

HOMework

Task 1

Write a program that accepts a string as input. This line is the path to the file.

If the file exists:

- 1) Check whether it is possible to write to it
- 2) Delete that file and finish the job

If there is no file:

- 1) Create it
- 2) Write a string with Latin letters and any symbols that are not in ASCII table (diacritical, cyrillic etc.) in UTF-8 encoding
- 3) Read and display the contents in WINDOWS-1251 and UTF-8 encoding
- 4) Delete that file and finish the job

HOMework

Task 2

Write a program that uses the list:

```
lst = [25, 1, 77.15, 0, "py", "2.50", "ten", False, "", True, 145, "10"]
```

- The program should divide number 100 into every element of the list.
- The program should count the sum of all division results greater or equal (\geq) 4.

```
sum = 0, 100/10=10 -> 10>=4 -> sum=0+10=10
```

```
sum = 10, 100/100=1 -> 1<4 -> sum = 10
```

```
sum = 10, 100/5=20 -> 20>=4 -> sum =10+20=30
```

- Catch the error of converting the string to the integer and the division by 0 error.
- Each error must be caught in one except block and the corresponding message should be displayed which is transmitted in the error object.

HOMework

Task 3 (not mandatory) *For cool boys and girls

This is a strange dictionary. The message is encrypted in it. Dictionary keys are ASCII characters. But they are not just symbols, the last two least significant bits are inverted in them. Values are a list of positions in the line where this symbol (key) occurs. Arrange the characters in the correct positions and you will decrypt the message.

{92: [1, 2, 5, 6, 7, 8, 9, 10, 13, 14, 17, 18, 19, 22, 23, 24, 25, 28, 29, 30, 31, 32, 33, 37, 38, 39, 40, 43, 44, 45, 46, 49, 53, 54, 55, 56, 57, 58, 59, 60, 69, 70, 84, 85, 90, 91, 92, 93, 99, 100, 105, 106, 117, 118, 119, 120, 145, 150, 151, 177, 178, 187, 193, 201, 204, 208, 209, 210, 217, 223, 235, 236, 237, 240, 243, 244, 245, 246, 249, 250, 251, 252, 257, 261, 263, 264, 265, 266, 267, 272, 273, 274, 275, 276, 279, 280, 281, 282, 284, 288, 290, 291, 292, 293, 294], 35: [3, 4, 11, 12, 15, 16, 20, 21, 26, 27, 34, 35, 36, 41, 42, 47, 48, 50, 51, 52, 63, 66, 68, 71, 74, 76, 78, 80, 83, 86, 89, 95, 96, 98, 101, 104, 107, 110, 112, 114, 116, 123, 125, 126, 128, 130, 132, 134, 136, 138, 143, 147, 149, 153, 154, 155, 156, 158, 160, 162, 164, 166, 168, 170, 171, 174, 176, 181, 183, 185, 189, 191, 195, 197, 198, 200, 203, 206, 211, 212, 213, 215, 219, 221, 225, 227, 230, 231, 233, 254, 255, 259, 269, 270, 286], 9: [61, 122, 180, 238, 296], 95: [62, 64, 72, 87, 102, 108, 124, 242, 248, 278], 44: [65, 67, 73, 75, 77, 82, 88, 94, 97, 103, 109, 113, 115, 121, 127, 129, 131, 133, 135, 137, 139, 141, 142, 144, 146, 148, 152, 157, 159, 161, 163, 165, 167, 169, 173, 175, 179, 182, 184, 186, 188, 190, 192, 194, 196, 199, 205, 207, 214, 216, 218, 220, 222, 224, 226, 228, 232, 234, 239, 241, 247, 253, 256, 258, 262, 268, 271, 277, 283, 285, 289, 295], 43: [79], 42: [81], 127: [111, 140, 172, 229, 260, 287], 47: [202]}



DMITRII DIACIKOVSKII
DEVOPS ENGINEER
dmitrii.diacikovskii@endava.com