**ALGORITHM AND PROBLEM SOLVING (15B11MA301)**



**Department of Computer Science**

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

(SECTOR – 62 NOIDA, UTTAR PRADESH)

**PROJECT WORK**

**Topic: IMAGE ENCODING USING HUFFMAN**

**Submitted to:**

**MRS. SHERRY GARG**

**ALGORITHM AND PROBLEM SOLVING**

**PROJECT BASED ON**

**Huffman coding algorithm and its application**

AN ASSIGNMENT UNDER

THE SUPERVISOR

**Mrs Sherry Garg**

**Algorithm and Problem Solving (15B11MA301)**

**SUBMITTED BY:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Name** | **Enrollment no.** | **Batch** |
| Tushar Gupta | 20103307 | B10 |
| Yash Singh | 20103292 | B10 |
| Atul Sharma | 20103312 | B10 |
| Chirag Sharma | 20103302 | B10 |

**ACKNOWLEDGEMENT**

Presentation inspiration and motivation have always played a key role in the success of any venture.

We pay our deep sense of gratitude to**, Vice Chancellor Dr. S C Saxena, Jaypee Institute of Information Technology, Noida** to encourage us to the highest peak and to provide us the opportunity to prepare the project. We are immensely obliged to for their elevating inspiration, encouraging guidance and kind supervision in the completion of our project.

We feel to acknowledge our indebtedness and deep sense of gratitude to our guide **MRS. SHERRY GARG** whose invaluable guidance and kind supervision given to us throughout the course which shaped the present work as its show.

Last, but not the least, **our parents** are also an important inspiration for time. So with due regards.

Huffman coding is one of the basic compression methods that have proven useful in image and video compression standards. When applying Huffman encoding technique on an Image, the source symbols can be either pixel intensities of the Image, or the output of an intensity mapping function.

The first step of Huffman coding technique is to reduce the input image to a ordered histogram, where the probability of occurrence of a certain pixel intensity value is as

prob\_pixel = numpix/totalnum

where **numpix** is the number of occurrence of a pixel with a certain intensity value and **totalnum** is the total number of pixels in the input Image.

**BRIEF ALGORITHM**

Step 1 :

Read the Image into a 2D array(image)

Create a Histogram of the pixel intensity values present in the Image.

Find the number of pixel intensity values having non-zero probability of occurrence.

Step 2:

Define a struct which will contain the pixel intensity values(pix), their corresponding probabilities(freq), the pointer to the left(\*left) and right(\*right) child nodes and also the string array for the Huffman code word(code).

These structs is defined inside main(), so as to use the maximum length of code(maxcodelen) to declare the code array field of the struct pixfreq

Step 3:

Define another Struct which will contain the pixel intensity values(pix), their corresponding probabilities(freq) and an additional field, which will be used for storing the position of new generated nodes(arrloc).

Step 4:

Declaring an array of structs. Each element of the array corresponds to a node in the Huffman Tree.

Step 5:

Initialize the two arrays pix\_freq and huffcodes with information of the leaf nodes.

Step 6:

Sorting the huffcodes array according to the probability of occurrence of the pixel intensity values.

Step 7:

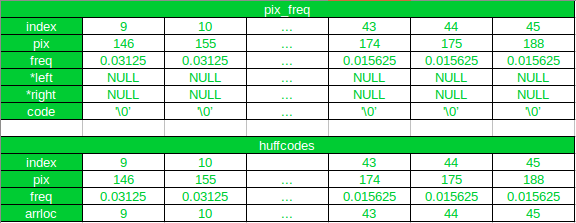
Building the Huffman Tree.

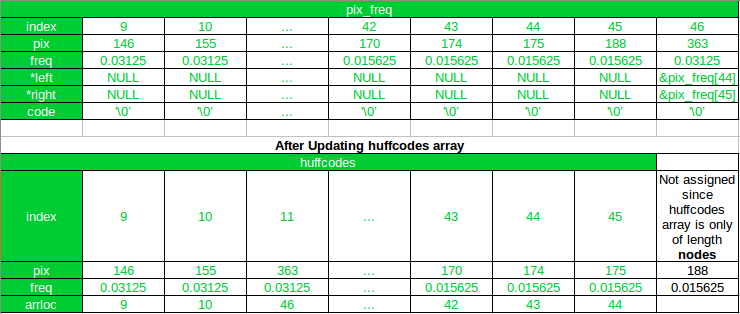
Backtrack from the root to the leaf nodes to assign code words

Step 8:

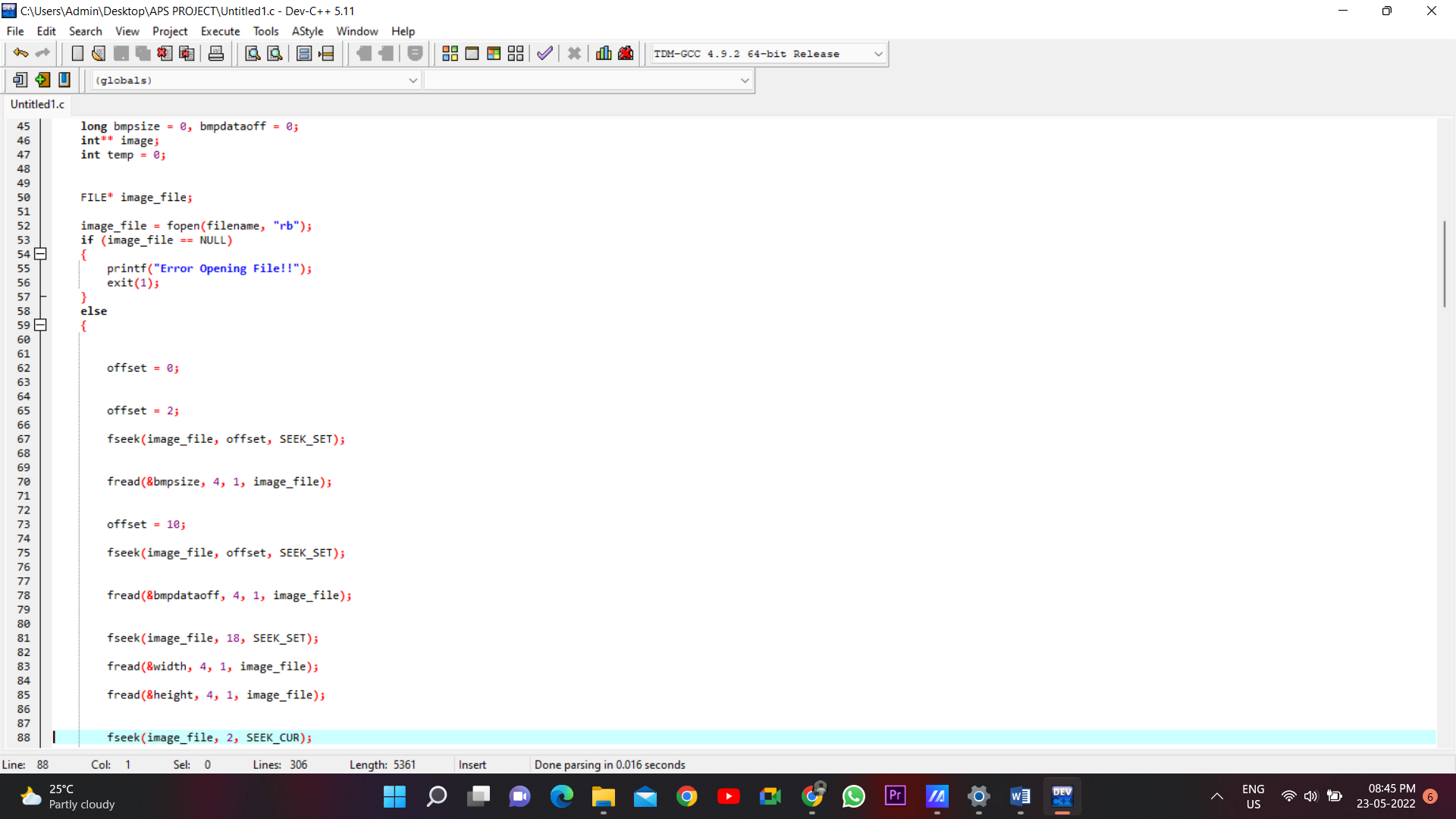
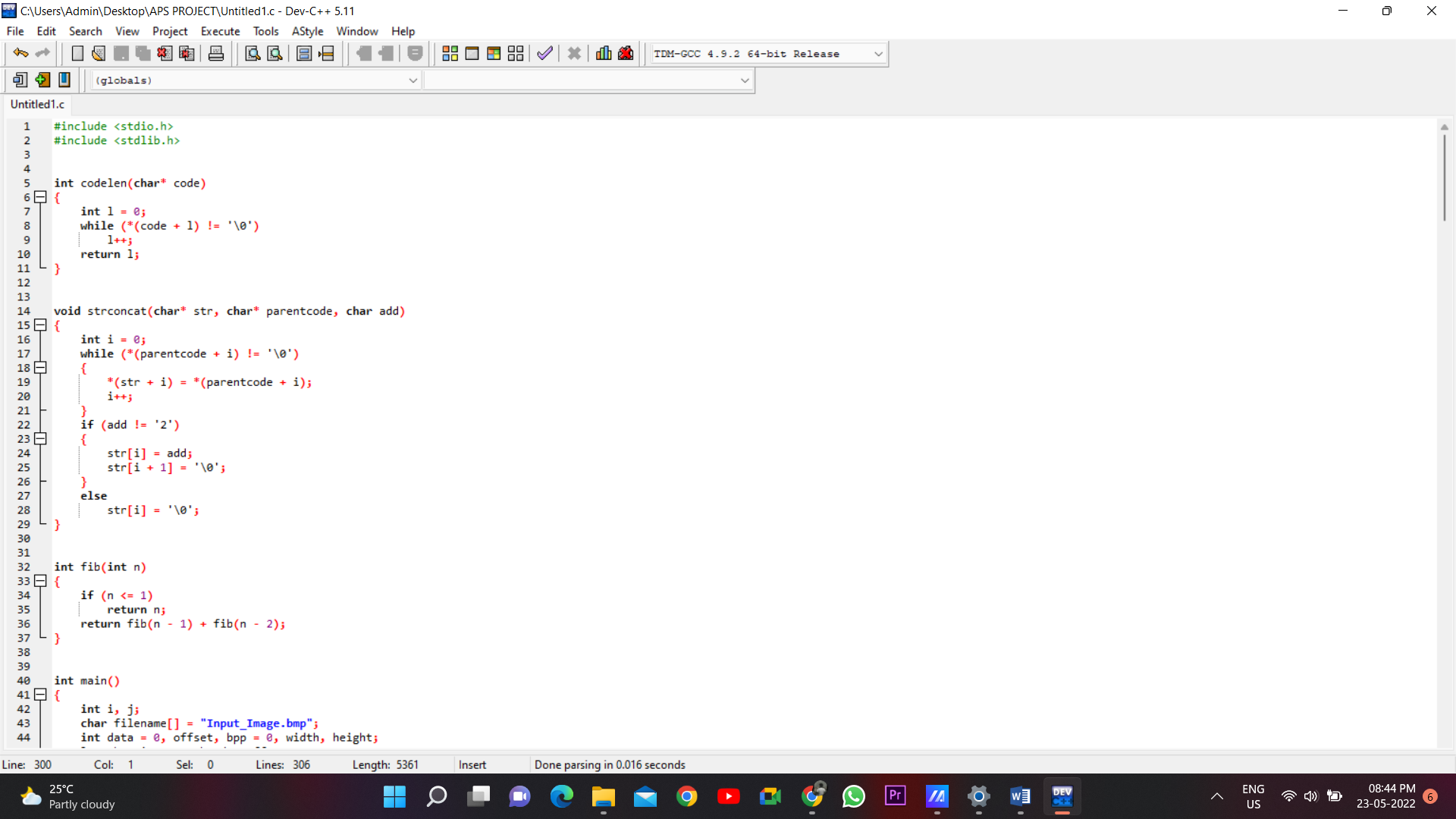
Encode the Image

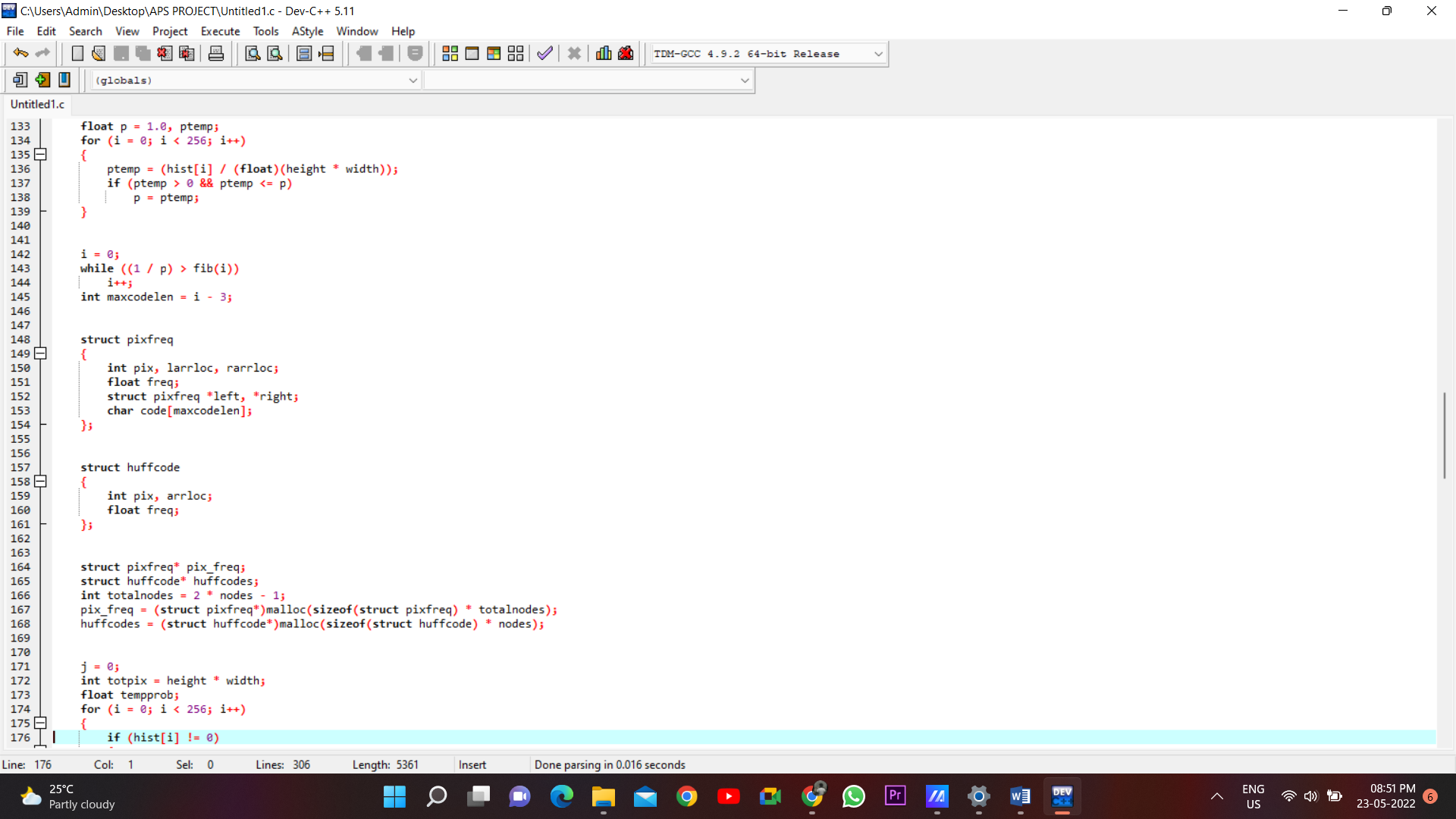
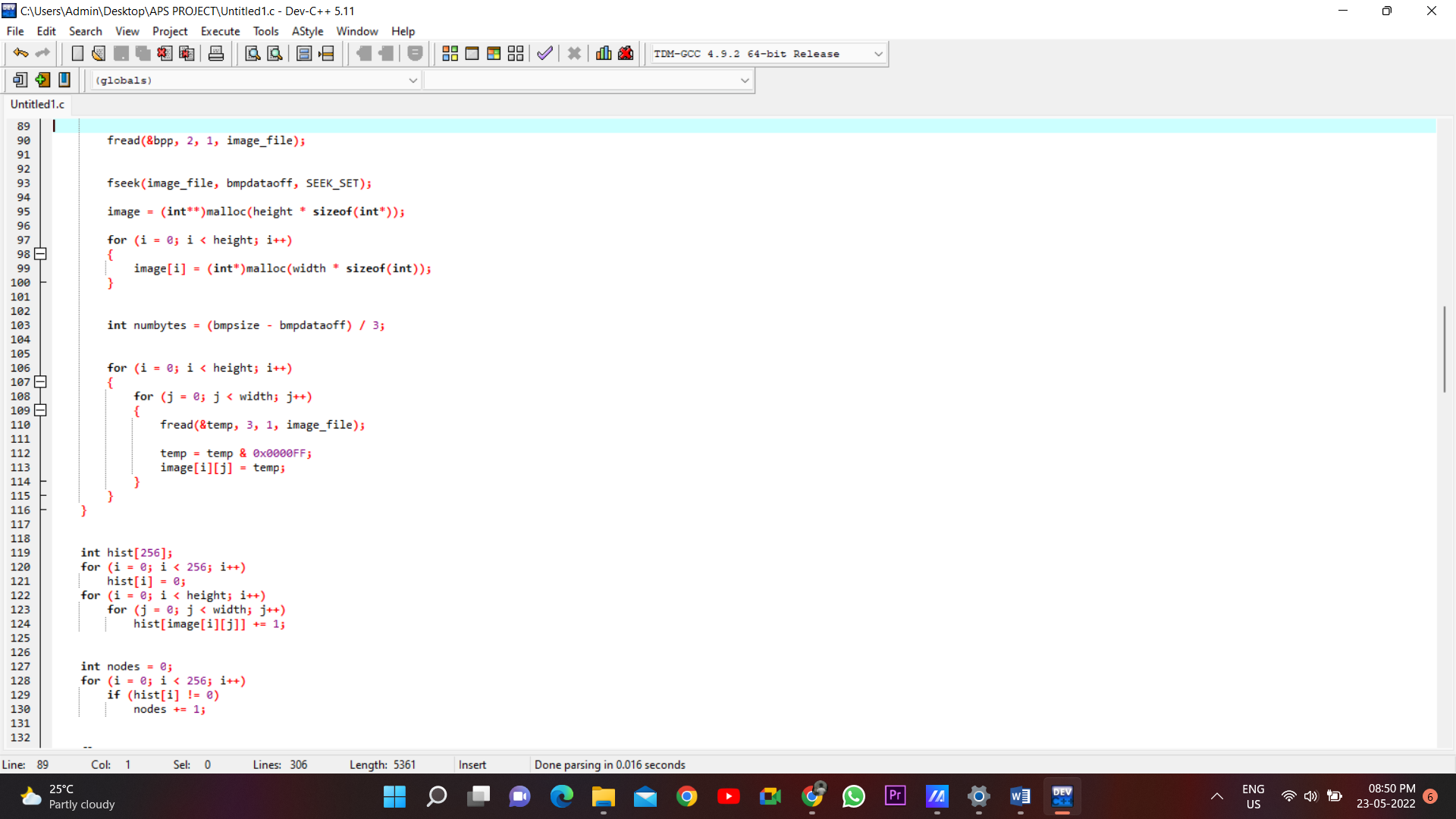
Average number of bits required to represent each pixel.

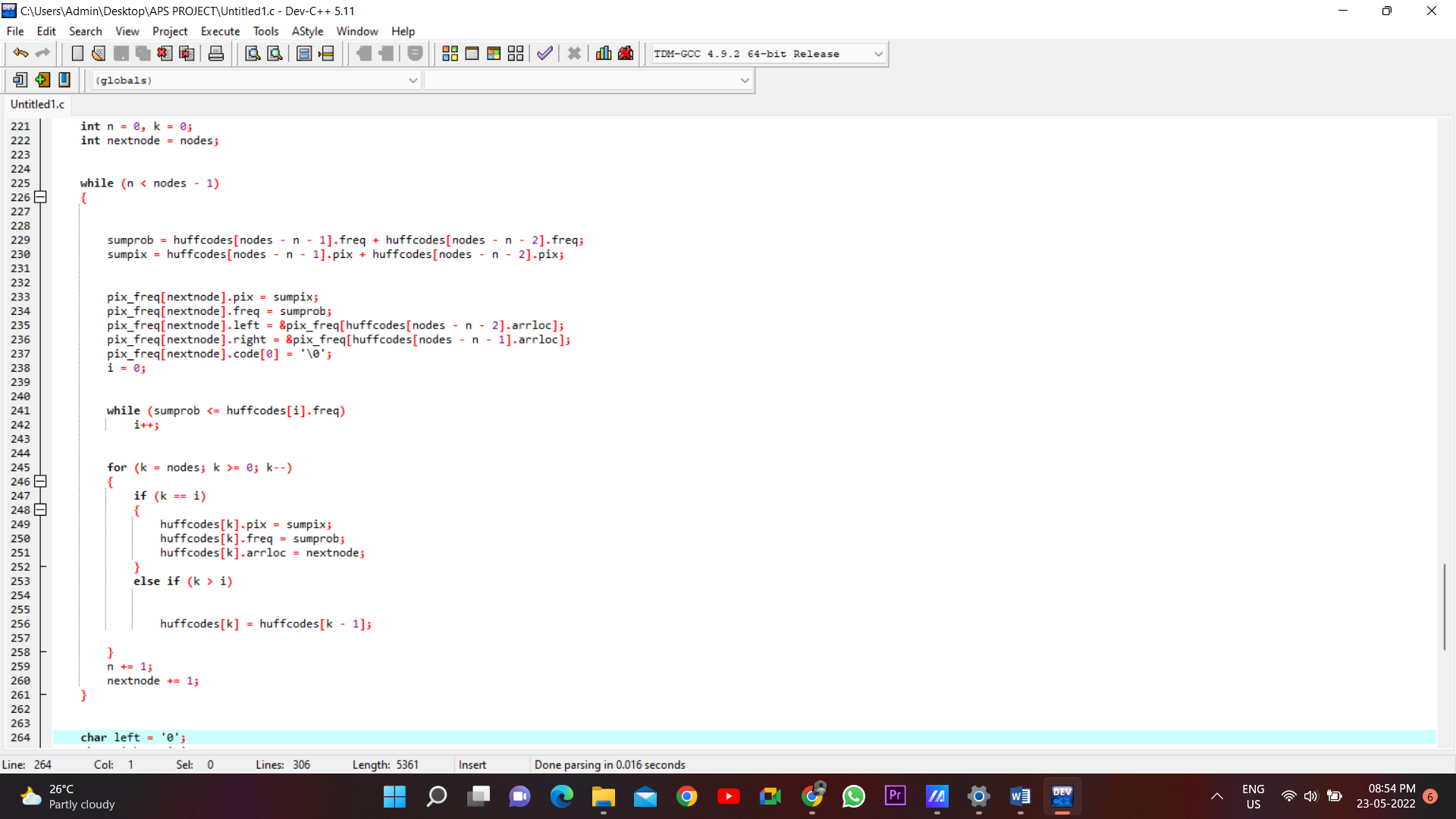
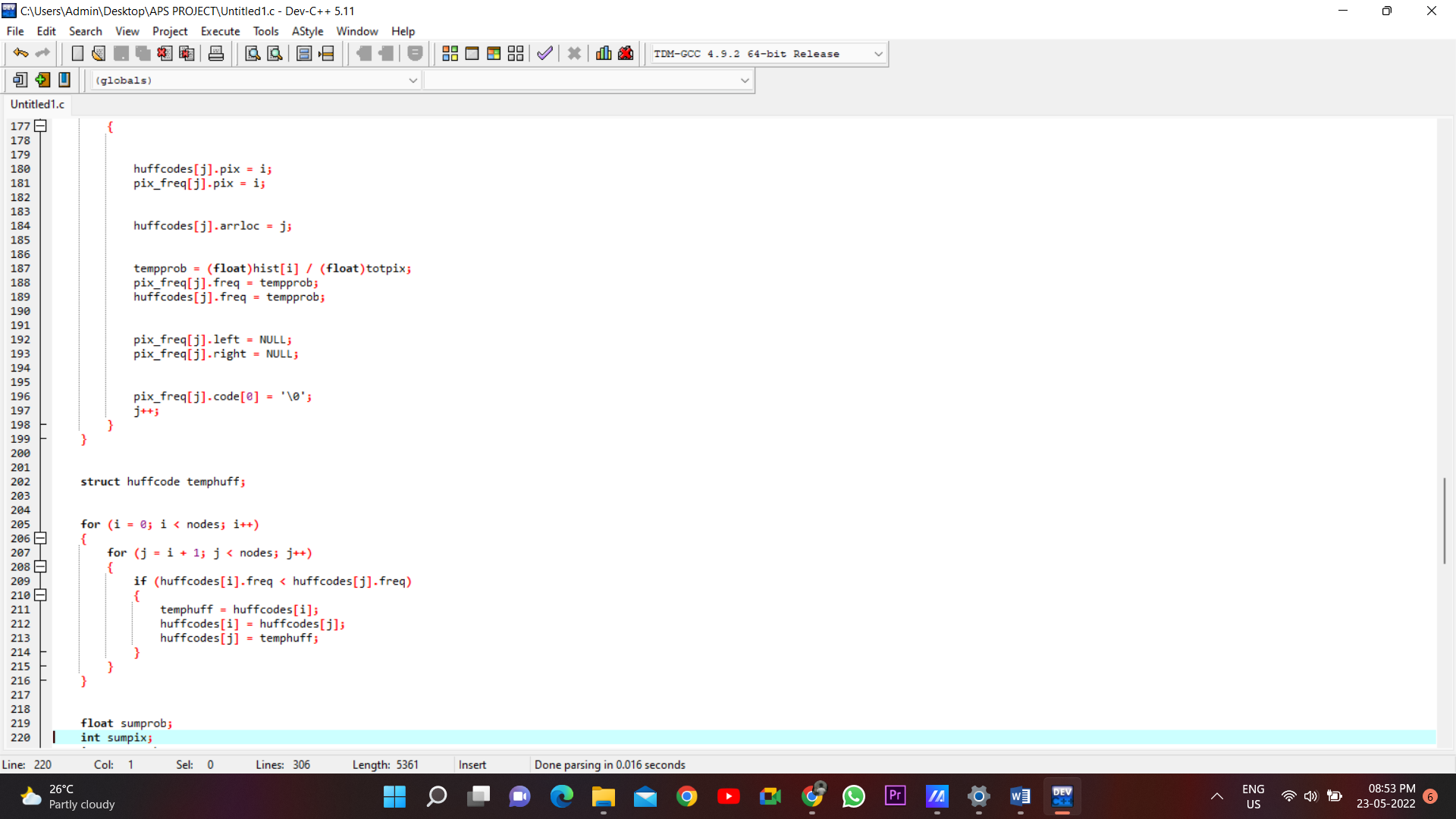


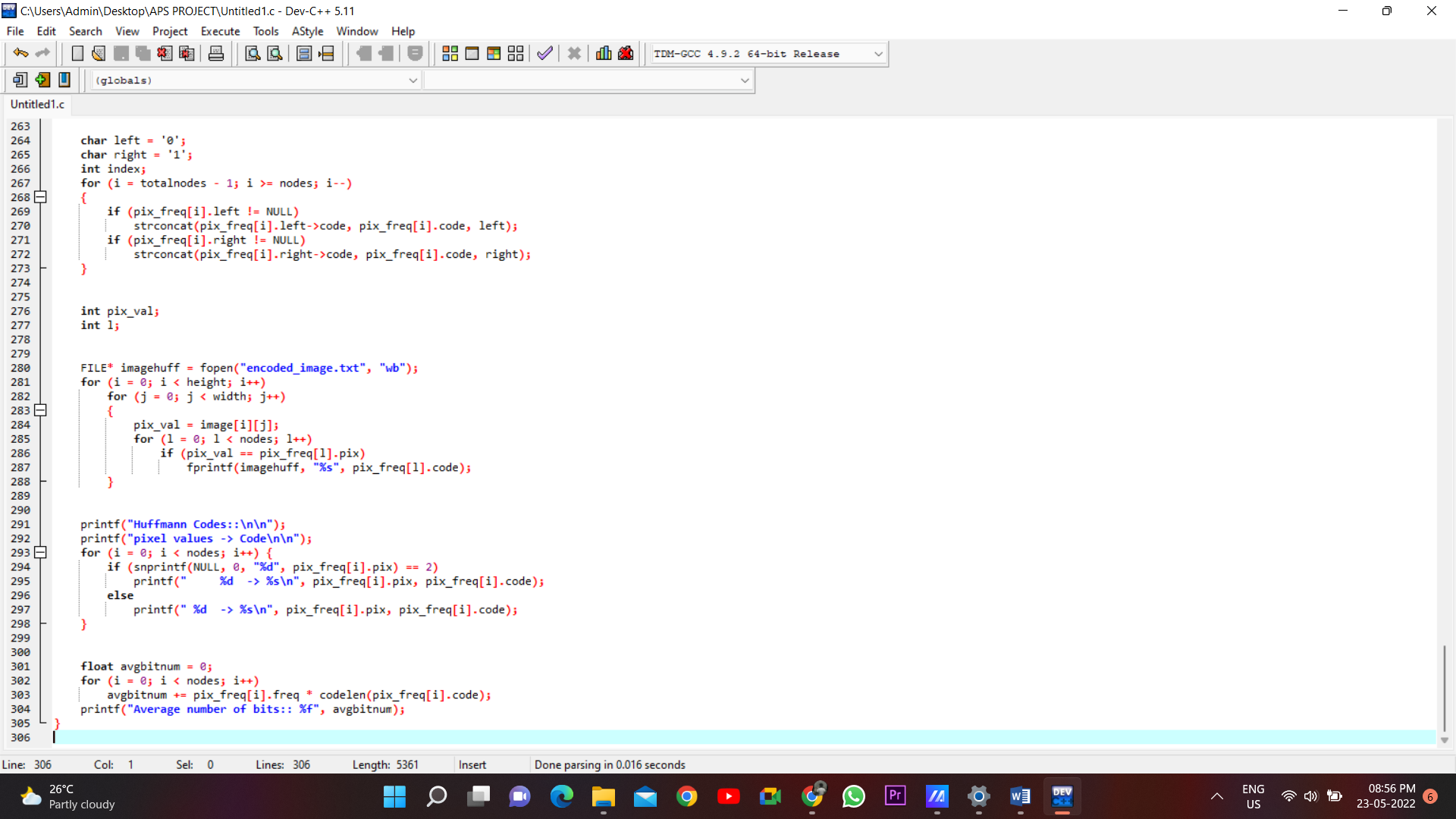


CODE:





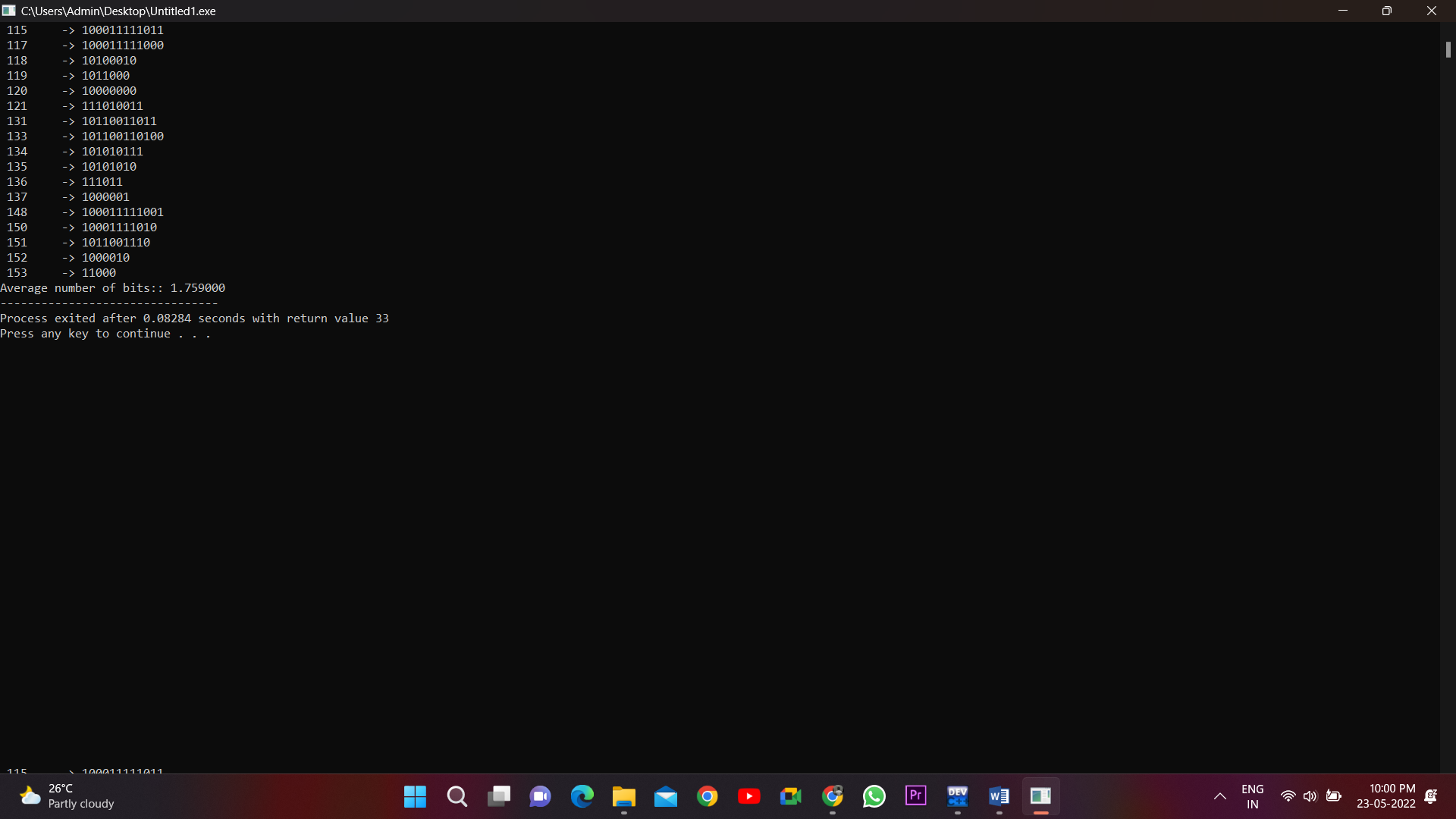
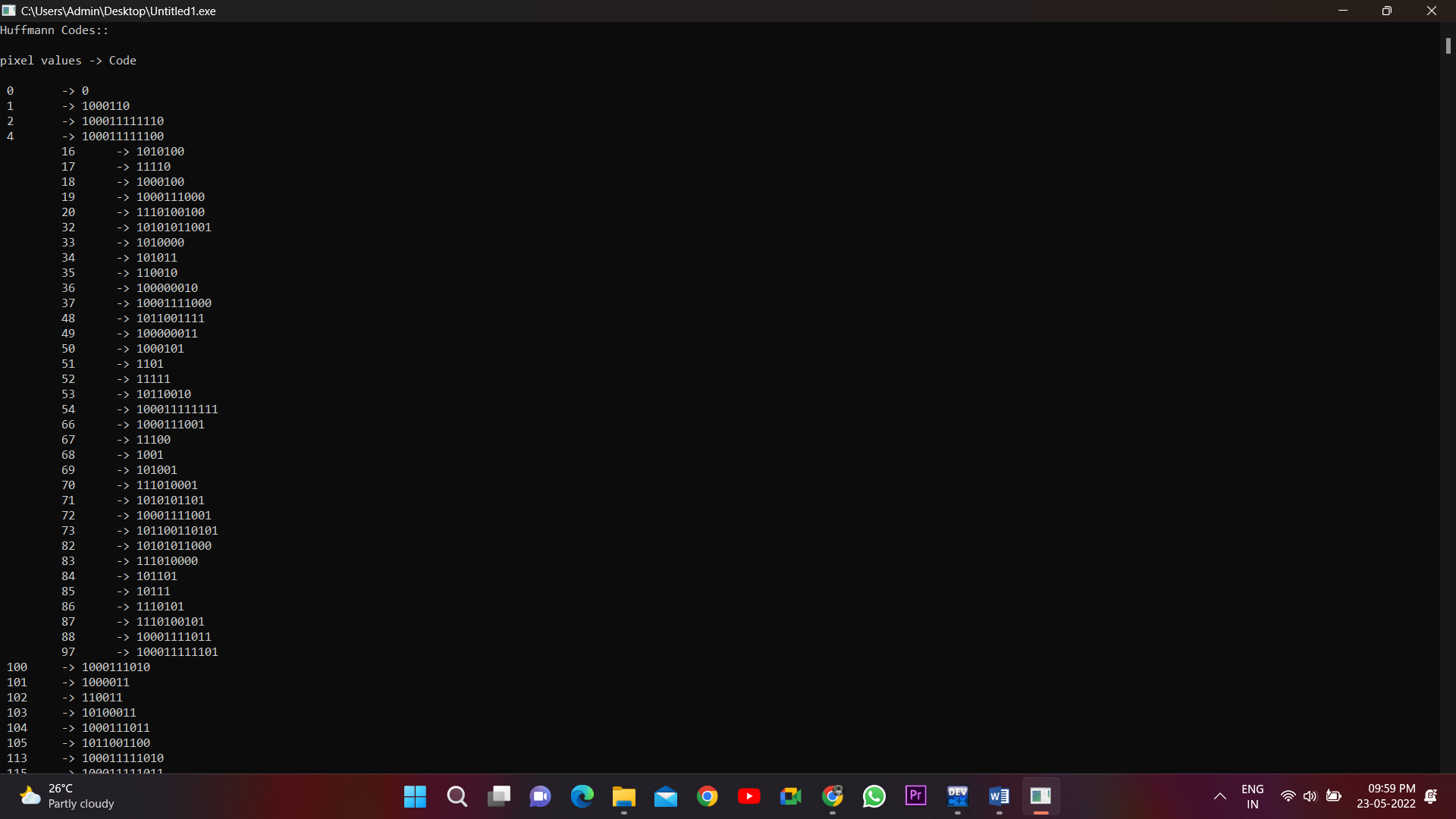




**SAMPLE IMAGE 1:**

****

OUTPUT:



**SAMPLE IMAGE 2:**

****

**OUTPUT:**

