct implementation of test for any one of length, lowercase and uppercase inside the function definition.
2 marks	**Response with some merit** Reasonable attempt at defining function (e.g. function header and body placed at an appropriate location in the program).

(v) **5 marks (B-5 scale)**

5 marks	**Correct response** Correct implementation using included solution or similar. Correct code but returns incorrect solution due incorrect function definition in previous question (no double penalty).
4 marks	**Almost correct response** Correct design of solution evident but syntax/semantic error in implementation.
3 marks	**Response about half-right** Correct solution but without using function calls or similar. Correct use of loop and conditional only or similar.
2 marks	**Response with some merit** Attempt at making function call (outside loop/conditional). Use of `print` statement.