**STRING PROGRAMS**

PROGRAM 1

Write a program in java to accept an encoded string and display the decoded string.

Sample input:

Kdyh d qlfh gdb!

Shift value: 3

Sample output:

Have a nice day!

87

Algorithm

Step 1: Start.

Step 2: Scanner class object declared.

Step 3: A sentence and shift value is accepted from the user.

Step 4: The following steps will be executed from 0 to the length of

the sentence .

Step 4.1: Each character is extracted based on the step value of i and stored in variable ch.

Step 4.2: if ch is between ‘a’ and ‘z’ then subtract shift value from ch.And store it in ch variable only. If ch is greater than ‘a’ then Perform (ch+’z’-‘a’+1) and store it to ch variable only and In variable decryptedmessage concatenate the character.

Step 4.3: if ch is between ‘A’ and ‘Z’ then subtract shift value from ch. And store it in ch variable only. If ch is greater than ‘A’ then Perform (ch+’Z’-‘A’+1) and store it to ch variable only and in variable decryptedmessage concatenate the character.

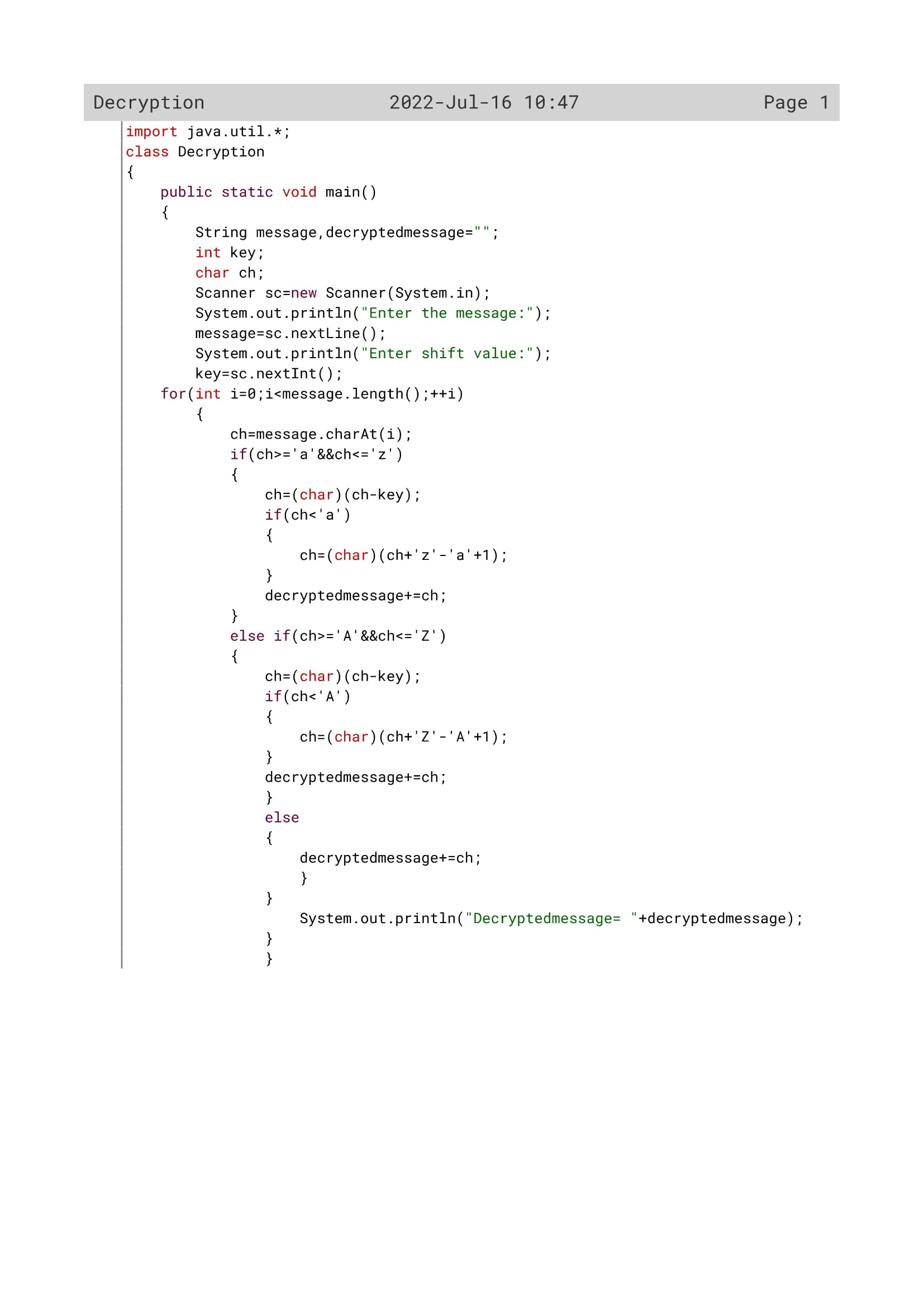
Step 4.4: If the above two conditions do not satisfy then simply concatenate the character to decryptedmessage.

Step 5: After the concatenation print the decoded string.

Step 6: Stop.

88

Source Code



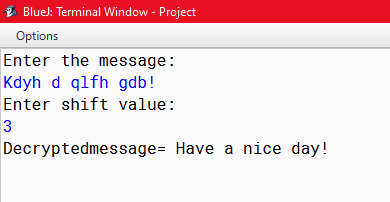
89

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| message | String | To input the message by the user |
| decryptedmessage | String | To display the decrypted message to the user |
| key | Int | To input the shift value |
| ch | Char | To change the letters of the message with respect to the shift value |
| i | Int | Loop variable |

90

OUTPUT



91

PROGRAM 2

Write a program in java to accept a string and display the new string after encoding.

Sample input:

Enter a sentence: Good morning.

Enter the shift value: 3

Sample output:

Encoded text: Jrrg pruqlqj

92

Algorithm

Step 1: Start.

Step 2: Scanner class object declared.

Step 3: A sentence in Uppercase and shift value is accepted fromthe user.

Step 4: The following steps will be executed from 0 to the length of the sentence l.

Step 4.1: Each character is extracted based on the step value of i and is converted to its ascii value and stored in x variable.

Step 4.2: if the sum of x and shift value is greater than 90 then subtract 26 from x and add the shift value. Store it in y variable. Convert the integer to character and concatenate it with st1 variable.

Step 4.3: if the sum of x and shift value is lesser than 90 then just addx and the shift value. Convert the integer to character and concatenate it with st1 variable

Step 5: the encoded string will be displayed.

Step 6: Stop.

93

Source Code



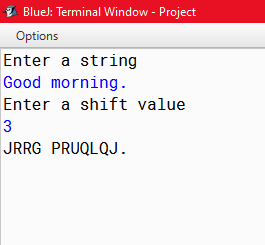
94

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| y | int | To find the letters after shifting |
| x | Int | To find the letters after shifting |
| st1 | String | To display the message to the user |
| st | String | To input the string by the user |
| v | int | To enter the shift value by the user |
| l | Int | To find the length |
| i | Int | Loop variable |
| ch | char | To store the letters of the message with respect to the shift value |

95

OUTPUT



96

PROGRAM 3

Write a program in java to accept sentence and display only the unique words in the sentences and count them.

Sample input:

Enter a sentence: Computer science by Sumita arora

Sample output:

Unique words in the sentence are: Computer

by

Sumita

97

Algorithm

Step-1: Start

Step-2: Scanner class object declared.

Step-3: A sentence is accepted in uppercase from the user .

Step-4: using StringTokenizer object separate the words of the sentence and find the length of each word.

Step-5: execute the following step until each word is extracted .

Step-5.1: Run a loop i from 65 to 90 and an inner loop from 0 to the length of word

Step 5.2: extract the characters of the word and check if ascii value of the character is equal to i value or not. if the condition satisfies then increment c Variable by 1

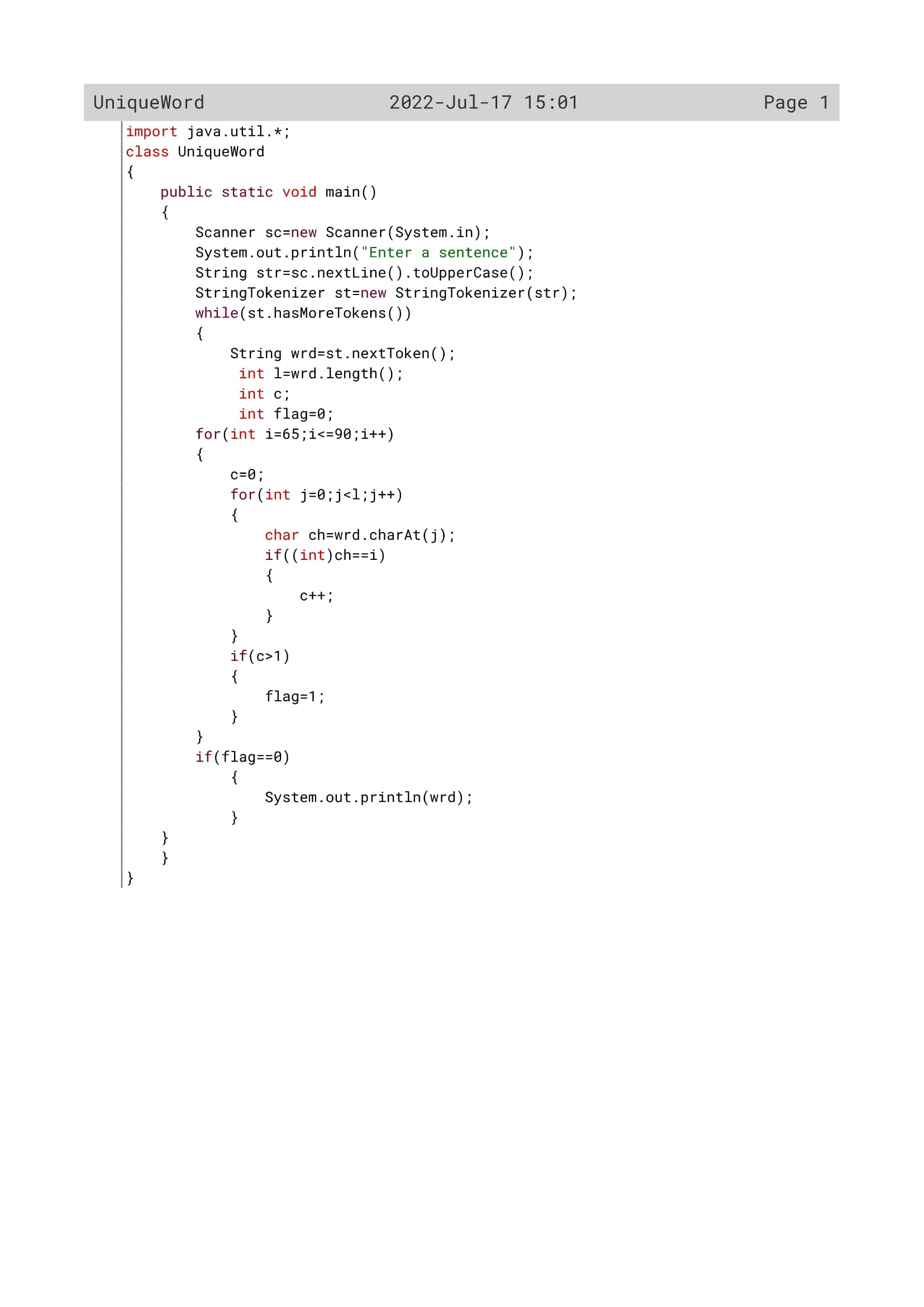
Step 5.3: Check if c is greater than 1 or not.if it is greater than 1 then initialize flag variable with 1.

Step 5.4: If flag=0 then print the word.

Step 6: Stop.

98

Source Code



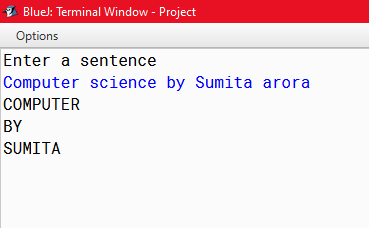
99

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | description |
| str | String | To input the string y the user |
| Wrd | String | To display the unique word to the user |
| i | Int | Loop variable |
| c | Int | Flag variable |
| l | Int | To find the length |
| ch | Char | To extract the letters and check whether the string is unique or not |
| flag | Int | Flag variable |

100

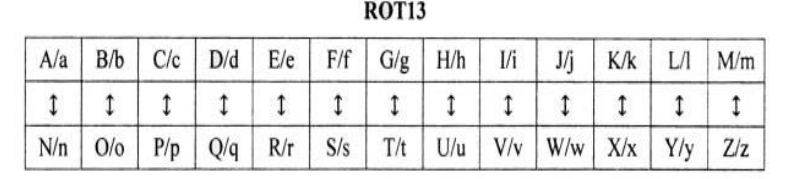
OUTPUT



101

PROGRAM 4

Caesar Cipher is an encryption technique which is implemented as ROT13 (‘rotate by 13 places’). It is a simple letter substitution cipher that replaces a letter with the letter 13 places after it in the alphabets, with the other characters remaining unchanged.



Write a program to accept a plain text of length L, where L must be greater than 3 and less than 100. Encrypt the text if valid as per the Caesar Cipher. Test your program with the sample data and some random data:

**Example 1**

INPUT : Hello! How are you?

OUTPUT : The cipher text is: Uryyb! Ubjnerlbh?

**Example 2**

INPUT : You

OUTPUT : INVALID LENGTH

102

Algorithm

Step 1: Start.

Step 2: Scanner class object declared.

Step 3: A sentence and shift value is accepted from the user.

Step 4: The length of the sentence is calculated.

Step 5: The following steps will be executed from 0 to the length.

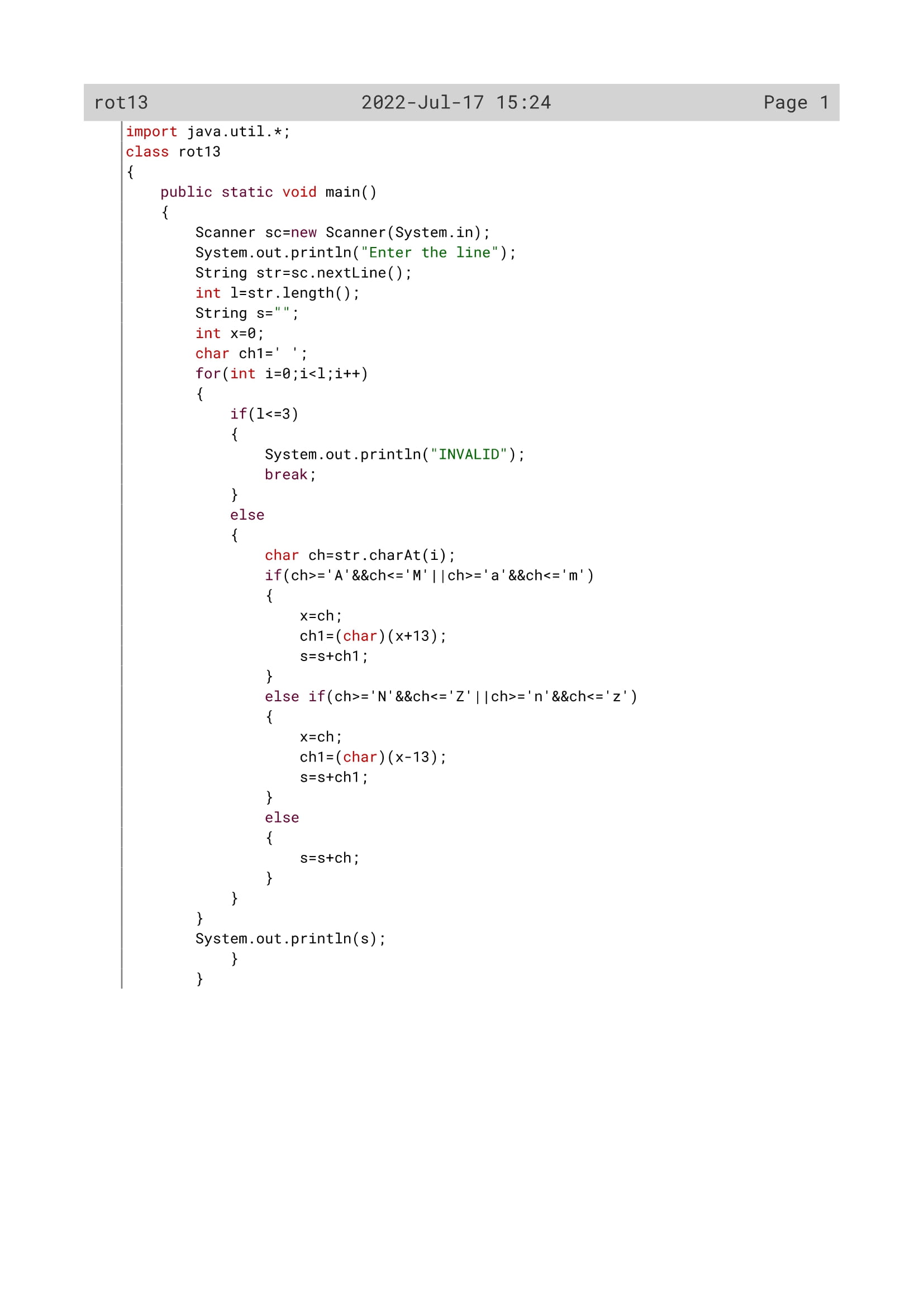
Step 5.1: If the length is greater than 3 and less than 100 then perform the operations

Step 5.2: Extract each character and find out its ASCII value. Add 13 with the values if the character is between ‘A’ and ‘M’ or ‘a’ and ‘m’. subtract 13 with the values if the character is between ‘N’ and ‘Z’. Convert the new values to its characters and concatenate them. Any special character or Blank space will remain just the same.

Step 6: Display the new encoded string.

Step 7: Stop.

103

Source Code

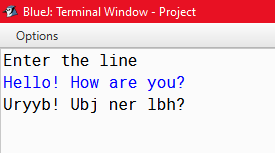
104

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| str | String | To accept the string by the user |
| l | Int | To find the length |
| x | Int | To store the letters |
| s | String | To display the string to the user |
| ch1 | Char | Used for correcting the letters which will exceed the alphabetical order if letters rotates by 13 places |
| i | Int | Loop variable |
| ch | Char | To extract the letters and rotate it by 13 places |

105

OUTPUT



106

PROGRAM 5

Write a program in java to accept a name and display its numerology number.

(note: raj= 18 +1 +10 = 29 = 11. 11 is the numerology number of raj. Numerology can be 1, 2, 3, 4, 5, 6, 7, 8, 9 and 11)

Sample input:

Enter your name: Riya

Sample output:

The numerology number of Riya is 8

107

Algorithm

Step-1: Start

Step-2: Scanner class object declared.

Step-3: A word is accepted in uppercase fromthe user

Step-4: the length of the word is calculated.

Step-5: an object Numeralogy is created.

Step-6: The following steps are executed from o to the length of the word

Step-6.1:Extract each character and convert it To Integer type. substract 65 from the value And add it with s variable.

Step-7: If the sum is greater than 9 then pass The number to sumofdigit function and Store the returned value in sum Variable And again initialize to s variable.

Step-7.1: If s value is equal to 11 then print The numeralogy Number of the word as 11

Step-8: Print the numeralogy number of the word

Step-9: stop.

**Algorithm for sumofdigit()**

Step-1: Start

Step-2: until the number is equal to 0, execute the following steps.

Step-2.1: extract each number and store it to r.

Step-2.2: find the summation by adding r to s variable.

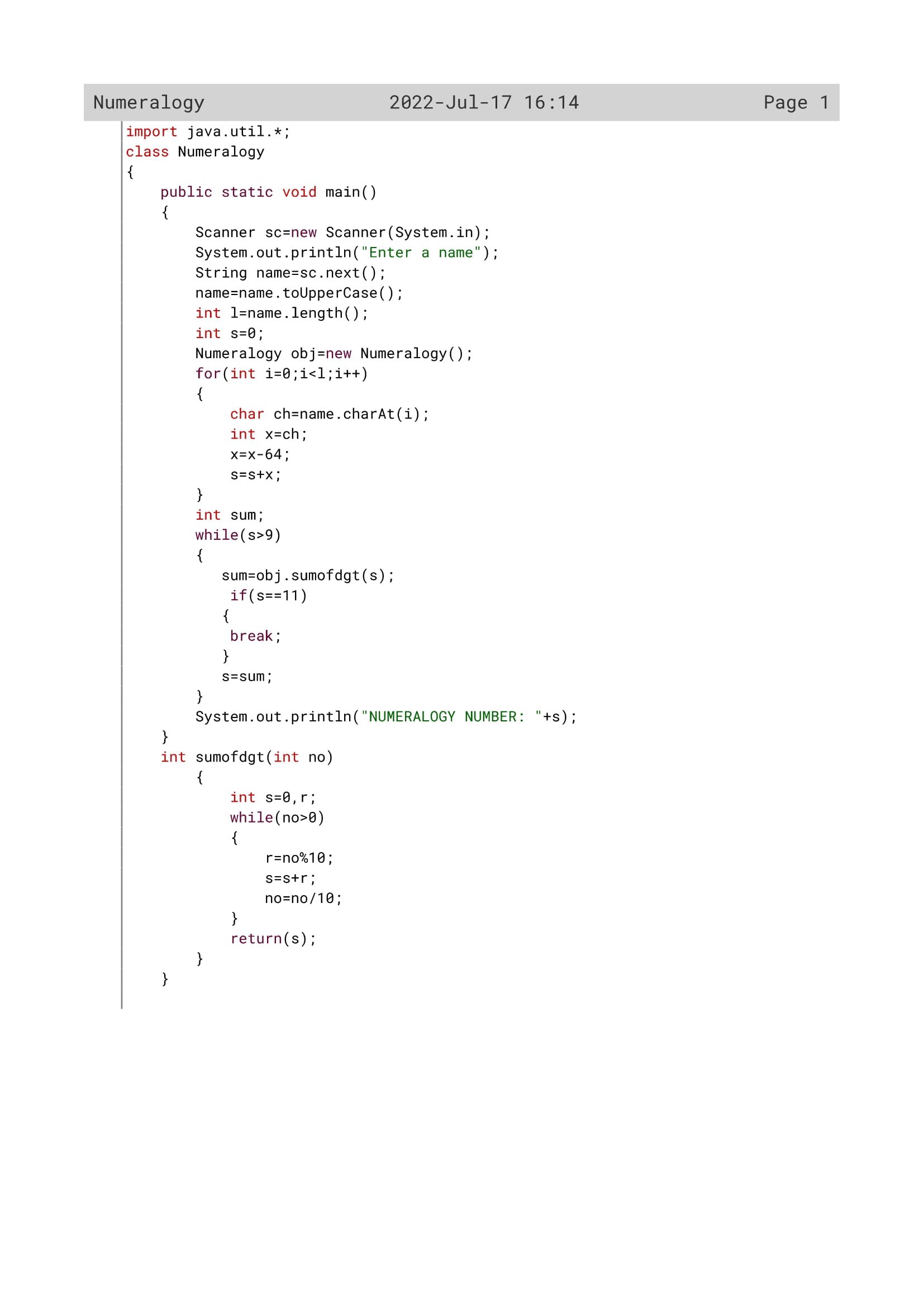
Step-2.3: find the quotient by no/10 operation.

Step-3: Return the variable s as sum of The Number.

Step-4: stop

108

Source Code



109

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | description |
| name | String | To accept a name by the user |
| l | Int | To find the length of the string |
| s | Int | To send the number to the method sumofdgt() if it is greater than 9 |
| x | Int | To calculate the numeric position of a letter |
| ch | Char | To extract the letters and find the numerology |
| r | Int | To find the remainder inside the method sumofdgt() |
| s | Int | To find the sum of itsdigits inside the method sumofdgt() |
| no | int | To act as parameter inside the method sumofdgt() |

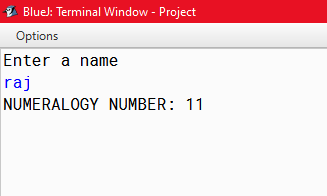
110

Function Description

|  |  |  |
| --- | --- | --- |
| Function name | Data type | description |
| sumofdgt() | Int | To calculate sum of digits |
| main() | void | To accept a name and calculate its numerology number |

111

OUTPUT



112

PROGRAM 6

Write a program that reads a sentence (the characters of the sentence may be capital or small or mixed) and two positive integers and output the same sentence after replacing those words present at those given integer places by the next character in a circular fashion according to the English Alphabets.

Input Sentence: He has good Books.

Input Integers: 2, 4

Output Sentence: He ibt good Cpplt.

(i.e. word number 2 and 4 have been replaced by the next characters in a circular fashion)

Input Sentence: Time and tide waits for none.

Input Integers: 3, 3

Output Sentence: Time and ujef waits for none.

113

Algorithm

Step 1: Start.

Step 2: Scanner class object declared.

Step 3: A sentence and two positions a and b are accepted from the user.

Step 4: using StringTokenizer object separate the words of the sentence and when each word is extracted increment the value of c by 1.

Step 5: check if a is equal to c or b is equal to c. if any one of the conditions satisfy then find the length of the word and store it to l variable.

Step 6: the following steps will be executed from 0 to l.

Step 6.1: each character is extracted and stored to variable ch. If ch is ‘Z’ or ‘z’ then store ‘A’ or ‘a’ respectively in nw variable. Else convert the character to integer, add 1 to it and convert it to its corresponding character, add the characters to nw variable.

Step 6.2: display the newword by printing nw.

Step 7: Stop.

114

Source Code



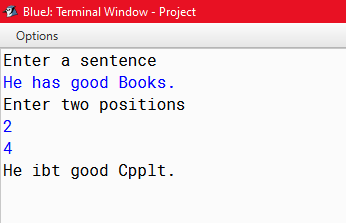
115

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| str | String | To accept a sentence from the user |
| a | Int | To enter the first position by the user |
| b | Int | To enter the second position by the user |
| c | int | Flag variable |
| w | String | To extract the words using string tokenizer |
| nw | String | To calculate the new word |
| i | Int | Loop variable |
| l | Int | To calculate the length |
| ch | char | To extract to letters from the word |
| X | Int | To store the data of ‘ch’ variable |

116

OUTPUT



117

PROGRAM 7

Write a program to accept a sentence which may be terminated by either’.’, ‘?’or’!’ only. The words may be separated by more than one blank space and are in UPPER CASE.

(a) Find the number of words beginning and ending with a vowel.

(b) Place the words which begin and end with a vowel at the beginning, followed by the remaining words as they occur in the sentence.

Test your program with the sample data and some random data:

**Example 1**

INPUT: ANAMIKA AND SUSAN ARE NEVER GOING TO QUARREL ANYMORE.

OUTPUT: NUMBER OF WORDS BEGINNING AND ENDING WITH A VOWEL= 3

ANAMIKA ARE ANYMORE AND SUSAN NEVER GOING TO

118

Algorithm

Step 1: Start.

Step 2: Scanner class object declared.

Step 3: A sentence and shift value is accepted from the user.

Step 4: The length of the sentence is calculated.

Step 5: The last character is extracted. Print invalid if the last character is not a full stop or question mark or exclamatory mark.

Step 6: The following Steps will be executed until all the words are extracted.

Step 6.1: Each word is extracted and its length is calculated.

Step 6.2: The first character and last character are extracted and ischecked if they are vowels or consonant. If they are vowels then the word will be concatenated to one variable w1 otherwise in another variable w2 with single blank space.

Step 6.3: The number of words beginning and ending with vowelsare also calculated using c variable and is displayed.

Step 7: Concatenate w1 and w2 to get the final output where the words which begin and end with a vowel are placed at the beginning, followed by the remaining words as they occur in the sentence. Display the sentence

Step 8: Stop

119

Source Code



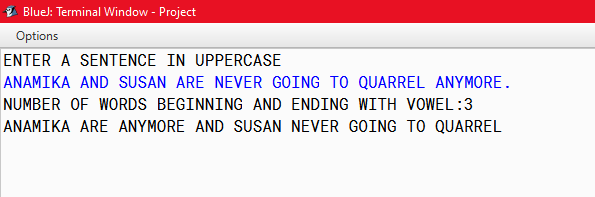
120

Variable Description

|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Description |
| sen | Int | To Enter A Sentence In Uppercase |
| F | Int | To contatenate variable w1 and w2 |
| w1 | String | To hold string values |
| w2 | String | To hold string values |
| word | String | To extract the words from a sentence |
| c | Int | Flag variable |
| ch | Char | To extract the first letter |
| ch1 | Char | To extract the last letter |
| length | int | To find the length |
| l | int | To find the length |
| ch2 | Char |  |
| s | String | To find the substring of a string |

121

OUTPUT



122