# Evidencias de solución a problema 2 Converter

## Pylint execution

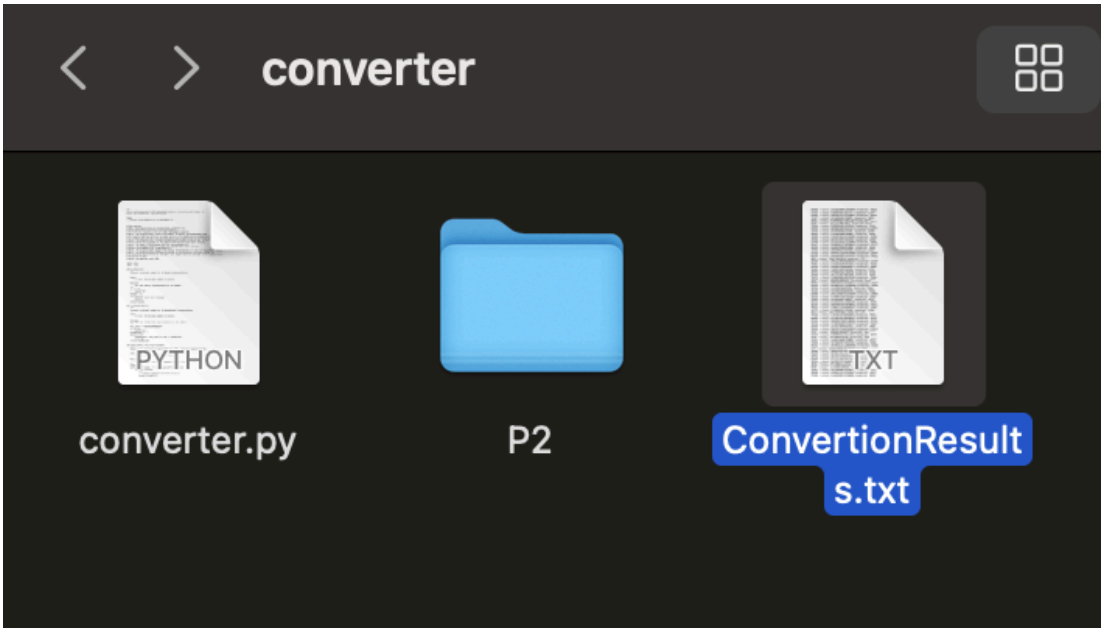
```
➜  converter git:(main) ✗ pylint converter.py

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

## Execution when file not provided

```
➜  converter git:(main) ✗ python3 converter.py
Usage: python convertNumbers.py fileWithData.txt
```

## Resulting file



## Test executions

Screenshots don’t contain the complete result due to screen size limit.

### TC1

```
→  converter git:(main) ✗ python3 converter.py P2/TC1.txt
6980368 -> Binary: 11010101000001100010000, Hexadecimal: 6A8310
5517055 -> Binary: 10101000010111011111111, Hexadecimal: 542EFF
1336159 -> Binary: 10100011000110101111, Hexadecimal: 14635F
6750185 -> Binary: 11001101111111111101001, Hexadecimal: 66FFE9
1771937 -> Binary: 11011000010011010001, Hexadecimal: 1B09A1
360952 -> Binary: 1011000000111111000, Hexadecimal: 581F8
5672561 -> Binary: 10101101000111001110001, Hexadecimal: 568E71
916583 -> Binary: 11011111110001100111, Hexadecimal: DFC67
2700138 -> Binary: 10100100011001101101010, Hexadecimal: 29336A
9645053 -> Binary: 10010011001010111111101, Hexadecimal: 932BFD
1181110 -> Binary: 10010000010110110110, Hexadecimal: 1205B6
1492185 -> Binary: 10110110001001101001, Hexadecimal: 16C4D9
4018595 -> Binary: 11110101010001101000011, Hexadecimal: 3D51A3
7654888 -> Binary: 11101001100110111101000, Hexadecimal: 74CDE8
7062453 -> Binary: 11010111100001110110101, Hexadecimal: 6BC3B5
2478010 -> Binary: 10010111001111101111010, Hexadecimal: 25CFBA
6134768 -> Binary: 10111011001101111110000, Hexadecimal: 5D9BF0
8420417 -> Binary: 100000000111110001000001, Hexadecimal: 807C41
2917489 -> Binary: 10110010000100011110001, Hexadecimal: 2C8471
3340773 -> Binary: 11001011111001111100101, Hexadecimal: 32F9E5
1115956 -> Binary: 10001000001110011010 0, Hexadecimal: 110734
9172192 -> Binary: 100010111111010011100000, Hexadecimal: 8BF4E0
6271996 -> Binary: 10111111011001111111100, Hexadecimal: 5FB3FC
8686939 -> Binary: 100001001000110101011011, Hexadecimal: 848D5B
50986 -> Binary: 1100011100101010, Hexadecimal: C72A
9376410 -> Binary: 100011110001001010011010, Hexadecimal: 8F129A
5962327 -> Binary: 10110101111101001010111, Hexadecimal: 5AFA57
```

## TC2

```
➜  converter git:(main) ✗ python3 converter.py P2/TC2.txt
7116776 -> Binary: 11011001001011111101000, Hexadecimal: 6C97E8
1666340 -> Binary: 110010110110100100100, Hexadecimal: 196D24
8886983 -> Binary: 100001111001101011000111, Hexadecimal: 879AC7
839365 -> Binary: 11001100111011000101, Hexadecimal: CCEC5
924280 -> Binary: 11100001101001111000, Hexadecimal: E1A78
1026310 -> Binary: 11111010100100000110, Hexadecimal: FA906
1615293 -> Binary: 11000101001010111101, Hexadecimal: 18A5BD
1063875 -> Binary: 10000001110111000011, Hexadecimal: 103BC3
679035 -> Binary: 10100101110001111011, Hexadecimal: A5C7B
5201970 -> Binary: 10011110110000000110010, Hexadecimal: 4F6032
593979 -> Binary: 10010001000000111011, Hexadecimal: 9103B
801371 -> Binary: 11000011101001011011, Hexadecimal: C3A5B
3796878 -> Binary: 11100111110111110001110, Hexadecimal: 39EF8E
7489201 -> Binary: 11100100100011010110001, Hexadecimal: 7246B1
9740020 -> Binary: 10010100100111011110100, Hexadecimal: 949EF4
9128737 -> Binary: 10001011010010110010100001, Hexadecimal: 8B4B21
5473463 -> Binary: 10100111000010010110111, Hexadecimal: 5384B7
8701957 -> Binary: 10000100110010000000101, Hexadecimal: 84C805
8238050 -> Binary: 11111011011001111100010, Hexadecimal: 7DB3E2
8679038 -> Binary: 10000100011011001111110, Hexadecimal: 846E7E
385912 -> Binary: 1011110001101111000, Hexadecimal: 5E378
5867340 -> Binary: 10110011000011101001100, Hexadecimal: 59874C
4894542 -> Binary: 10010101010111101001110, Hexadecimal: 4AAF4E
8999451 -> Binary: 10001001010100100011011, Hexadecimal: 89521B
4392535 -> Binary: 10000110000011001010111, Hexadecimal: 430657
2078131 -> Binary: 11111011010110110011, Hexadecimal: 1FB5B3
3070124 -> Binary: 10111011011000101011100, Hexadecimal: 2ED8AC
7451998 -> Binary: 11100011011010101011110, Hexadecimal: 71B55E
5635510 -> Binary: 10101011111110110110110, Hexadecimal: 55FDB6
1233932 -> Binary: 10010110101000001100, Hexadecimal: 12D40C
6089867 -> Binary: 10111001110110010001011, Hexadecimal: 5CEC8B
1792316 -> Binary: 11011010110010011100, Hexadecimal: 1B593C
6298637 -> Binary: 11000000001110000001101, Hexadecimal: 601C0D
2408038 -> Binary: 10010010111111001100110, Hexadecimal: 24BE66
8510100 -> Binary: 10000001110110101010100, Hexadecimal: 81DA94
991581 -> Binary: 11110010000101011101, Hexadecimal: F215D
6455739 -> Binary: 11000101000000110111011, Hexadecimal: 6281BB
```

**TC3**

```
➜  converter git:(main) ✗ python3 converter.py P2/TC3.txt
-39 -> Binary: , Hexadecimal:
-36 -> Binary: , Hexadecimal:
8 -> Binary: 1000, Hexadecimal: 8
34 -> Binary: 100010, Hexadecimal: 22
17 -> Binary: 10001, Hexadecimal: 11
49 -> Binary: 110001, Hexadecimal: 31
5 -> Binary: 101, Hexadecimal: 5
39 -> Binary: 100111, Hexadecimal: 27
0 -> Binary: 0, Hexadecimal: 0
33 -> Binary: 100001, Hexadecimal: 21
12 -> Binary: 1100, Hexadecimal: C
-6 -> Binary: , Hexadecimal:
27 -> Binary: 11011, Hexadecimal: 1B
-4 -> Binary: , Hexadecimal:
-38 -> Binary: , Hexadecimal:
26 -> Binary: 11010, Hexadecimal: 1A
49 -> Binary: 110001, Hexadecimal: 31
29 -> Binary: 11101, Hexadecimal: 1D
42 -> Binary: 101010, Hexadecimal: 2A
-16 -> Binary: , Hexadecimal:
-28 -> Binary: , Hexadecimal:
34 -> Binary: 100010, Hexadecimal: 22
20 -> Binary: 10100, Hexadecimal: 14
0 -> Binary: 0, Hexadecimal: 0
25 -> Binary: 11001, Hexadecimal: 19
45 -> Binary: 101101, Hexadecimal: 2D
3 -> Binary: 11, Hexadecimal: 3
-46 -> Binary: , Hexadecimal:
-46 -> Binary: , Hexadecimal:
29 -> Binary: 11101, Hexadecimal: 1D
33 -> Binary: 100001, Hexadecimal: 21
29 -> Binary: 11101, Hexadecimal: 1D
26 -> Binary: 11010, Hexadecimal: 1A
-5 -> Binary: , Hexadecimal:
-36 -> Binary: , Hexadecimal:
12 -> Binary: 1100, Hexadecimal: C
45 -> Binary: 101101, Hexadecimal: 2D
```

TC4

```
➜  converter git:(main) ✗ python3 converter.py P2/TC4.txt
Invalid data encountered and skipped: ABC
Invalid data encountered and skipped: ERR
Invalid data encountered and skipped: VAL
-39 -> Binary: , Hexadecimal:
-36 -> Binary: , Hexadecimal:
8 -> Binary: 1000, Hexadecimal: 8
34 -> Binary: 100010, Hexadecimal: 22
17 -> Binary: 10001, Hexadecimal: 11
49 -> Binary: 110001, Hexadecimal: 31
5 -> Binary: 101, Hexadecimal: 5
0 -> Binary: 0, Hexadecimal: 0
33 -> Binary: 100001, Hexadecimal: 21
12 -> Binary: 1100, Hexadecimal: C
-6 -> Binary: , Hexadecimal:
27 -> Binary: 11011, Hexadecimal: 1B
-4 -> Binary: , Hexadecimal:
-38 -> Binary: , Hexadecimal:
26 -> Binary: 11010, Hexadecimal: 1A
49 -> Binary: 110001, Hexadecimal: 31
29 -> Binary: 11101, Hexadecimal: 1D
42 -> Binary: 101010, Hexadecimal: 2A
-16 -> Binary: , Hexadecimal:
34 -> Binary: 100010, Hexadecimal: 22
20 -> Binary: 10100, Hexadecimal: 14
0 -> Binary: 0, Hexadecimal: 0
25 -> Binary: 11001, Hexadecimal: 19
45 -> Binary: 101101, Hexadecimal: 2D
3 -> Binary: 11, Hexadecimal: 3
-46 -> Binary: , Hexadecimal:
-46 -> Binary: , Hexadecimal:
29 -> Binary: 11101, Hexadecimal: 1D
33 -> Binary: 100001, Hexadecimal: 21
29 -> Binary: 11101, Hexadecimal: 1D
26 -> Binary: 11010, Hexadecimal: 1A
-5 -> Binary: , Hexadecimal:
-36 -> Binary: , Hexadecimal:
```