**Lab\_10. RegEX**

1. Implement a function encode\_symbol . \_ \_ The function takes a character and a number to shift it by. The function encrypts only Latin characters, leaving all others unchanged.
2. Implement a function encode\_str . \_ \_ The function takes a string and an encryption key and returns the encrypted string.
3. Implement function encode\_file ( input\_file\_path , output\_file\_path , key) .
4. Implement a GUI for your application with easygui . To open an encrypted file, use the function easygui . fileopenbox ( msg = None , title = None ) . To save the encrypted file, use easygui . filesavebox ( msg = None , title = None ) . For input key - easygui.integerbox ( msg ="", title=" ") .
5. Highlight with tag **< strong >** repetitive words.

6. Select existing dates between 1000 and 2022. Seconds may be omitted.   
Note: Each month has 30 days.

7. Select **IPv4 \_** addresses in all possible representations: decimal, hexadecimal and octal. With and without dots. Learn more about IP addresses [on wikipedia](https://ru.wikipedia.org/wiki/IPv4)

8. Select a sequence of non-repeating characters in alphabetical order. Spaces should be ignored.

9. Remove repeating spaces and tabs, leave one space between words and two spaces between sentences.

10. Choose repeating words that are directly under each other.   
Supposed usage monospace font . Strings longer than 32 characters are transferred .

11. Choose a well-formed IRC message.   
Here is a link to [Russian version of the specification .](http://www.kvirc.ru/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_IRC_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%B0_(rfc1459)_%D0%BD%D0%B0_%D1%80%D1%83%D1%81%D1%81%D0%BA%D0%BE%D0%BC_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5#.D0.A4.D0.BE.D1.80.D0.BC.D0.B0.D1.82_.D1.81.D0.BE.D0.BE.D0.B1.D1.89.D0.B5.D0.BD.D0.B8.D1.8F_.D0.B2_.27.D0.BF.D1.81.D0.B5.D0.B2.D0.B4.D0.BE.27_BNF)

12. Write a regular expression that finds the given word in the text.

13. Create a regular expression that finds all ciphers in a given string or file.

14. Create a regular expression to find any word that starts with an arbitrary character preceding the letters lo . Find combinations of hello and dilo in the given file.

15. You need to check the list of phone numbers using regular expressions.

16. Split a string into multiple delimiters.

17. Need to extract data from html file that is between the tags.

18. Time format is hours:minutes. Both hours and minutes consist of two digits, for example: 09:00. Write a RegEx expression to find the time in the string: "Breakfast at 09:00". Please note that " 37:98" is incorrect time .