3. WRITTEN RESPONSES

3 a.

3.a.i.

The intended purpose of this program is to tell the user what any lowercase letter or full word would be in binary using an array of all stored binary lowercase letter values.

3.a.ii.

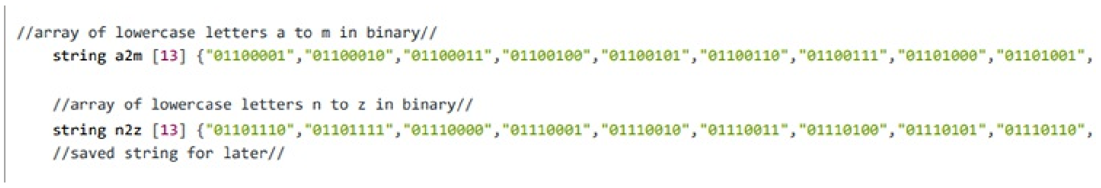
The program asks a question to get input from the user and then outputs the input in a binary format using a predefined function and a list that stores all readable inputs that can be printed in binary.

3.a.iii.

The program outputs questions to get the user to input specific data. Once the data is known and read by the program the program outputs the corresponding input in a binary format.

3 b.

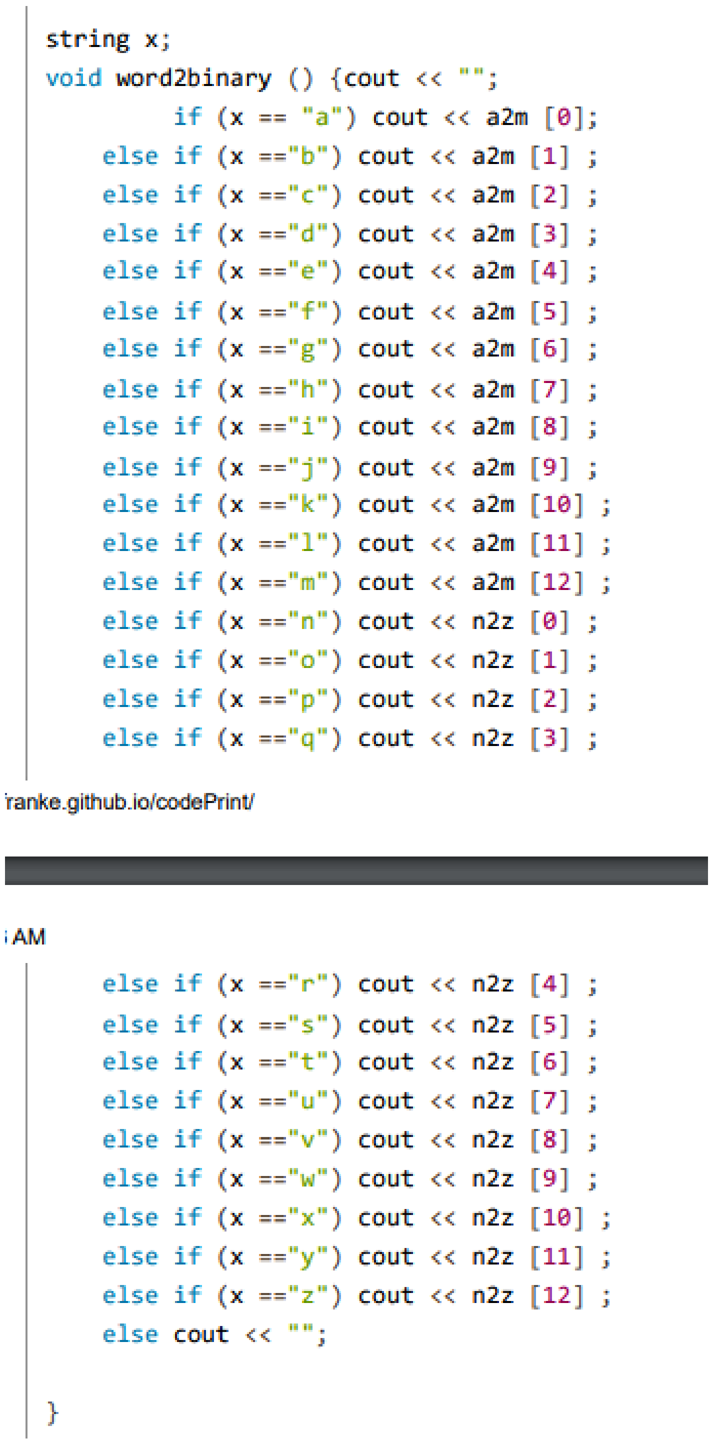
3.b.i.



```
//array of lowercase letters a to m in binary//
    string a2m [13] {"01100001","01100010","01100011","01100100","01100101","01100110","01100111","01101000","01101001",

    //array of lowercase letters n to z in binary//
    string n2z [13] {"01101110","01101111","01110000","01110001","01110010","01110011","01110100","01110101","01110110",
    //saved string for later//
```

3.b.ii.



```
string x;
void word2binary () {cout << "";
          if (x == "a") cout << a2m [0];
     else if (x =="b") cout << a2m [1] ;
     else if (x =="c") cout << a2m [2] ;
     else if (x =="d") cout << a2m [3] ;
     else if (x =="e") cout << a2m [4] ;
     else if (x =="f") cout << a2m [5] ;
     else if (x =="g") cout << a2m [6] ;
     else if (x =="h") cout << a2m [7] ;
     else if (x =="i") cout << a2m [8] ;
     else if (x =="j") cout << a2m [9] ;
     else if (x =="k") cout << a2m [10] ;
     else if (x =="l") cout << a2m [11] ;
     else if (x =="m") cout << a2m [12] ;
     else if (x =="n") cout << n2z [0] ;
     else if (x =="o") cout << n2z [1] ;
     else if (x =="p") cout << n2z [2] ;
     else if (x =="q") cout << n2z [3] ;
```

ranke.github.io/codePrint/

AM

```
     else if (x =="r") cout << n2z [4] ;
     else if (x =="s") cout << n2z [5] ;
     else if (x =="t") cout << n2z [6] ;
     else if (x =="u") cout << n2z [7] ;
     else if (x =="v") cout << n2z [8] ;
     else if (x =="w") cout << n2z [9] ;
     else if (x =="x") cout << n2z [10] ;
     else if (x =="y") cout << n2z [11] ;
     else if (x =="z") cout << n2z [12] ;
     else cout << "";


}
```

3.b.iii.

the list name is a2m and n2z

3.b.iv.

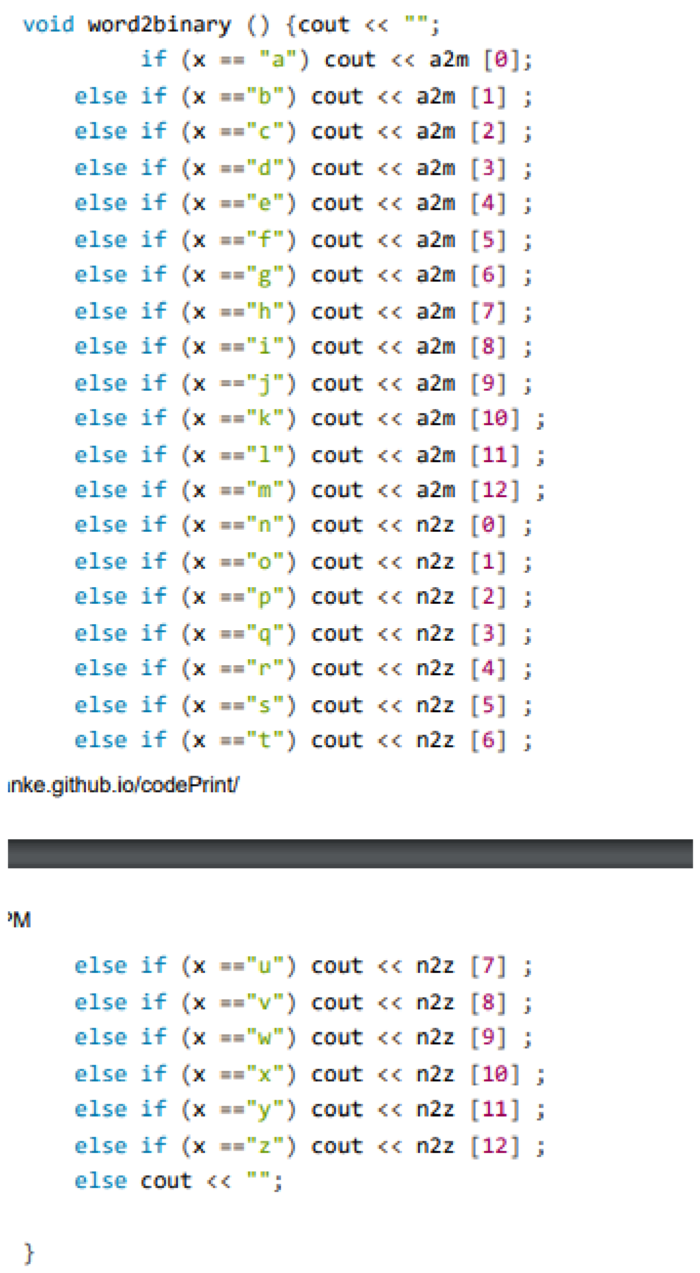
the data contained in the list a2m is all lowercase letters in binary from a to m saved in a string and the data contained in list n2z is all the lowercase letters in binary from n to z also saved as a string.

3.b.v.

the list manages complexity by storing all possible known binary lowercase letter values and only uses the ones corresponding to the inputted letter typed. If the code were to be written with no list then the code would be much longer and more confusing due to not knowing which letter equals which number on the list. With the list all of the data needed is known by the computer already and can be easily accessed and outputted by the function.

3 c.

3.c.i.



```
void word2binary () {cout << "";
         if (x == "a") cout << a2m [0];
    else if (x =="b") cout << a2m [1] ;
    else if (x =="c") cout << a2m [2] ;
    else if (x =="d") cout << a2m [3] ;
    else if (x =="e") cout << a2m [4] ;
    else if (x =="f") cout << a2m [5] ;
    else if (x =="g") cout << a2m [6] ;
    else if (x =="h") cout << a2m [7] ;
    else if (x =="i") cout << a2m [8] ;
    else if (x =="j") cout << a2m [9] ;
    else if (x =="k") cout << a2m [10] ;
    else if (x =="l") cout << a2m [11] ;
    else if (x =="m") cout << a2m [12] ;
    else if (x =="n") cout << n2z [0] ;
    else if (x =="o") cout << n2z [1] ;
    else if (x =="p") cout << n2z [2] ;
    else if (x =="q") cout << n2z [3] ;
    else if (x =="r") cout << n2z [4] ;
    else if (x =="s") cout << n2z [5] ;
    else if (x =="t") cout << n2z [6] ;
```

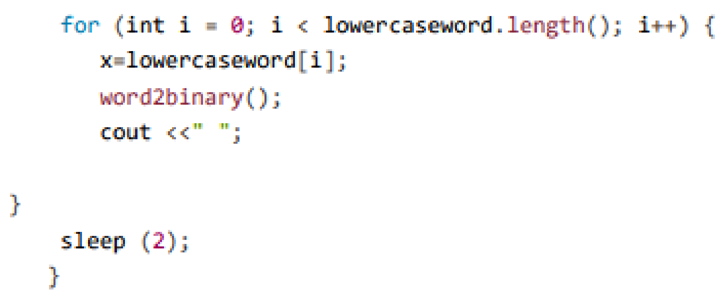
nke.github.io/codePrint/

'M

```
    else if (x =="u") cout << n2z [7] ;
    else if (x =="v") cout << n2z [8] ;
    else if (x =="w") cout << n2z [9] ;
    else if (x =="x") cout << n2z [10] ;
    else if (x =="y") cout << n2z [11] ;
    else if (x =="z") cout << n2z [12] ;
    else cout << "";

}
```

3.c.ii.



```
    for (int i = 0; i < lowercaseword.length(); i++) {
       x=lowercaseword[i];
       word2binary();
       cout <<" ";

}
    sleep (2);
   }
```

3.c.iii.

the general procedure of the program consists of reading users' input of their "favorite lowercase letter" and printing it out in binary using the list element. The procedure takes the input of the question and checks to see if it's equal to "==" any of the string values. If it is equal to a stored string value the procedure outputs the input string into binary using the list.

3.c.iv.

The algorithm works by taking the value of the user's input and outputting the corresponding array value if the comparison between the saved string and input is true, the code will print out a list element. For example, if the input "a" is equal to the string value "a" then output [0] from array a2m as 0 is the first value of the array and a is the first letter in the alphabet. It repeats this for all possible letters in the alphabet and only stops when a is == to the inputted letter by using if and else if statements to check every value until there is a match.

3 d.

3.d.i.

First call:

the first call of conditional arguments appears on line 60 as this is the first output of the function. this call to the function outputs unique to the input of the user. for example, is the user typed lowercase letter a then the function prints 0110001 but if the user were to type b the function would output 01100010. This is an argument because the code only outputs according to the letter value of the inputted word. It goes through every possible letter until it gets the one that was input by the user.

Second call:

The function gets called again in a for loop and this time reads different input data which causes different outputs to occur.
the different input data comes from the program's output of its specific questions which only accepts specific different answers forcing a different output.

3 d.ii.

Condition(s) tested by first call:

the conditions being tested are, is the input variable equal to "==" the string. If not keep going through the procedure until something matches. if nothing "else" matches than the code segment outputs nothing and continues through code.

Condition(s) tested by second call:

the conditions being tested are, "is the letter a lowercase letter" and if so which letter. differently to the first call the value of x (stored string value) changes when input changes.

3.d.iii.

Results of the first call:

There are a total of 27 possible outcomes that can come from the first call of the function 26 of them come from the input actually being true to at least one string. The one remaining output is for when the input is not equal to "==" anything, leaving the procedure to output space.

Results of the second call:

the result of the second call of the function is to output the input using variable equal to all stored and not stored possible answers. The result only changes due to user input on both calls of the function but the change in question sets up for different answers changing the result of both calls.