Section C

You are advised to spend no more than **60 minutes** on this section.

Enter your answers to **Section C** in your Electronic Answer Document. You **must save** this document at regular intervals.

These questions require you to load the **Skeleton Program** and to make programming changes to it.

1 3 This question extends the functionality of the **Skeleton Program**. A mirror image is to be produced from the loaded image. For example, **Figure 4** shows the image loaded from the `ascii.txt` data file. **Figure 5** shows the mirror image.

Figure 4

	0	1	2	3	4
0	#	#	#	@	@
1	@	@	@	A	A
2	B	B	!	!	!

Figure 5

	0	1	2	3	4
0	@	@	#	#	#
1	A	A	@	@	@
2	!	!	!	B	B

What you need to do:

Task 1
Write a new subroutine `MirrorImage` that produces a mirror image of the image loaded into `Grid` and stores it in a new data structure. The subroutine then displays the mirror image.

Task 2
Amend subroutine `DisplayMenu` to include the new option:

```
M - Mirror image
```

Task 3
Amend subroutine `Graphics` to call `MirrorImage` when the user chooses option `M`

Task 4
Test that the changes you have made work by conducting the following test:
- run the program
- enter `L`
- load **`image1`**
- enter `M`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

1 3 . 1 Your PROGRAM SOURCE CODE for the entire subroutines `MirrorImage`, `DisplayMenu` and `Graphics`

[6 marks]

1 3 . 2 SCREEN CAPTURE(S) showing the requested test, including the menu and the display of the mirror image.

[1 mark]

Turn over for the next question

1 4 This question refers to the subroutine `LoadGreyScaleImage`

A greyscale image can contain a hidden message, which has been encrypted using a key. The message can be revealed by decrypting the image using the method described below.

If an image contains a hidden message, the key to use is the integer value of the digit at the end of the image title in the first line of the file. For example, if the image title is `image4` then the key is 4

When decrypting a message from an image file, each greyscale value is used in turn with the key to produce a result. The key is subtracted from the greyscale value and the result is interpreted according to **Table 2**.

Table 2 describes the rules that should be followed to determine what secret character, if any, is hidden in the greyscale image.

Table 2

Value of result	Interpretation
0	The decrypted character is the space character.
> 0 AND ≤ 26	The result is the position in the alphabet of the decrypted character.
> 26	The result means that the character was not part of the message that was encrypted, and should be represented by an underscore character, ie the _ character.

Example 1
Greyscale value is 1 and the key is 0
1 – 0 = 1
The decrypted character is A as this is the first letter in the alphabet. This should be appended to the hidden message.

Example 2
Greyscale value is 5 and the key is 3
5 – 3 = 2
The decrypted character is B as this is the second letter in the alphabet. This should be appended to the hidden message.

Example 3
Greyscale value is 7 and the key is 7
7 – 7 = 0
The decrypted character is a space as the result was 0. This should be appended to the hidden message.

Example 4
Greyscale value is 52 and the key is 1
52 – 1 = 51
The result is greater than 26 so the greyscale value was not an encrypted character and an underscore character should be added to the hidden message.

The hidden message is built by concatenating each of the decrypted characters.

What you need to do:

Task 1
Write a new subroutine, `FindSecretChar`, which takes as parameters the `PixelValue` and `Key`, where `Key` is the key used to decrypt the hidden character. The subroutine should return the decrypted character if the pixel value was a character from the message, otherwise it should return the underscore character.

Task 2
Amend the subroutine `LoadGreyScaleImage` to:
- get the key from the image title
- call the subroutine `FindSecretChar` in the appropriate place and build up the hidden message from the returned characters
- output the complete hidden message.

Task 3
Test that the changes you have made work by conducting the following test:
- run the program
- enter `L`
- load `greyscale`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

1 4 . 1 Your PROGRAM SOURCE CODE for the entire subroutines `FindSecretChar` and `LoadGreyScaleImage`

[9 marks]

1 4 . 2 SCREEN CAPTURE(S) showing the requested test, including the display of the message and the image.

[1 mark]

Turn over for the next question

1 5 This question extends the functionality of the **Skeleton Program**. A new option, `C`, is to be added to compress an existing ASCII art image file using run-length encoding and store it in a new file.

The number of symbols in a run of symbols is to be counted and this count, separated by a comma from the symbol, is stored in the new file. Each count and symbol pair should be put on a new line.

For example, **Figure 6** shows the contents of the `ascii.txt` data file.

Figure 6

```
TestImage1,5,3,A
###@@@@@AABB!!!
```

The compressed file will contain the file header followed by several value pairs and the file type in the header should be changed from `A` to `C`, as shown in **Figure 7**.

Figure 7

```
TestImage1,5,3,C
3,#
5,@
2,A
2,B
3,!
```

You can assume that there are no newline characters in the ASCII art image file.

What you need to do:

Task 1
Write a new subroutine, `CompressFile`

This subroutine is to:
- ask the user for the name of the file to be compressed
- read the existing file header and change the file type to `C` to indicate that this is a compressed file
- create a new file with the existing filename preceded by `CMP`, for example, if the file read in was `image2` the new filename should be `CMPimage2`
- save the edited file header to the new file
- save each pair of symbol count and symbol separated by a comma on a new line to the new file.

Task 2
Amend subroutine `DisplayMenu` to include the new option:

```
C - Compress file
```

Task 3
Amend subroutine `Graphics` to call `CompressFile` when the user chooses option `C`

Task 4

Test that the changes you have made work by conducting the following test:

- run the program
- enter `C`
- enter the file name `image2`
- load the file `CMPimage2` into a text editor.

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 5 . 1 Your PROGRAM SOURCE CODE for the entire subroutine `CompressFile`

[12 marks]

1 5 . 2 SCREEN CAPTURE(S) showing the requested test, including the menu and all of the contents of the file `CMPimage2` in the text editor.

[1 mark]

END OF QUESTIONS