Q	Answer/Guidance	Marks
1a	`EMPTYSTRING`	1
1b	`GetNextSymbol` // `GetNextLetter`	1
1c	`i`	1
1d	`LetterEnd` // `ProgramEnd`	1
1e	`Dash` // `Letter` // `Dot` // `MorseCode`	1
1f	`Dash` // `Dot`	1
1g	`Length` // `Write` // `ReadLine`	1
1h	`i` // `Transmission`	1
2	1 mark for each of the following: • String variable declared • Initialised to user/console input • Specified file accessed/opened/passed to `StreamReader` object • Different/Transmission variable set to first line of the file	4
3	1 mark for each of the following: • (Repeatedly) prompt the user / accept user input • Until X is entered / loop terminates at X	2
4	1 mark for each of the following **(max 3)**: • Integer array • Contains pointers • Indicates which element to move to next… • … if the next Morse signal is a dash	3
5	1 mark for each of the following: • Initially set to the first character in `Transmission` • As spaces are removed, it points to the new first character	2
6	1 mark for each of the following: • Integer is declared • Set to the index of the last character in `Transmission` • Loop repeats while `LastChar` / last character is a space • If the last character is a space, remove it (from `Transmission`) • Decrement `LastChar` / integer variable • Return `Transmission`, with all spaces removed from end/right	6
7	1 mark for each of the following **(max 3)**: • Gets ASCII value of `PlainTextCharacter` • Gets ASCII value of A / Gets value 65 • Subtracts ASCII value of A / 65 from ASCII value of `PlainTextCharacter` and adds 1 • If `PlainTextLetter` is A, `Index` is 1 (for example)	3
8	1 mark for each of the following: • `Catch` block executed if `Try` block fails to execute correctly / generates exception • File name mistyped // file not found // error reading file // error/exception in `StripLeadingSpaces` // error/exception in `StripTrailingSpaces` // `Transmission`/`EOL` not being a valid string	2

Q	Answer/Guidance	Marks
9	1 mark for each of the following **(max 2)**: • Constants won't be accidentally changed • By being at the start of the code, the code is easier to read/understand (*must be clear that this is for the benefit of the human, not the computer*) • No need to remember (precise) values // constant names more memorable than values // code is more readable	**2**
10	1 mark for each of the following **(max 3)**: • A substring is part of a string • Is used here to trim the first character/space from a string • The substring is all characters besides the first one • Is called repeatedly if multiple spaces exist	**3**
11	1 mark for `StripLeadingSpaces` instance: • If the length of `Transmission` is zero 1 mark for `GetTransmission` instance: • If there is a file error (accept any error relating to code in the `Try` block) // if the `Catch` block executes // if the `Try` block fails / generates an error/exception 3 marks for `GetNextSymbol` instance: • If the symbol is not a dot… • … not a dash / minus sign… • … not a space	**5**
12	1 mark for each of the following: • `CodedLetterLength` variable set to 4 • `For` loop to run four times • `Symbol` initially set to − (dash) • `Pointer` set to 20 • `Symbol` then set to dot (on next iteration) • `Pointer` set to 14 • `Pointer` set to 4 (on next iteration) • `Symbol` then set to dash (on next iteration) • `Pointer` set to 24 • X retrieved from `Letter` array / X returned (*only credit this mark if there is an attempt to parse the arrays*)	**10**
TOTAL MARKS		**50**