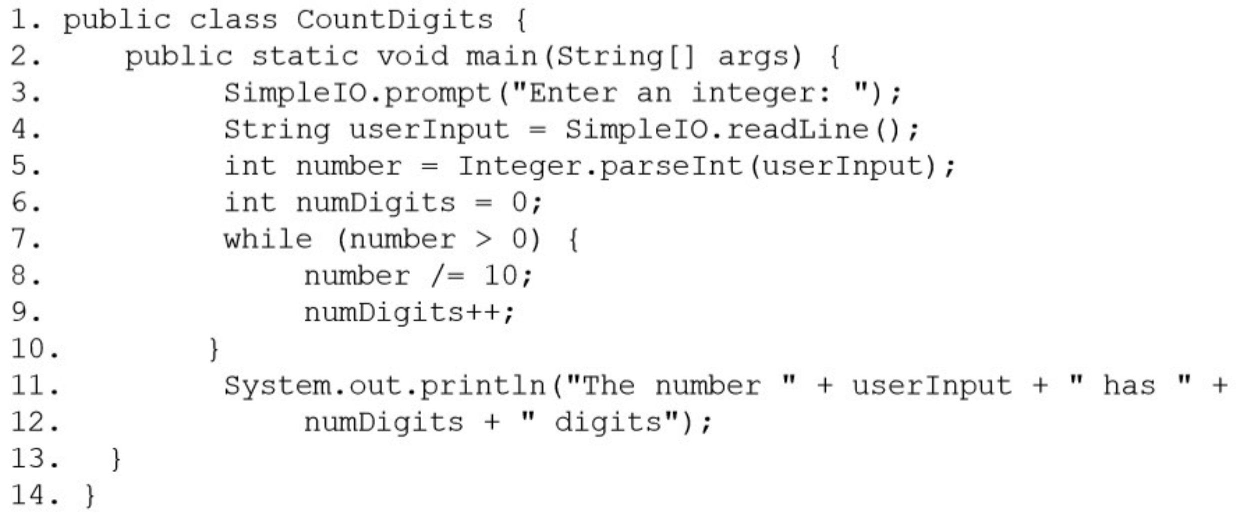
Daryna Aloff – HW1

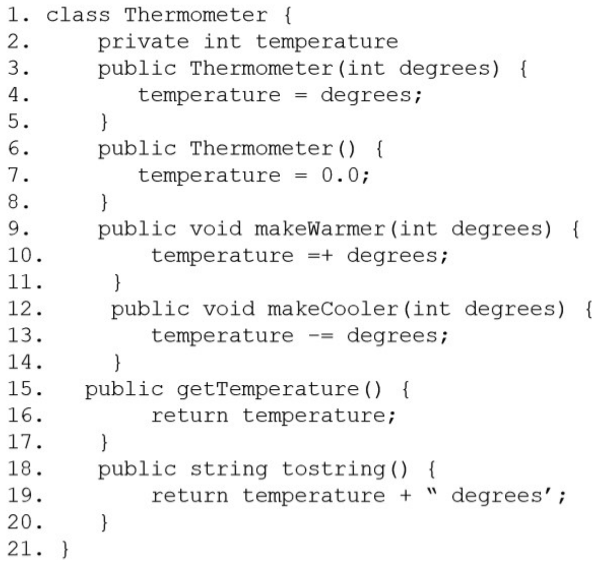
1) How many lexemes does the following Java code contain?



```
1. public class CountDigits {
2.     public static void main(String[] args) {
3.            SimpleIO.prompt("Enter an integer: ");
4.            String userInput = SimpleIO.readLine();
5.            int number = Integer.parseInt(userInput);
6.            int numDigits = 0;
7.            while (number > 0) {
8.                number /= 10;
9.                numDigits++;
10.          }
11.           System.out.println("The number " + userInput + " has " +
12.               numDigits + " digits");
13.   }
14. }
```

4 + 11 + 7 + 9 + 10 + 5 + 7 + 4 + 3 + 1 + 17 (12+5) + 1 + 1 = **80 total**

2) The following class contains several errors that violate the rules of Java:



```
1. class Thermometer {
2.     private int temperature
3.     public Thermometer(int degrees) {
4.        temperature = degrees;
5.     }
6.     public Thermometer() {
7.        temperature = 0.0;
8.     }
9.     public void makeWarmer(int degrees) {
10.        temperature =+ degrees;
11.     }
12.     public void makeCooler(int degrees) {
13.        temperature -= degrees;
14.     }
15.   public getTemperature() {
16.        return temperature;
17.    }
18.    public string tostring() {
19.        return temperature + ‟ degrees’;
20.    }
21. }
```

2. syntactic – missing ;

7. semantic – a double value cannot be assigned to int

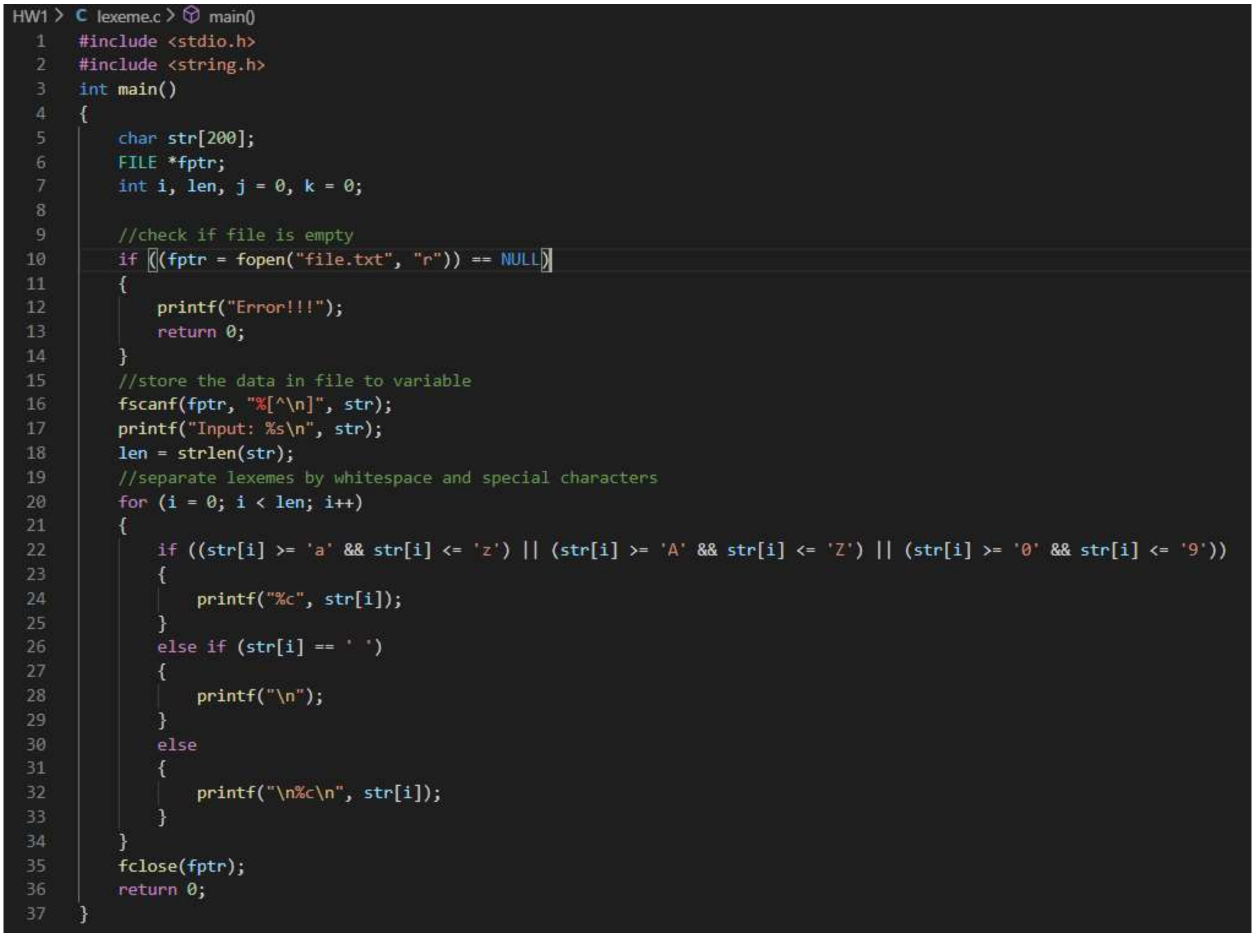
10. semantic – should be +=

15. semantic – there is no return type

18. semantic – 2 errors string ->String

19. lexical – should be double quotes

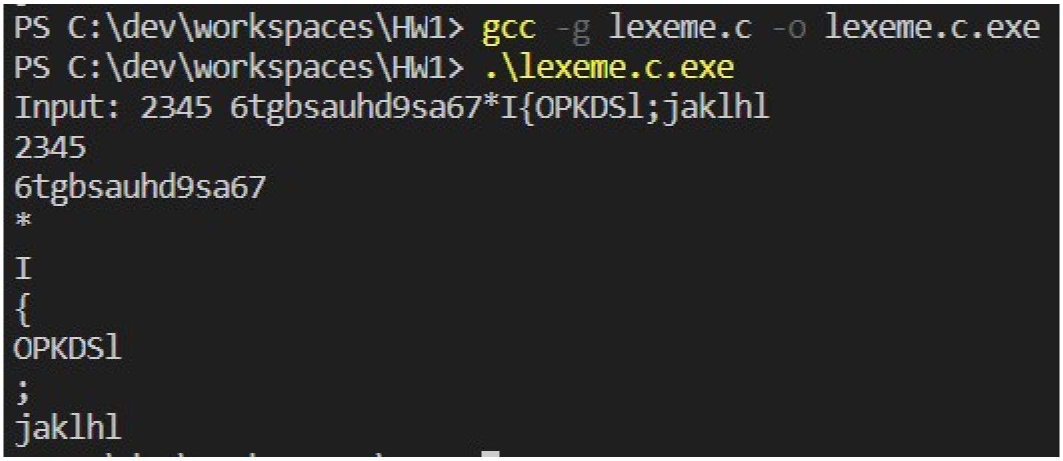
3) Lexical analyzer code in C programming language:



```c
#include <stdio.h>
#include <string.h>
int main()
{
    char str[200];
    FILE *fptr;
    int i, len, j = 0, k = 0;

    //check if file is empty
    if ((fptr = fopen("file.txt", "r")) == NULL)
    {
        printf("Error!!!");
        return 0;
    }
    //store the data in file to variable
    fscanf(fptr, "%[^\n]", str);
    printf("Input: %s\n", str);
    len = strlen(str);
    //separate lexemes by whitespace and special characters
    for (i = 0; i < len; i++)
    {
        if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= '0' && str[i] <= '9'))
        {
            printf("%c", str[i]);
        }
        else if (str[i] == ' ')
        {
            printf("\n");
        }
        else
        {
            printf("\n%c\n", str[i]);
        }
    }
    fclose(fptr);
    return 0;
}
```

Output:



```
PS C:\dev\workspaces\HW1> gcc -g lexeme.c -o lexeme.c.exe
PS C:\dev\workspaces\HW1> .\lexeme.c.exe
Input: 2345 6tgbsauhd9sa67*I{OPKDSl;jaklhl
2345
6tgbsauhd9sa67
*
I
{
OPKDSl
;
jaklhl
```