MORSE CODE: Programming Tasks

The following tasks require you to open the skeleton program and make modifications.

Task 1

This task refers to `GetTransmission`.

Currently, the user must include the suffix '.txt' in order to load a transmission file. If this suffix is not included, the file is not found.

Modify the code so that if '.txt' is not entered by the user as part of the file name, it is added automatically.
If the user enters a file name that includes '.txt', the program should add nothing to the file name.

Evidence you need to provide: **(4 marks)**

- Your amended SOURCE CODE PROGRAM for `GetTransmission`
- One screen capture showing the result of selecting option R from the main menu, followed by entering the file name 'message'
- One screen capture showing the result of selecting option R from the main menu, followed by entering the file name 'message.txt'

Task 2

This task refers to `SendReceiveMessages`.

At present, the main menu loops repeatedly until the user enters a valid option. Modify the code so that, in the event of an invalid entry, the error message 'Please select an option from the menu' is displayed before the user is presented the menu again.

Evidence you need to provide: **(4 marks)**

- Your amended SOURCE CODE PROGRAM for `SendReceiveMessages`
- One screen capture showing the result of selecting option Q from the main menu
- One screen capture showing the result of selecting option S from the main menu

Task 3

This task refers to `SendReceiveMessages`.

Currently, the main menu only accepts uppercase letters. Modify the code so that lowercase letters are accepted in the same way as uppercase letters. For example, a lowercase 'r' should have the same effect as an uppercase 'R'.

All menu options should be selectable using their current uppercase option as well as the lowercase equivalent.

Evidence you need to provide: **(5 marks)**

- Your amended SOURCE CODE PROGRAM for `SendReceiveMessages`
- One screen capture showing the result of selecting option r from the main menu
- One screen capture showing the result of selecting option R from the main menu

Task 4

This task refers to `SendMorseCode`.

When a plaintext message is input for encoding, it must consist of uppercase letters and spaces only. Modify the code to allow users to input messages using lowercase letters, as well as uppercase or a combination.

If the user enters a purely uppercase message, it should be processed as normal. If the user enters lowercase, or a combination of uppercase and lowercase, they should be presented with the following message:

```
Lowercase letters detected - convert to uppercase? (y/n)
```

If they enter 'n', they should be re-prompted to enter the message. Otherwise, any lowercase characters in the message should be converted to their uppercase equivalents.

Evidence you need to provide: **(14 marks)**

- Your amended SOURCE CODE PROGRAM for `SendMorseCode`
- One screen capture showing the result of selecting option S from the main menu, followed by entering the message 'Basic', followed by 'y'
- One screen capture showing the result of selecting option S from the main menu, followed by entering the message 'Basic', followed by 'n'
- One screen capture showing the result of selecting option S from the main menu, followed by entering the message 'BASIC'

Task 5

This task refers to `DisplayMenu` and `SendReceiveMessages`, as well as a new subroutine, `ShowAlphabet`.

Add the following option to the main menu (in any position):

```
D - Display Morse alphabet
```

If the user selects 'D' while `GetMenuOptions` is executing, a new subroutine called `ShowAlphabet` should be called. `SendReceiveMessages` should pass the arrays `Letter` and `MorseCode` to `ShowAlphabet`.

`ShowAlphabet` should display each plaintext character, followed by a space, followed by the Morse code equivalent. Each plaintext letter, together with its Morse code equivalent, should occupy one line, e.g. the first line will be:

```
A .-
```

You should not display the equivalent of a space. After the Morse code alphabet has been displayed, a blank line should be output, followed by the main menu.

Evidence you need to provide: **(11 marks)**

- Your amended SOURCE CODE PROGRAM for `DisplayMenu`
- Your amended SOURCE CODE PROGRAM for `SendReceiveMessages`
- The SOURCE CODE for a new subroutine, in full, called `ShowAlphabet`
- One screen capture showing the result of selecting option D from the main menu

Task 6

This task refers to `GetTransmission` as well as a new subroutine, `GetManualTransmission`.

At the moment, a transmission signal must enter the program from a text file. A transmission signal consists of a series of equals symbols and spaces. Modify `GetTransmission` so that the first lines of code present a menu to the user with two options:

```
F - Read the transmission from a file
M - Enter the transmission manually
```

If the user enters 'F', execution carries on as before, with the program prompting the user for a file name. If the user enters 'M', a new subroutine `GetManualTransmission` should be called. This subroutine should prompt the user for a signal by displaying the following text:

```
Enter transmission signal:
```

When the user enters the signal, it should be passed back as a string to `GetTransmission` and processed in the same way as a file-read string.

No validation is required in `GetManualTransmission`, but the new submenu within `GetTransmission` should loop until the user has entered either 'F' or 'M'.

Evidence you need to provide: **(11 marks)**

- Your amended SOURCE CODE PROGRAM for `GetTransmission`
- The SOURCE CODE for a new subroutine, in full, called `GetManualTransmission`
- One screen capture showing the result of the following set of actions:
    - Select R from the main menu
    - Select M from the submenu
    - Enter the following, substituting each ⌂ with a space, and press enter:

=⌂===⌂⌂⌂=⌂=⌂=

Task 7

This task refers to `StripLeadingSpaces`.

Currently, any spaces at the start of a transmission are removed by `StripLeadingSpaces`, but this approach also removes spaces that were intentionally included within the transmission.

Modify the code so that any transmissions that begin with seven spaces are not trimmed. Transmissions that begin with fewer than seven spaces should be trimmed as normal.

Evidence you need to provide: **(3 marks)**

- Your amended SOURCE CODE PROGRAM for `StripLeadingSpaces`

Task 8

This task refers to `GetTransmission`.

At present, the message 'No transmission found' is displayed for files that are not found as well as files that cause other exceptions.

Modify the code so that a file not being found calls `ReportError` with the string 'File not found', while any other exception should result in a call to `ReportError` with the string 'File error'.

Evidence you need to provide: **(9 marks)**

- Your amended SOURCE CODE PROGRAM for `GetTransmission`
- One screen capture showing the result of selecting option R from the main menu, then entering 'mmm' as the file name

Task 9

This task refers to `GetTransmission`.

Currently, `GetTransmission` returns a string read from a text file without validating that string. The text file should contain dots represented by a single equals symbol, and dashes represented by three consecutive equals symbols. Any other number of consecutive equals symbols would cause an invalid transmission.

After calls to `StripLeadingSpaces` and `StripTrailingSpaces`, and the code that adds the 'end-of-line' character, add code that checks for the following within the `Transmission` string:

- Four equals symbols in a row
- Two equals symbols followed by a space
- Two equals followed by the 'end-of-line' character

If any of these occur, a call should be made to `ReportError`, passing the string 'Invalid file format'. `GetTransmission` should then return an empty string.

Evidence you need to provide: **(6 marks)**

- Your amended SOURCE CODE PROGRAM for `GetTransmission`

Task 10

This task refers to `SendMorseCode`.

Currently, the program converts user-input plaintext into Morse code and displays the Morse code on the screen.

Modify the code so that each new Morse code message generated in `SendMorseCode` is converted to a transmission format (e.g. =⌂===⌂⌂⌂=⌂=⌂=) and displayed in this format on the console.

Evidence you need to provide: **(10 marks)**

- Your amended SOURCE CODE PROGRAM for `SendMorseCode`
- One screen capture showing the result of selecting 'S' on the main menu, then entering 'AQA AS'

Task 11

This task refers to `Decode`.

An individual dot or dash is known as a *symbol*. All possible combinations of one, two or three symbols are assigned to letters. All combinations of four symbols are assigned with the exception of the following:

```
----
---.
.-.-
..--
```

Modify `Decode` so that if any of these four sequences, or any sequence longer than four symbols, is passed to Decode in the `CodedLetter` parameter, an asterisk `*` is returned to `ReceiveMorseCode`. This is to signify that an error has occurred.

Evidence you need to provide: **(4 marks)**

- Your amended SOURCE CODE PROGRAM for `Decode`

Task 12

This task refers to `DisplayMenu` and `SendReceiveMessages`, as well as a new subroutine, `ConvertMorseCode`.

Currently, there is no option for the message to be entered in Morse code at the keyboard. Modify `DisplayMenu` and `SendReceiveMessages` to add the following new menu option:

```
C – Convert Morse code
```

This new menu option will need to call a new subroutine `ConvertMorseCode` and pass two arguments, the arrays `MorseCode` and `Letter`. The new subroutine should ask the user to enter the message in Morse code and print out the decoded message.

Any errors that arise as a result of users entering invalid characters, combinations not present in the Morse alphabet, or incorrect numbers of spaces should be handled with a call to `ReportError` with the following string:

```
Data entry error
```

Evidence you need to provide: **(23 marks)**

- Your amended SOURCE CODE PROGRAM for `DisplayMenu`
- Your amended SOURCE CODE PROGRAM for `SendReceiveMessages`
- The SOURCE CODE for a new subroutine, in full, called `ConvertMorseCode`
- One screen capture showing the result of the following set of actions:
    - Select 'C' from the main menu
    - Enter the following Morse code, substituting each ⌂ with a space, and press Enter:

```
....⌂..⌂⌂⌂-⌂....⌂.⌂.-.⌂.
```


Task 13

This task refers to `GetTransmission`.

Currently, the program reads the first line of the text file and stores it in the variable `Transmission`. Modify the code so that the user is asked 'Which line number' after they have specified a file. If the user specifies '2', only the second line should be stored in transmission.

If the user requests a line that does not exist, or if they enter a value other than an integer, the program should display 'No transmission found' as normal.

Evidence you need to provide: **(7 marks)**

- Your amended SOURCE CODE PROGRAM for `GetTransmission`
- One screen capture showing the result of the following set of actions:
  - Select 'R from the main menu
  - Enter 'message.txt' at the first prompt
  - Enter '2' at the second prompt
- One screen capture showing the result of the following set of actions:
  - Select 'R' from the main menu
  - Enter 'message.txt' at the first prompt
  - Enter '1' at the second prompt

Task 14

This task refers to `SendMorseCode`.

Morse code has been used in the past by the army and navy in times of war. As well as encoding characters in Morse code, they would also probably be encrypted to make them less intelligible to the enemy.

Modify `SendMorseCode` to implement a Caesar cipher with a shift value of 5. The shift values are as follows, with ⌂ used here to represent a space. The top row represents plaintext and the bottom row represents cipher text.

⌂	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	⌂	A	B	C	D

For example, if the character L were to be sent, the Morse code for the character Q would be displayed.

Evidence you need to provide: **(5 marks)**

- Your amended SOURCE CODE PROGRAM for `SendMorseCode`
- One screen capture showing the result of selecting option S from the main menu and entering the message 'BY VWN'

Task 15

This task refers to `SendReceiveMessages` and `SendMorseCode`

Currently, the program works only with letters. In this task, you will modify the arrays `Dash`, `Dot`, `Letter` and `MorseCode` to incorporate numerals using the Morse code values shown in the table below.

You should append the values in the order shown below to the `Letter` and `MorseCode` arrays. You will also need to modify the `Dot` and `Dash` arrays, so that the binary tree structure functions for numerals as well as letters.

You will need to ensure that `SendMorseCode` assigns values of `Index` appropriate to each numeral's position in the array, i.e. the value for 0 (zero) will be stored in the 27th element of the `MorseCode` array, therefore an `Index` value of 27 should be applied. For 1 (one), an `Index` value of 28 should be applied, and so on.

0	`-----`
1	`.----`
2	`..---`
3	`...--`
4	`....-`
5	`.....`
6	`-....`
7	`--...`
8	`---..`
9	`----.`

Evidence you need to provide: **(19 marks)**

- Your amended SOURCE CODE PROGRAM for `SendReceiveMessages`
- Your amended SOURCE CODE PROGRAM for `SendMorseCode`
- One screen capture showing the result of selecting option S from the main menu, then entering '123' as the message to be encoded

