Lana Pinjic

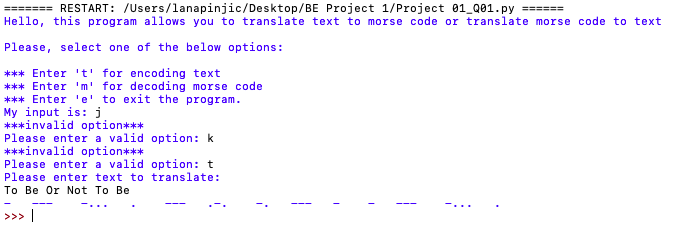
Dr. Thaer W. Jayyousi

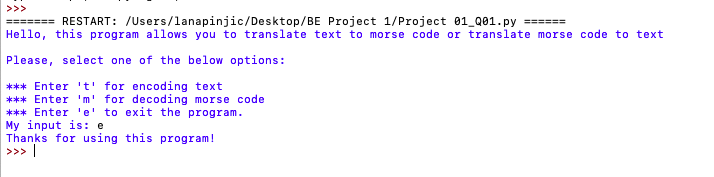
BE\_1600\_2109\_005

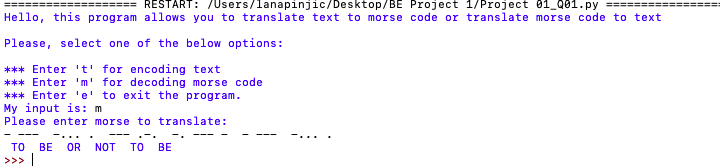
November 29 2021

BE Project 1600

Problem 1

1. The problem here asks us to be able to provide a menu that asks options for user input, once the user has decided whether they would like to either exit, encrypt or decrypt they would then either be closed out of the program or they would be asked for the sentence to encrypt or the morse code to decrypt. In the end you would either get an output of, saying that the program is ending if you choose to do so. If you chose to decrypt a line of morse code, then your output would be the original string, something readable. If you chose to encrypt a word or sentence of your choosing, then your output would be that exact word/sentence but just written in morse code.
2. The major steps to solving the problem came to first creating a dictionary where all the translations for morse would be contained, this we need for both encryption and decryption, along with a function which I named ‘function’ to store the dictionary. Next you would have to create the menu. After all it is the first thing to pop up on the screen when the user opens the program. Here we would create another function inside of the first one which I named ‘menu’ hence what it contains. Here we will print out the instructions for the reader. The next step is to create a third function underneath the first which I named “encrypt” which its job is to take the message that the inputs to encrypt and to change each letter individually by looking up the letters in the dictionary and printing the corresponding value as morse code. Here we will also write a code to separate each word with 3 spaces and one space in between each letter. The 4th function we will be creating is for the purpose of decrypting the morse code that the user will input, which I named ‘decrypt’. Here the main goal is to firstly add an extra space a the end of the code and then write the code to check the number of spaces to keep track. The next thing we will do is write what the code will do when it encounters a space, to format it as either a word or just a character. The last thing we will do will be the opening of the dictionary and accessing the keys using their values (the reverse of encryption). The 5th function we will then be using is the main function where we will put together all of the information above and write it all out in simpler terms, I named my function ‘mainfunc’, and started asking the user for their input. If they answered with the letter ‘e’, then the program just prints a final remark before closing. If they answer with the letter ‘t’, then we can ask for the sentence they would like to encrypt, into morse code. From there our code will encrypt it using function ‘encrypt’ from earlier, using the uppercase format of the letters in the sentence and then print the encrypted result. If they answer with the letter ‘m’, then we will again ask for a sentence in morse code to translate. Here we would write the code to infer the function decrypt, and then print the result. However, if a user inputs any other input that is not ‘t’,’m’, or ‘e’, then the program will alert them that this is an invalid option and to try again. Here we will create an infinite loop, until the user inputs a correct option. Once the user does so, it will carry out the same actions listed above or ‘t’, ‘m’, and ‘e’, and then the infinite loop will end. Then at the end it is critical to close all of the functions to make sure that everything is working together seamlessly.
3. 





Problem 2

1. The problem here asks us to input a user choice where they can chose to access the database of emails and names, or edit it, or add more information, or to exit out of the program entirely. The output varies for each of the options selected by the user, either they will be able to successfully edit information, add new information, be alerted that the information already exists, be alerted if searching that something is not in the dictionary, to close the program, or to reiterate a key or value in the dictionary when the user searches for it. Once this has been completed the program finishes.
2. The major steps I took to complete this problem began with creating the dictionary which we through our runs will be able to open it and write in it and edit the existing information. I started off by creating 2 separate clauses in code, to read the dictionary, but if there is nothing in the dictionary (empty) to be able to write it, transferring all the information in the file to the dictionary. Then I was able to split up my keys and values identifying variables to relate to each of those separating them by a space (to identify as separate words). Then I was able to create the first function for the first option, I would have created the function for the main menu which would be the first thing to pop up on the user’s screen however, I needed to be able to define each of the actions for each choice/input before the main menu function where the computer would just reference back to find these actions and specific variables. This second function I named ‘choiceone’ in reference to if the user chose choice 1 in the menu. Here I started by accessing the dictionary, and then seeing If the key or names in the dictionary at that moment matched the user input, if they did this would print out both the name and the email (value) associated with it. If not, it would print that nothing was found and associate it with False instead of True. The third function was for the second user input choice names ‘choicetwo’ where I again accessed the dictionary and allowed for the user to input both the name and the email, relating them to the keys and values in the dictionary. Then I saw if any of the keys or values matched the ones in the dictionary, and if they were it would print a message that says so. But, if there are no matches than we can open the txt file, which will be added to the dictionary, and write in it creating a new key and value to be stored for later. The fourth function that I created was for the user input of three which I named ‘choicethree’, started by accessing the dictionary, and then taking in another input for the name, this is how we will identify what email they would like to change if it exists. If they identify a key/name as imputed, and then open dictionary and edit using the write function, and update the value associated with that key. The fifth function that I created, taking in the user input of selection 4, I named ‘choicefour’ and it will begin by accessing the dictionary. Then I will ask the user to input for the name or the key in the dictionary and if there is a match, we can easily remove it from the dictionary entirely. To do this we identify that we want to remove it, then we will open the dictionary to write it and save it in the file. The last function and sixth function we will be creating will be for the main menu! Here we will first print the options for the user to see and then take in a user input I labeled as the variable ‘choice’, then we will identify what the program will do if the user input is not numbers 1, 2, 3, 4, and 5 and will print out a message saying its invalid and prompting the menu again. However, if they do say a valid menu choice then for each specific number will have its own functions/directions to refer to the above functions. Here I will use the if and elif functions in order to allows for conditional checking as if all of the if’s are not included it will move on the elif’s, which is for efficiently and used when I need multiple if statements which is here. For each choice if it is equal to the number ‘==’ then it will execute the funtions above “choiceone() or “choicetwo(), or choicethree() or choicefour()”. Because choice 5, only needs to end the program it does not need its own function and instead will just print out a statement doing so. Then close out the main menu function and your done!!!!
3. 