

UNIVERSITI KEBANGSAAN MALAYSIA

The National University of Malaysia

TITLE: DITS – DAHS

INDIVIDUAL ASSIGNMENT 2

NAME : TEH JUN MING
IC NO. : 991028 - 01 - 5301
MATRIC NO. : A176607
GROUP : GROUP 9
SUBJECT : TTTK1143 – PROGRAM DESIGN AND PROBLEM SOLVING
LECTURER : DR. SYAHANIM MOHD SALLEH
TUTOR : TS. NOOR FARIDATUL AINUN BINTI ZAINAL
COURSE : INFORMATION TECHNOLOGY
FACULTY : INFORMATION SCIENCE AND TECHNOLOGY

	TABLE OF CONTENT	PAGE
1.0	Introduction	1 , 2
2.0	Data Structure	3
3.0	Binary Tree Data Structure	4
4.0	Lists of Classes	5 - 8
4.1	A176607_MorseCode1 Class	6 , 7
4.2	treeNode Class	8
5.0	Algorithm	9 - 14
5.1	Encode Method	9 , 10
5.1.1	Encode Method Algorithm	9
5.1.2	Encode Recursive Method Algorithm	10
5.2	Decode Method Algorithm	10
5.3	Display all Letters and Morse Codes Method	11
5.3.1	Print Inorder Method Algorithm	11
5.3.2	Inorder Recursive Method Algorithm	11
5.4	Main Method Algorithm	12 - 14
6.0	Input and Output of the Program Screenshot	15 - 17

1.0 INTRODUCTION

Swallow Reef, also known as Pulau Layang-layang, is an oceanic atoll of the Spratly Islands administered by Malaysia, which is situated approximately 300 kilometers northwest of Kota Kinabalu, Sabah. The atoll is about 7 km long and 2 km wide. There is a lighthouse on the atoll that sends weather and security conditions periodically to the mainland. Due to the long distance and atmospheric surroundings which are not conducive to modern telecommunications, the only means of communication is by using Morse Code over a low powered radio transceiver.

Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes or dits and dahs. Morse code is named after Samuel Morse, an inventor of the telegraph. The International Morse Code encodes the 26 English letters A through Z, some non-English letters, the Arabic numerals (ten digits which are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9) and small set of punctuation and procedural signals (prosigns). There is no distinction between upper-case and lower-case letters. The following figure shows a list of Morse codes for characters, symbols and Arabic numerals.

Morse Code:									
A	.-	N	-.	.	.-.-.-	1	.-----		
B	-...	O	---	,	---.-	2	..----		
C	-.-.	P	.---	:	---...	3	...---		
D	-..	Q	---.	"	.-...-	4-		
E	.	R	.-.	'	.-----	5		
F	..-.	S	...	!	-.-.---	6	-.....		
G	--.	T	-	?	..--..	7	--...		
H	U	..-	@	.--.-.	8	---..		
I	..	V	...-	-	-.....-	9	-----.		
J	.----	W	.-	;	-.--.	0	-----		
K	-.-	X	-..-	(-.-.				
L	.-...	Y	-.--)	-.-.-				
M	--	Z	--..	=	-...-				

Figure 1 Letters - Morse Code Conversion

Morse code needs trained and experienced officers to be able to communicate efficiently and not easy to obtain as Morse code usage is getting less and lesser by time. Thus, it is best to have a program that can encode and decode Morse messages so that any officer can read or send them in plain text. Although in real life, the codes are sent as sounds.

The java program A176607_MorseCodeApp2 is created to help officer to encode and decode the Morse code and it can print a menu to perform three main process which are encoding letters , decoding Morse code and displaying all letters, symbols and Arabic numerals corresponding to its Morse code.

To ensure each code is unique, the data structure has been used for all Morse code that had shown in **Figure 1 Letter to Morse Code Conversion**. A data structure is a particular way of organizing data in a computer so that it can be used effectively.

2.0 DATA STRUCTURE

A data structure is a particular way of organizing data in a computer so that it can be used effectively. In computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.

In the java program A176607_MorseCodeApp2 tree data structure (binary tree) is been used to perform three main process which are encoding letters , decoding Morse code and displaying all letters, symbols and Arabic numerals corresponding to its Morse code.

The tree is a nonlinear data structure which consist of nodes. The main node or first node is called root and each node is connected to the other node by a link. The node can also have its children which children is the new nodes. In computer science, a tree is a widely used abstract data type that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node, represented as a set of linked nodes. Binary tree is a hierarchical tree data structure in which each node has at most two children generally referred as left child and right child.

Other data structures such as stack, queue, list are linear data structures that store data sequentially. In order to perform any operation in a linear data structure, the time complexity increases with the increase in the data size. But it is not acceptable in today's computational world. Different tree data structures allow quicker and easier access to the data as it is a non-linear data structure. Hence the java program A176607_MorseCodeApp2 is implemented using tree data structure (binary tree).

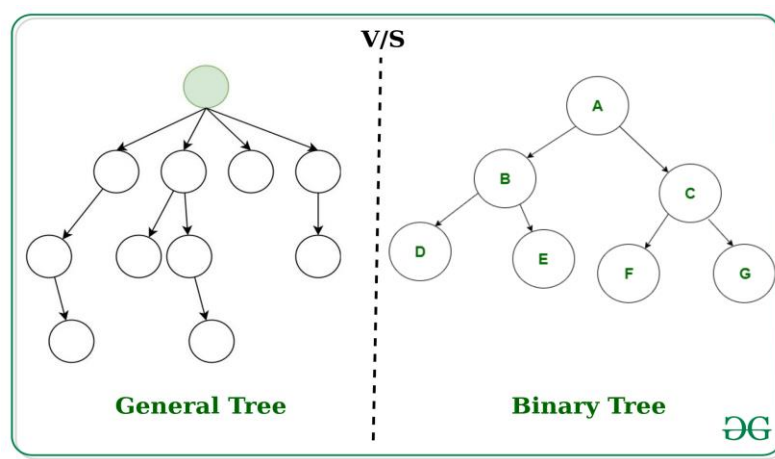


Figure 2 Comparison between tree and binary tree data structure

3.0 BINARY TREE DATA STRUCTURE

In this program, dot or dit are written as “.” and dashes or dahs are written as “-”. Letters are separated by a space, while one word to another are separated by two spaces. The binary tree below can be used to encode and decode Morse code. Each left child represents a dot or dit in the code, and each right child represents a dash or dah in the code. Below is the illustration of the structure of Morse code after inserting all the letter, symbol and Arabic numerals in it based on its International Morse Code : -

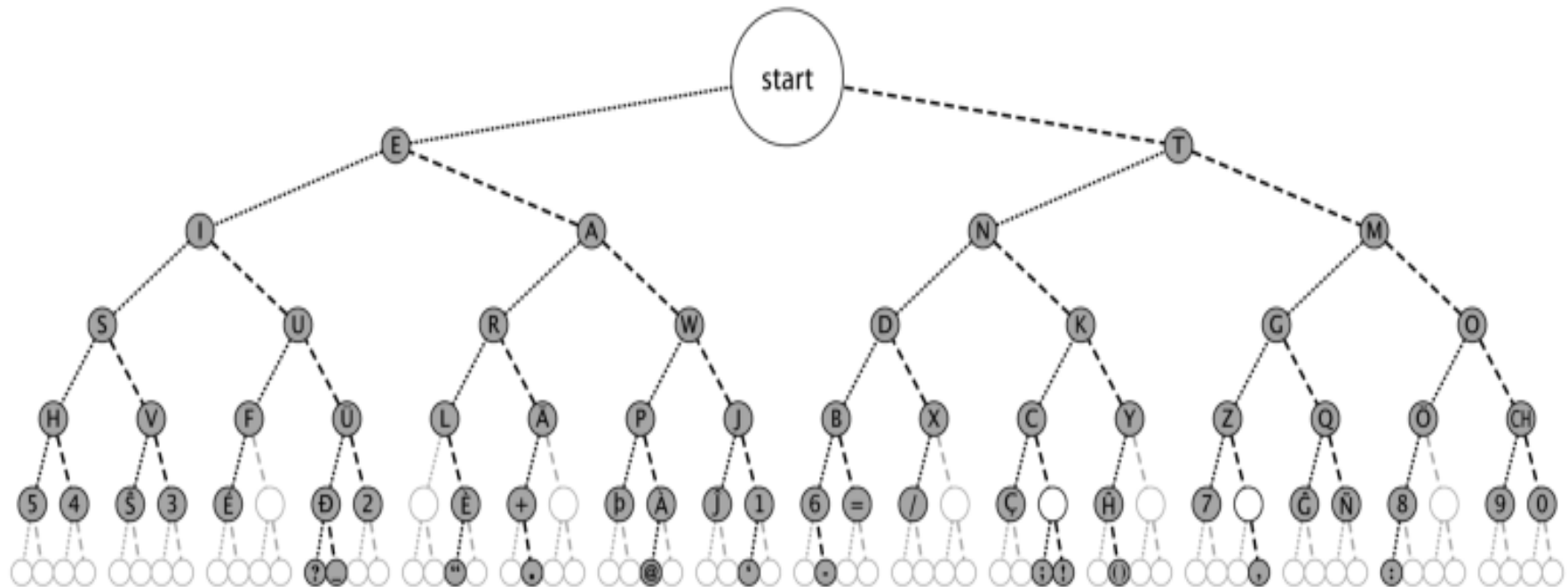


Figure 3 Morse Code - Letter Binary Tree

4.0 LIST OF CLASSES (3 CLASSES)

Class Name	Description
A176607_MorseCode1	Class that allows user to build a binary tree for Morse code and perform some instructions. It creates a binary tree, read a file and insert the letters and Morse codes from the file into the binary tree. It also includes the methods that allow user to encode the letter, decode the Morse code, display the transmission summary, display the summary analysis and display all letters, symbols and Arabic numerals corresponding to its Morse code
treeNode	Subclass of A176607_MorseCode1 that allows user to create a node for the tree
A176607_MorseCodeApp2	The main Morse code application class that allows user to execute and interact with it

Table 1 List of Classes and its Description

4.1 A176607_MorseCode1 Class

A176607_MorseCode1	Description
- root : treeNode	Dynamic attribute (object) that represents the node (root) of a tree
- countLine : int	Dynamic attribute that represents the number of the line
- countWord : int	Dynamic attribute that represents the number of the word
- countCharacter : int	Dynamic attribute that represents the number of the character
- countSymbol : int	Dynamic attribute that represents the number of the symbol
- countNumber : int	Dynamic attribute that represents the number of the Arabic numerals
# lineNumber : String	Dynamic attribute that represents the number of the line
# wordNumber : String	Dynamic attribute that represents the number of the word
# characterNumber : String	Dynamic attribute that represents the number of the character
# symbolNumber : String	Dynamic attribute that represents the number of the symbol
# numberNumber : String	Dynamic attribute that represents the number of the Arabic numerals
+ A176607_MorseCode1()	Constructor that will set all its attributes to their default value, create the tree node, build a binary tree, read a file and insert the letters and Morse codes from the file into the binary tree

+ encode(String) : void	Method that will take word (String) as an argument and will separate it to character, count countLine (int), countWord (int), countCharacter (int), countSymbol (int) and countNumebr (int). It will also recurse the encodeRec(treeNode, char) method to encode the word
+ encodeRec(treeNode, char) : void	A recursive method that will take root (treeNode) and letter (char) as an argument and will transverse the Morse code - letter binary tree to search the same letter with it and display its corresponding Morse code
+ transmissionSummary() : void	Method that will cast countLine (int), countWord (int), countCharacter (int), countSymbol (int) and countNumebr (int) into String, assign them to lineNumber (String), wordNumber (String), characterNumber (String), symbolNumber (String) and numberNumber (String) respectively, encode them and set them back to default value, 0. It will also encode "EOT"
+ decode(String) : String	Method that will take Morse code (String) as an argument and will separate it to character, search the same Morse code with it and display its corresponding letter and return the word (String)
+ summaryAnalysis() :void	Method that will display lineNumber (String), wordNumber (String), characterNumber (String), symbolNumber (String) and numberNumber (String) that have been assigned while at encode(String) method
+ printInorder() : void	Method that will recurse the inorderRec(treeNode) method
+ inorderRec(treeNode) : void	Method that will transverse the Morse code - letter binary tree and displaying all letters, symbols and Arabic numerals corresponding to its Morse code.

Table 2 A176607_MorseCode1 Class and its Description

4.2 treeNode Class (Subclass of A176607_MorseCode1 Class)

treeNode	Description
- letter : char	Static attribute that represents the letter, symbol or Arabic numeral
- morseCode : String	Static attribute that represents the Morse code
- leftChildNode : treeNode	Dynamic attribute (object) that represents the left child node of a node (root) in a tree
- rightChildNode : treeNode	Dynamic attribute (object) that represents the right child node of a node (root) in a tree
+ treeNode()	Constructor that will set all its attributes to their default value

Table 3 treeNode Class and its Description

5.0 ALGORITHMS

5.1 Encode Method

5.1.1 Encode Method Algorithm

Read the word (String)

Covert the String to uppercase and assign it to an uppercase word (String)

Assign root to current node (treeNode)

Split the word (String) and assign it to an array

Repeat until array is end

Split word is trimmed

Number of the word add 1

Repeat until variable i reach the end of length of uppercase word (String)

Assign character at index i of the uppercase word (String) to letter (char)

If letter (char) is equal to symbol

Number of symbol add 1

Else if letter (char) is equal to digit

Number of number add 1

If letter (char) is not equal to space

Recurse the encode(treeNode, char) method

Number of character add 1

Assign root to current node (treeNode)

Else if letter (char) is equal to space

Display space

Number of the line add 1

Display a line

5.1.2 Encode Recursive Method Algorithm

Read the root (treeNode) and the letter (char)

If root (treeNode) is equal to null

Return

Else

Recurse the encode(treeNode, char) method using the left child node of the root

If the letter of the root is equal to the letter (char)

Display the Morse code of the root and a space

Recurse the encode(treeNode, char) method using the right child node of the root

5.2 Decode Method Algorithm

Read the Morse code (String)

Declare and initialize a word (String)

Assign root to current node (treeNode)

Repeat until variable i reach the end of length of Morse code (String)

Assign character at index i of the Morse code (String) to symbol (char)

If symbol (char) is equal to dot or dit

Assign the left child node of the current node (treeNode) to current node (treeNode)

Else if symbol is equal to dash dah

Assign the right child node of the current node (treeNode) to current node (treeNode)

Else if symbol (char) is equal to space and variable i is not equal to the end of the Morse code (char)

Word (String) add the letter of the current node (treeNode)

Assign root to current node (treeNode)

Word (String) add the letter of the current node (treeNode)

Display the word (String)

Return word

5.3 Display All Letters and Morse Codes Method

5.3.1 Print Inorder Method Algorithm

Recurse inorderRec(treeNode) method

5.3.2 Print Inorder Method Algorithm

Read the root (treeNode)

If root (treeNode) is equal to null

Return

Else

Recurse the inorder(treeNode) method using the left child node of the root

If the letter of the root is equal to 3, 2, ., J, B, C,), , or 8

Display a line

If the letter of the root and the Morse code of the root is not equal to null

Display the letter of the root and Morse code of the root

Recurse the inorder(treeNode) method using the right child node of the root

5.4 Main Method Algorithm

Create a scanner

Create a Morse code tree (A176607_MorseCode1)

Try

Read number (int) input

While number (int) is not equal to 4

If number (int) is among 1 to 3

Create another Scanner

Switch number (int)

Case 1

Read word (String) input

If word (String) is equal to VV

While word (String) is not equal to EOM

Encode the word (String)

Read word (String) input

Encode EOM

Run transmission summary method

Else

Display fail message

Display a line

Break

Case 2

Read Morse code (String)

If Morse code (String) is equal ...- ...- or ...- ...-

While Morse code (String) is not equal to . --- -- or . --- --
 Decode Morse code (String)
 Read Morse code (String)

Decode . --- --

Read Morse code (String), decode it and assign it to lineNumber (String)

Read Morse code (String), decode it and assign it to wordNumebr (String)

Read Morse code (String), decode it and assign it to characterNumebr (String)

Read Morse code (String), decode it and assign it to symbolNumebr (String)

Read Morse code (String), decode it and assign it to numberNumebr (String)

Read Morse code (String) and decode it

Display a line

Run summary analysis method

If lineNumber (String) is equal to lineNumber (String) of the Morse code tree (A176607_MorseCode1) and wordNumber (String) is equal to wordNumber (String) of the Morse code tree (A176607_MorseCode1) and characterNumber (String) is equal to characterNumber (String) of the Morse code tree (A176607_MorseCode1) and symbolNumber (String) is equal to symbolNumber (String) of the Morse code tree (A176607_MorseCode1) and numberNumber (String) is equal to numberNumber (String) of the Morse code tree (A176607_MorseCode1)
 Display consistent result
Else
 Display inconsistent result

Set lineNumber (String) of the Morse code tree (A176607_MorseCode1), wordNumber (String) of the Morse code tree (A176607_MorseCode1), characterNumber (String) of the Morse code tree (A176607_MorseCode1), symbolNumber (String) of the Morse code tree (A176607_MorseCode1), numberNumber (String) of the Morse code tree (A176607_MorseCode1) to the default value, null

Else

Display fail message

Display a line

Case 3

Run Print Inorder method

Display 2 lines

Break

Else

Display fail message

Display menu of the program

Read the number (int)

Display exit message

Catch Exception

Display fail message

6.0 INPUT AND OUTPUT OF THE PROGRAM SCREENSHOT

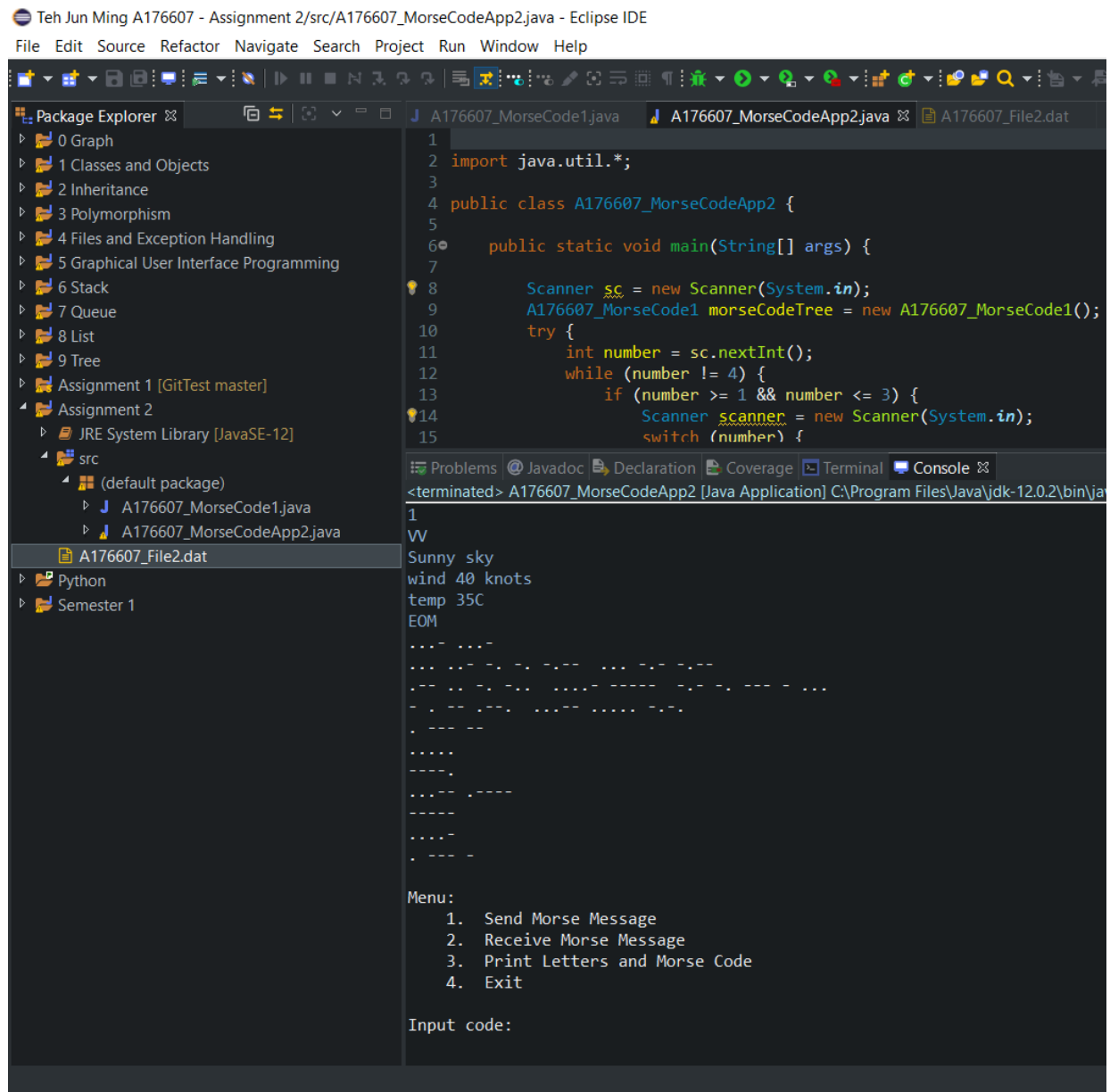


Figure 4 Input and Output of the Program Screenshot 1

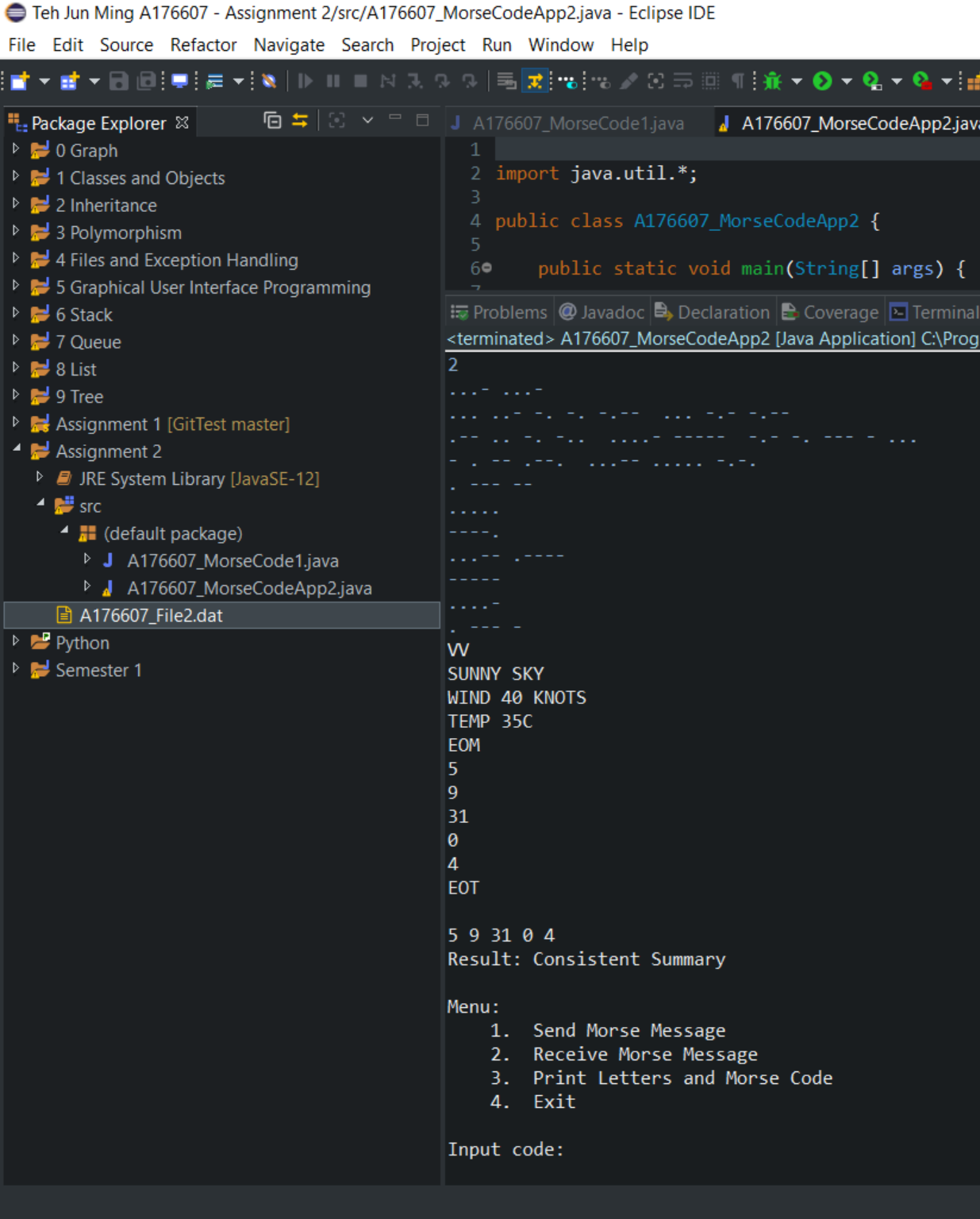


Figure 5 Input and Output of the Program Screenshot 2

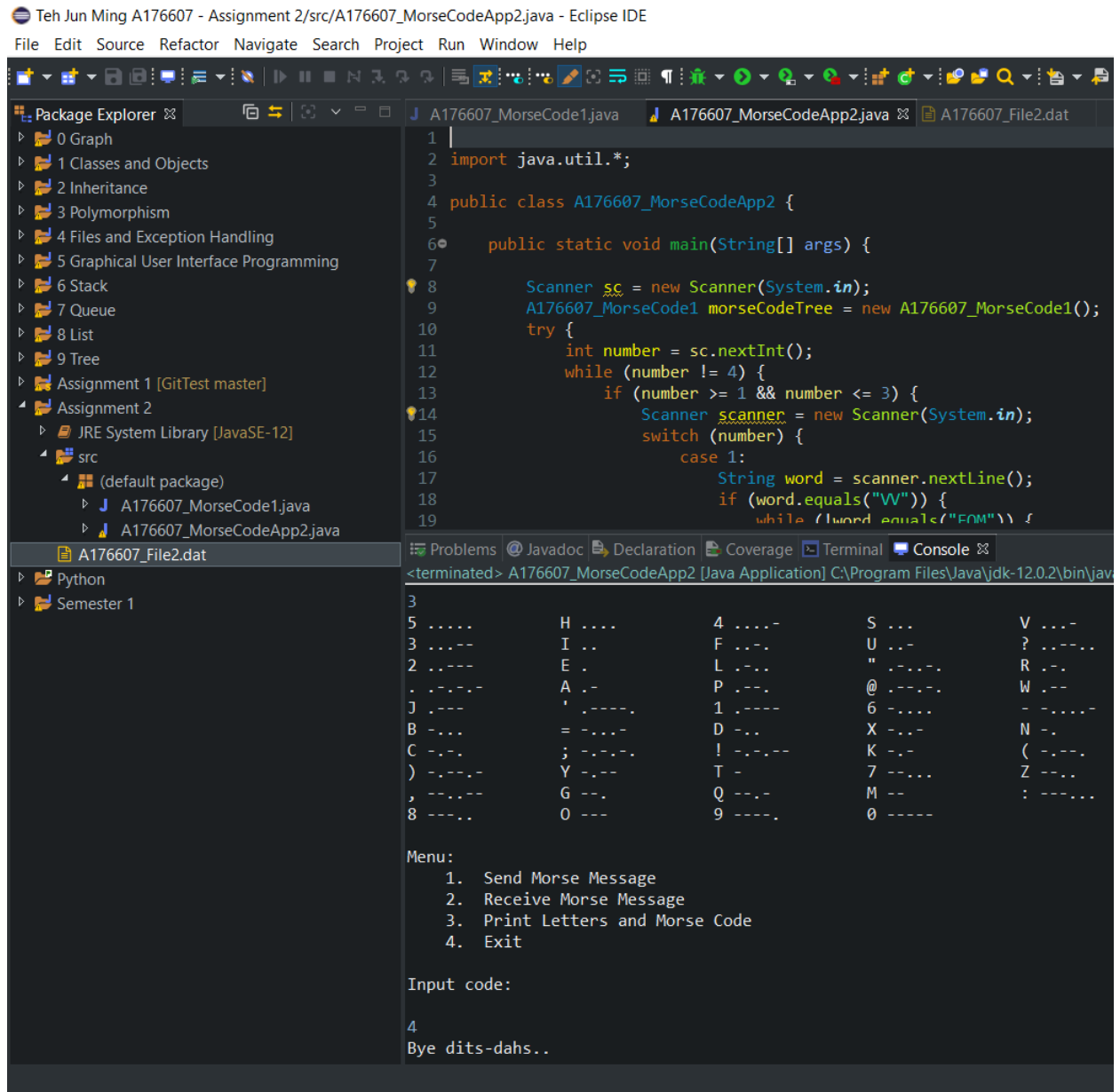


Figure 6 Input and Output of the Program Screenshot 3