Computer Project #4

This assignment focuses on the design, implementation and testing of a Python program which is all about numbers and images representation and converting or computing with them.
It is worth 40 points (4% of course grade) and must be completed no later than **11:59 PM on Monday, February 13** (two weeks because of the first exam).
After the due date, your score will be deducted by 1point for every 1 hour late or fraction of it for up to 24 hours. **Note that you must click on the "Submit" button to stop the counter. Otherwise, it will not be considered as submitted.**

Assignment Overview

In this assignment, you will practice with functions, strings, controls. **You are not allowed to use any build-in function that converts to/from binary representation. You are not allowed to use any advanced data structure (e.g., Lists, Dictionaries, Classes, etc.). Using any of these will result in a zero.**

Assignment Background

Number representation
When working with any kind of digital electronics in which numbers are being represented, it is important to understand how numbers are represented in these systems.
Human beings use decimal (base 10) for counting and measurements. Computers use binary (base 2) number system because they are made from electronics operating in two states - on and off (only two possible digits 0 or 1). In computing, we also use hexadecimal (base 16) or octal (base 8) systems. You should all have some familiarity with the decimal system. For instance, to represent the integer 123 as a decimal number, we can write:

$$123= 1\times100 + 2\times10 + 3\times1 = 1\times10^2 + 2\times10^1 + 3\times10^0$$

Binary number system has two symbols: 0 and 1, called bits. Eight bits is called a byte. It is also a positional notation, for example,

$$10101100 = 1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 0\times2^0$$

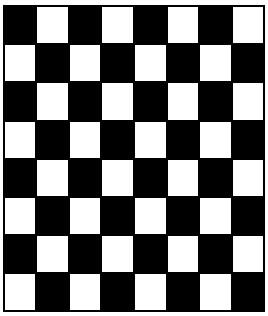
In a binary number, the right digit is considered the lowest digit whereas the leftmost is considered the highest digit. Due to the positional notation, the **L**east **S**ignificant **B**it (LSB) is also known as the rightmost bit. It is the opposite of the **M**ost **S**ignificant **B**it (MSB), which carries the highest value in a multiple-bit binary number as well as the number which is farthest to the right. For example, in the binary number `110110,` the LSB is `0` and the MSB is `1.`

'10101100'

String representation
Computers use numeric encodings to represent character data (strings) inside the memory of the machine, in which each character is assigned an integral value. Character codes, however, are not very useful unless they are standardized. The widely adopted character encoding is Unicode which is a superset of the ASCII code. So each character is represented by a numerical value and stored as a binary number. Below are a couple of examples of characters and their corresponding decimal number and binary number. You can find the decimal code for any character by using the `ord()` function in python. You can look at the "ASCII printable characters" table at **http://www.ascii-code.com/** for the character-decimal value conversions. Below is some examples of ASCII character table and decimal equivalent along with their binary representation.

Black and white images representation

In this section, we will explore the representation of images using 0's and 1's. Let's begin by considering just 8-by-8 black-and-white images such as the one below:



Each cell in the image is called a "pixel". A black pixel is represented by the digit `1` and a white pixel is represented by the digit `0`. The first digit represents the pixel at the top left corner of the image. The next digit represents the pixel in the top row and the second column. The eighth bit (digit) represents the pixel at the right end of the top row. The next bit represents the leftmost pixel in the second row and so forth. Therefore, the image above is represented by the following binary string of length 64:

```
'1010101001010101101010100101010110101010010101011010101001010101'
```

Color images representation

In this section, we will explore the representation of color images using 0's and 1's. Let's begin by considering just 6-by-6 colored images such as the one below. In an image that uses 4 colors, 2 bits are needed for each pixel. The following example uses two bits to store the following colors:

```
00 – White; 01 – Black; 10 – Yellow; 11 – Blue
```

10	10	10	10	10	10
10	00	10	10	00	10
10	11	10	10	11	10
10	10	10	10	10	10
10	01	01	01	01	10
10	10	10	10	10	10

The image above is represented by the following binary string of length 72:

```
'101010101010100010100010101110101110101010101010010101011010101010'
```

The first two digits represents the pixel at the top left corner of the image. The next two digits represents the pixel in the top row and the second column. The sixth two bits (digits) represents the pixel at the right end of the top row.

Embedding Messages in Images

Steganography is the art of concealing a message, image, or file within another message, image, or file. Least significant bit (LSB) insertion is a common and simple approach to embedding message bits in image pixels. The algorithm of LSB to embed text message in an image is as follows (note that we modified the algorithm to make it work with our project, an example is provided in the assignment specifications):

- Step1: Read the image and text message, which is to be hidden in the image.
- Step 2: Convert text message into binary.
- Step 3: Calculate LSBs of image pixels.
- Step 4: Replace LSB of image pixels with each bit of secret message one by one.

Assignment Specifications

Tatooine planet is under attack from stormtroopers, and there is only one line of defense remaining. You have been hired by TASA (Tatooine Air and Space Administration). TASA has a deep-space satellite that takes 8-by-8 black-and-white images and sends them back to Tatooine as binary strings described above. To save the planet, the freedom fighters needs to be able to embed messages into those image.

You will develop a Python program which allows the user to select from a menu of options and which performs the requested calculations. You must use at least the four functions specified below. You are encouraged to use more functions of your own design. Global variables are not allowed. That is, any variable names used within any function must be parameters or be created in the function unless those variables are constants. You are not allowed to use advanced data structures such as list, dictionaries, classes, etc. However, you are allowed to read ahead and use try-except. You are also not allowed to use built-in functions that are not specified in the function descriptions below:

`numtobase(N, B)---→ str:`

a. This function accepts as input a non-negative integer `N` and a base `B` (no error checking); it should return a string representing the number N in base B. The string should always be of length multiple of 8 unless the number is 0. Your function must output the empty string when the input value of N is 0. (This avoids leading zeros!)

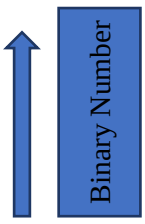
a. Parameters: `N (int), B (int)`
b. Returns : `str`
c. The function displays nothing.
d. Steps to convert decimal to other base system:

Step 1 − Divide the decimal number to be converted by the value of the new base.
Step 2 − Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
Step 3 − Divide the quotient of the previous divide by the new base.
Step 4 − Record the remainder from Step 3 as the next digit (to the left) of the new base number.
Step 5 − Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3. The last remainder thus obtained will be the Most Significant Bit (MSB) of the new base number.

Example on how to convert from decimal to binary base system:

```
Decimal Number: 29
Calculating Binary Equivalent –
```

Step	Operation	Quotient	Remainder
Step 1	29 / 2	14	1
Step 2	14 / 2	7	0
Step 3	7 / 2	3	1
Step 4	3 / 2	1	1
Step 5	1 / 2	0	1




```
Decimal Number – 29 = Binary Number – 00011101
```

As you can see, the remainders have to be arranged in the reverse order so that the first remainder becomes the Least Significant Bit (LSB) and the last remainder becomes the Most Significant Bit (MSB).

Now ask yourself... what has to change in order to output the number in base B instead of base 2?

```
In [1]: numtobase(29, 2)
Out[1]: '00011101'
```

```
In [2]: numtobase(4, 4)
Out[2]: '00000010'

In [3]: numtobase(0, 2)
Out[3]: '' # notice the empty string for an input N of 0 !
```

basetonum (S, B) ---→ int:

a. This function accepts as input a string S and a base B where S represents a number in base B where B is between 2 and 10 inclusive. It should then return an integer in base 10 representing the same number as S. It should output 0 when S is the empty string.

b. Parameters: `S (string), B (int)`
c. Returns : `integer`
d. The function displays nothing.
e. Steps to convert any base system to decimal:
Step 1 − Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
Step 2 − Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
Step 3 − Sum the products calculated in Step 2. The total is the equivalent value in decimal.

Example
`Binary Number – 00011101`
`Calculating Decimal Equivalent –`

Step	Binary Number	Decimal Number
Step 1	00011101	$(0\times2^7) + (0\times2^6) + (0\times2^5) (1\times2^4) + (1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0)$
Step 2	00011101	0 + 0 + 0 + 16 + 8 + 4 + 0 + 1
Step 3	00011101	29

`Binary Number – 00011101 = Decimal Number – 29`

Again, the key is to ask yourself... what has to change in order to output base B instead of base 2?

```
In [1]: basetonum('00011101', 2)
 Out[1]: 29
In [2]: basetonum('', 4)
 Out[2]: 0
In [3]: basetonum('00004321', 5)
 Out[3]: 586
```

basetobase(B1,B2,S_B1)---→ str:

a. Now, we can assemble what we've written to write a function that takes three inputs: a base `B1`, a base `B2` and `S_B1`, which is a string representing a number in base `B1`. Then, your function should return a string representing the same number in base `B2`. `S_B1` is always of length multiple of 8.
b. Parameters: `B1(int), B2(int), S_B1(str)`
c. Returns : `string`
d. The function displays nothing.
e. Don't rewrite any conversions at all! Instead, convert to decimal and then back to the desired final base!

```
In [1]: basetobase(2, 10, "00000011") # 11 in base 2 is 3 in base 10
Out[1]: '00000003'
In [2]: basetobase(10, 2, "00000003") # 3 in base 10 is 11 in base 2
Out[2]: '00000011'
In [3]: basetobase(3, 5, "00000011") # 11 in base 3 is 4 in base 5
Out[3]: '00000004'
In [4]: basetobase( 2, 3, '00101010' )
Out[4]: '00001120'
In [5]: basetobase( 2, 4, '00101010' )
Out[5]: '00000222'
In [6]: basetobase( 2, 10, '00101010' )
Out[6]: '00000042'
In [7]: basetobase( 5, 2, '00004321' )
Out[7]: '0000001001001010'
In [8]: basetobase( 2, 5, '0000001001001010' )
Out[8]: '00004321'
```

`encode_image(image,text,N)---→ string or None:`

a. This function takes a binary string `image` representing the image, `text` representing the message to be hidden in the image and `N` representing how many bits represent each pixel, and returns another binary string as output. The output binary string should be the original image with the text embedded using the LSB algorithm defined earlier. If `image` is empty, the function should return an empty string. If `text` is empty, the `image` should not change. If the image is not big enough to hold all the text, the function should return `None`.

b. You may need a helper function or two - you may name these whatever you like. Also, you may want to use some of the functions previously defined.

c. Parameters: `image(str), text(str), N(int)`

d. Returns : `string or None`

e. The function displays nothing.

Let's begin by considering a 5-by-5 colored image. The image is represented by the following binary string of length 50:

```
image = '10101010101010001010001010111010111010101010101010'
```

This image uses two bits to store the colors: `N = 2`

Let's say we want to embed the text `'CS'` in the image above:

```
text = 'CS'
```

Using the above algorithm:

- Step 2: Convert text message into binary. You need to convert each character in the text to binary string then concatenate all binary strings. To convert each character to binary, you need to first convert the character to its decimal number (Hint: `ord()`) then from decimal to binary using the function defined in this project.

  `'CS' -----→ '0100001101010011'`

  `'C' -----→ 67 -----→ '01000011'`

  `'S' -----→ 83 -----→ '01010011'`

  `'CS' -----→ '01000011' + '01010011' -----→ '0100001101010011'`

- Step 3: Calculate LSBs of image pixels. The LSB for each character in the binary string is the rightmost bit of each pixel (position `N` of each pixel shown in red in the below example):

```
'1010101010101000101000101011101011101010101010'
```

- Step 4: Replace LSB of image pixels (colored in red in the original image) with each bit of secret message one by one (colored in green in the encoded image). If the LSB of the image pixel is the same as the bit of the secret message do not change it (colored in blue in the encoded image):

```
text = '0100001101010011'

Original message:
'1010101010101000101000101011101011101010101010'
Encoded message:
'1011101010101101101100111010111111010101010101010'
  ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
  0 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1
```

Here are a couple of examples of `encode_image()` in action:

```
In [1]: encode_image( image, text, N )
Out[1]: '1011101010101101101100111010111111010101010101010'
In [2]: encode_image( '',text,N )
Out[2]: ''
In [3]: encode_image( image, '', N )
Out[3]: '1010101010101000101000101011101011101010101010101010'
In [4]: encode_image( '101010101010', text, N )
Out[4]:
```

`decode_image(stego,N)---→ string:`

a. This function takes a binary string and `N` representing how many bits represent each pixel as input and returns a string as output, which is the hidden text. The function "inverts" or "undoes" the encoding in your `encode_text()` function. That is, `decode_image(encode_image(image,text,N),N)` should give back `text` and some more characters as gibberish.

b. You may need a helper function or two. Also, you may want to use some of the functions previously defined.

c. Parameters: `stego(str),N(int)`

d. Returns : `string`

e. The function displays nothing.

f. Hints: Remember that each character is represented by an 8 bit binary number (i.e., `"A"` --→ `"0100000"`). So the length of the binary string that represents the text should always be a multiple of 8 (e.g., 0, 8, 16, …).

Here are a couple of examples of `decode_image` () in action:

```
stego = '1011101010101101101100111010111111011101010111011100101010110101010101010'

In [2]: text_out = decode_image(stego,2)
Out[2]: 'CSEx'
```

`main():`

a. This function is used to interact with the user. It takes no input and returns nothing. Call the functions from here. The program should prompt the user to choose between 6 options (capitalization does not matter) until the user enters `'X'`:
   - `'A'` - Convert a decimal number to another base system.
   - `'B'` – Convert decimal number from another base.
   - `'C'` – Convert from one representation system to another.
   - `'E'` - Encode an image with a text.
   - `'D'` - Decode an image.
   - `'M'` - Display the menu of options.
   - `'X'` - Exit from the program.

b. The program will repeatedly prompt the user to enter a letter (upper or lower case) which corresponds to one of the supported options. For example, the user will enter the letter A (or the letter a) to select Option A (Convert a decimal number in another base system).

c. The program will display the menu of options once when execution begins, whenever the user selects Option `M`, and whenever the user selects an invalid menu option.

d. If the user enters option `A`, the program will ask the user to enter a numeric value N and a base number B in the range between 2 and 10 inclusive. If N is a non-negative number and B is valid (an integer between 2 and 10), the program will calculate and display the string representation of the number in base B. Otherwise, the program will display an appropriate message and reprompt to enter the invalid input. Note that the string method `.isdigit()` is useful here.

e. If the user enters option `B`, the program will ask the user to enter a string S and a base number B in the range between 2 and 10 inclusive. If B is valid, it will calculate and display the decimal representation of the string. Otherwise, the program will display an appropriate message and reprompt to enter the invalid input

f. If the user enters option `C`, the program will ask the user to enter three inputs: a base `B1`, a base `B2` (both of which are between 2 and 10, inclusive) and `S_B1`, which is a string representing a number in base `B1`. If `B1` and `B2` are valid, it will calculate and display a string representing `S_B1` in base `B2`. Otherwise, the program will display an appropriate message and reprompt to enter the invalid input.

g. If the user enters option `E`, the program will ask the user to enter: (1) a binary string for the image, (2) then, a text to hide in the image, and (3) finally, a number of bits representing the pixels. The program will then encode the image and display the new encoded string of the image.

h. If the user enters option `D`, the program will ask the user to enter: (1) a binary string representing an encoded image, and (2) a number of bits representing the pixels. The program will then decode the image and displays the hidden text.

i. If the user enters option `M`, the program will display the menu.

j. If the user enters `X`, the user wants to quit. The program should display a goodbye message.

Assignment Notes and Hints

1. The coding standard for CSE 231 is posted on the course website:

   http://www.cse.msu.edu/~cse231/General/coding.standard.html

Items 1-9 of the Coding Standard will be enforced for this project.

2. The program will produce reasonable and readable output, with appropriate labels for all values displayed.

3. You may assume that the user enters a string representing a numeric value when prompted for a number. However, you should not assume that the value is valid in that particular context. For example, the user might enter '-5' for Option A; that numeric value is not a non-negative number.

4. Be sure to prompt the user for the inputs in the correct order. And, your program cannot prompt the user for any supplementary inputs.

5. Several built-in function might be useful for this project:

```
      str() – to convert a number to a string
      int() – to convert a string to an int
      .isdigit() – to check if the string is only digits
      ord() – to get the ASCII code of a character. Convert character to a number.
      chr() – to get the character that corresponds to the ASCII code
      zfill() – to add zeros (0) at the beginning of the string, until it reaches the
specified length.
```

6. We provide a `proj04.py` program for you to start with.

7. You are not allowed to use advanced data structures such as list, dictionaries, classes, etc. However, you are allowed to read ahead and use try-except.

8. If you "hard code" answers, you will receive a grade of zero for the whole project. An example of hard coding is to simply print the correct value of the binary number rather than calculating it and then printing it.

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Cycle through the following steps to incrementally develop your program:
  - Edit your program to add new capabilities.
  - Run the program on Spyder and fix any errors.
  - Use the **Coding Rooms** system to submit the current version of your program.

- Be sure to use the **Coding Rooms** system to submit the final version of your program.

- Be sure to log out when you leave the room, if you're working in a public lab.

The last version of your solution is the program which will be graded by your TA.
*You should use the **Coding Rooms** system to back up your partial solutions, especially if you are working close to the project deadline. That is the easiest way to ensure that you won't lose significant portions of your work if your machine fails or there are other last-minute problems.*

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj04.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Coding Rooms system** before the project deadline.

Grading Rubric

```
Computer Project #04                                  Scoring Summary
General Requirements:
  ( 4 pts) Coding Standard 1-9
      (descriptive comments, function headers, mnemonic identifiers, format,
      etc...)

Implementation:
 ( 4 pts)  numtobase() function

 ( 3 pts)  basetonum() function

 ( 2 pts)  basetobase() function

 ( 4 pts)  encode_image() function

 ( 4 pts)  decode_image() function

 ( 2 pts)  Test 1

 ( 2 pts)  Test 2

 ( 2 pts)  Test 3

 ( 3 pts)  Test 4

 ( 3 pts)  Test 5

 ( 3 pts)  Test 6

 ( 4 pts)  Test 7

Note: hard coding an answer earns zero points for the whole project
-10 points for not using main() in a meaningful way.
```

CSE 231 **Spring 2023**

Test 1:

```
              A long time ago in a galaxy far, far away...
             A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: n

Error:  unrecognized option [N]

Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: M

Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: X

May the force be with you.
```

Test 2:

```
               A long time ago in a galaxy far, far away...
              A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: a

      Enter N: 0

      Enter Base: 2

       0 in base 2:

      Enter option: A

      Enter N: 29

      Enter Base: 2

       29 in base 2: 00011101

      Enter option: a

      Enter N: 1.6

      Error: 1.6 was not a valid non-negative integer.

      Enter N: -5

      Error: -5 was not a valid non-negative integer.

      Enter N: 42

      Enter Base: 12

      Error: 12 was not a valid integer between 2 and 10 inclusive.

      Enter Base: 1

      Error: 1 was not a valid integer between 2 and 10 inclusive.

      Enter Base: 10

       42 in base 10: 00000042

      Enter option: x

May the force be with you.
```

Test 3:

```
              A long time ago in a galaxy far, far away...
             A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: b

      Enter string number S: 00000011

      Enter Base: 2

       00000011 in base 2: 3

      Enter option: B

      Enter string number S:

      Enter Base: 10

        in base 10: 0

      Enter option: b

      Enter string number S: 00000011

      Enter Base: 12

      Error: 12 was not a valid integer between 2 and 10 inclusive.

      Enter Base: 1

      Error: 1 was not a valid integer between 2 and 10 inclusive.

      Enter Base: 10

       00000011 in base 10: 11

      Enter option: x

May the force be with you.
```

Test 4:

```
               A long time ago in a galaxy far, far away...
              A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: c

      Enter base B1: 10

      Enter base B2: 1

      Error: 1 was not a valid integer between 2 and 10 inclusive.

      Enter base B2: 12

      Error: 12 was not a valid integer between 2 and 10 inclusive.

      Enter base B2: 2

      Enter string number: 00000003

       00000003 in base 10 is 00000011 in base 2...

      Enter option: C

      Enter base B1: 10

      Enter base B2: 2

      Enter string number: 0

       0 in base 10 is  in base 2...

      Enter option: x

May the force be with you.
```

Test 5:

```
              A long time ago in a galaxy far, far away...
             A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: e

      Enter a binary string of an image:
1010101010101000101000101011101011101010101010100101010110101010101010

      Enter number of bits used for pixels: 2

      Enter a text to hide in the image: CSE

       Original image:
1010101010101000101000101011101011101010101010100101010110101010101010

       Encoded image:
1011101010101101101100111010111110111010101110111001010101101010101010

      Enter option: E

      Enter a binary string of an image:

      Enter number of bits used for pixels: 2

      Enter a text to hide in the image: hello

      Image not big enough to hold all the text to steganography

      Enter option: E

      Enter a binary string of an image:
1010101010101000101000101011101011101010101010100101010110101010101010

      Enter number of bits used for pixels: 2

      Enter a text to hide in the image:

       Original image:
1010101010101000101000101011101011101010101010100101010110101010101010

       Encoded image:
1010101010101000101000101011101011101010101010100101010110101010101010

      Enter option: e

      Enter a binary string of an image: 101010101010
```

```
      Enter number of bits used for pixels: 2

      Enter a text to hide in the image: CSE

      Image not big enough to hold all the text to steganography

      Enter option: E

      Enter a binary string of an image:
10101010101010001010001010111010111010101010101010010101011010101010101010101010101010001
01000101011101011101010101010101001010101101010101010101010101010101000101000101011101011
10101010101010100101010111010101010101010101010101010100010100010101110101110101010101010100
10101011010101010101010

      Enter number of bits used for pixels: 4

      Enter a text to hide in the image: Hello!

       Original image:
10101010101010001010001010111010111010101010101010010101011010101010101010101010101010001
01000101011101011101010101010101001010101101010101010101010101010101000101000101011101011
10101010101010100101010111010101010101010101010101010100010100010101110101110101010101010100
10101011010101010101010

       Encoded image:
10101011101010001011001010101010111010111011101010000101011010111010101110111010101110011
01000101010101111111010101110111000010001101011101110101011101110111001101000101011101011
10101010101011100101010111010101010101010101010101010100010100010101110101110101010101010100
10101011010101010101010

      Enter option: x

May the force be with you.
```

Test 6:

```
               A long time ago in a galaxy far, far away...
              A terrible civil war burns throughout the galaxy.
  ~~ Your mission: Tatooine planet is under attack from stormtroopers,
                   and there is only one line of defense remaining
                   It is up to you to stop the invasion and save the planet~~


Please choose one of the options below:
             A. Convert a decimal number to another base system
             B. Convert decimal number from another base.
             C. Convert from one representation system to another.
             E. Encode an image with a text.
             D. Decode an image.
             M. Display the menu of options.
             X. Exit from the program.

      Enter option: d

      Enter an encoded string of an image:
1011101010101101101100111010111110111010101110111001010101101010101010

      Enter number of bits used for pixels: 2

       Original text: CSEx

      Enter option: D

      Enter an encoded string of an image:

      Enter number of bits used for pixels: 4

       Original text:

      Enter option: D

      Enter an encoded string of an image:
10101011101010001011001010101010111010111011101010000101011010111010101110111010101110011
01000101010101111111010101110111000010001101011101110101011101110111001101000101011101011
10101010101011100001010110101110111010101010

      Enter number of bits used for pixels: 4

       Original text: Hello!X

      Enter option: X

May the force be with you.
```