GCSE
COMPUTER SCIENCE

Component 3 Non-exam assessment

For candidates entering for the **June 2018** 8520/CA/CB/CC/CD/CE examination.

To be issued to candidates on 1 September 2017 or as soon as possible after that date.
All teacher-assessed marks to be returned to AQA by 31 March 2018.

Time allowed
- 20 hours

Instructions
- Evidence for the assessment must include a complete listing of all program code together with a report. The report should describe the design of the solution, any features of the coded solution which are not evident from the listing, the testing and any potential enhancements and refinements to the solution.
- Students must use one of the following programming languages:
  - C#
  - Java
  - Pascal/Delphi
  - Python
  - VB.Net

Information
- The assessment is designed to be completed in 20 hours.
- The assessment period is not required to be continuous.
- There are restrictions on when and where students can work on this problem. Please see the Teachers' Notes which accompany this task for more information about these restrictions.
- Students may need to use the Internet to research certain parts of the problem. This must be within the 20 hours.
- Submission may be paper based or electronic using CD/DVD.
- Students will need to complete and sign a Candidate Record Form which declares that the work is their own. This must be countersigned by the teacher and a member of the senior leadership team at your school or college.
- Copyright permission is granted by AQA to use the copyright in the materials on the condition that such use is limited strictly to the personal use by each teacher and their students for the purpose of the preparation for and conduct of the Non-Examination Assessment only. The materials are not to be provided to anyone other than the teacher and the students undertaking the task. The teacher must collect this task back from the students at the end of each session. The use of the materials for the production and publication in any format of teaching materials or any other such material (other than for the teacher's personal use) is strictly forbidden.

Password Checker and Generator

A program needs to be created that allows the user to check the strength of passwords and to generate strong passwords.

The program should check the strength of a password based on a point-scoring system. When a password is entered, points are awarded based on the length of the password and the types of characters contained within the password. Points are deducted if characters are used in a limited way.

The program should also be able to generate a password that, when checked for strength, is classified as a strong password.

The program should work in the following way:

1 A menu is displayed allowing the user to select from the following options:

- Check Password
- Generate Password
- Quit.

2 If the user selects the 'Check Password' option:

- they are asked to enter a password
- if the length of the entered password is less than 8 characters or greater than 24 characters then an appropriate error message is displayed and the user returned to the menu
- the program checks that the password entered only contains allowed characters. The allowed characters are:

  i. upper case letters (`A` to `Z`)
  ii. lower case letters (`a` to `z`)
  iii. digits (`0` to `9`)
  iv. allowed symbols (see **Figure 1**).

  If the password contains a character that is not allowed then an appropriate error message is displayed and the user returned to the menu.

Figure 1

The allowed symbols are:

```
! $ % ^ & * ( ) - _ = +
```

The space character is **not** allowed.

3 A point score is calculated for the entered password. The score is set to the length of the password. For example, if the password is 12 characters in length then the score is set to 12.

Points are added for the following:

- if the password contains at least one upper case letter (A to Z) then 5 points are added to the score
- if the password contains at least one lower case letter (a to z) then 5 points are added to the score
- if the password contains at least one digit (0 to 9) then 5 points are added to the score
- if the password contains at least one of the allowed symbols shown in **Figure 1** then 5 points are added to the score
- if the password contains at least one upper case letter (A to Z) **and** at least one lower case letter (a to z) **and** at least one digit (0 to 9) **and** at least one allowed symbol shown in **Figure 1**, then an additional 10 points are added to the score.

Points are subtracted for the following:

- if the password **only** contains upper and lower case letters (A to Z and a to z) then 5 points are subtracted from the score
- if the password **only** contains digits (0 to 9) then 5 points are subtracted from the score
- if the password **only** contains allowed symbols as shown in **Figure 1** then 5 points are subtracted from the score
- if the password contains three consecutive letters based on the layout of a UK QWERTY keyboard (see **Figure 2**) then 5 points are subtracted from the score for each set of three.

Figure 2

When looking for sequences of three consecutive letters, the case of the letters does not matter. For example, QWE would be the same sequence as QwE or qwe. The password must be checked against each row of letters separately. Examples of sequences are shown in **Figure 3**.

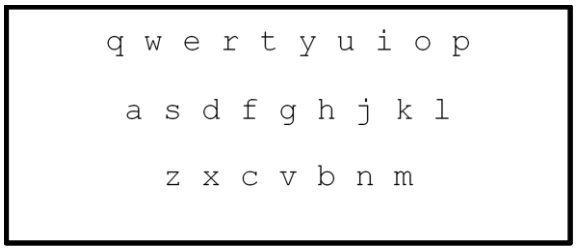

Figure 3

Example sequence	Explanation
`tYu`	Three letters are next to each other on the top row of the QWERTY keyboard, therefore 5 points are subtracted from the score of any password that contains this sequence.
`asdFG`	Five letters are next to each other on the second row of the QWERTY keyboard. In this case there are three sets of three consecutive alphabetic letters: `asd`, `sdF`, and `dFG`, therefore 15 points are subtracted from the score of any password that contains this sequence.
`ZxcTyui`	Three letters are next to each other on the bottom row of the QWERTY keyboard, `Zxc`. Four letters are next to each other on the top row of the QWERTY keyboard, `Tyui`. This gives three sets of consecutive letters: `Zxc`, `Tyu` and `yui`, therefore 15 points are subtracted from the score of any password that contains this sequence.

4 The point score is then used to determine if the password strength is weak, medium or strong. If the point score is over 20 then the password is strong. If the point score is zero or less then the password is weak. The password strength and the point score should be displayed to the user. The user should then be returned to the menu.

5 If the user selects the 'Generate Password' option:

a) the program generates a random number between 8 and 12 inclusive. This number will be the length of the password
b) the program then generates a random sequence of characters using letters, digits and/or allowed symbols to create a password of the length set in a)
c) the point score for the password is then calculated (in the same way as for a user-entered password)
d) parts a) to c) should be repeated until the password strength is strong
e) the generated password and point score should then be displayed and the user returned to the menu.

6 If the user selects the 'Quit' option then a suitable message should be displayed and the program ends.

The following examples show how the 'Check Password' option should work.

EXAMPLE 1

Entered password: `aSD7V^&*gS77+`

Initial score	**Points**
Length is 13	13
Additions	**Points**
At least one upper case character (`S`, `D` and `V`)	5
At least one lower case character (`a` and `g`)	5
At least one symbol (`^` and `&` and `*` and `+`)	5
At least one digit (`7`)	5
One of each of the above types	10
Subtractions	**Points**
Sequence of letters (`aSD`)	–5

Point score for password `aSD7V^&*gS77+` is 13 + 30 – 5 = 38 so this is a strong password.

EXAMPLE 2

Entered password: `qwerty123`

Initial score	**Points**
Length is 9	9
Additions	**Points**
At least one lower case character (`q`, `w`, `e`, `r`, `t` and `y`)	5
At least one digit (`1`, `2` and `3`)	5
Subtractions	**Points**
Consecutive alphabetic characters (`qwe`, `wer`, `ert` and `rty`)	–20

Point score for password `qwerty123` is 9 + 10 – 20 = –1 so this is a weak password.

END OF NON-EXAM ASSESSMENT TASK