

# Contents

<a href="#">1 Basic Test Results</a>	2
<a href="#">2 wordsearch.py</a>	4

# 1 Basic Test Results

```
1 Starting tests...
2 Thu Aug 5 21:45:15 IDT 2021
3 fc0a09fe588c1e1ec14854117e1daa1cd0b0d313 -
4
5
6 Archive: /tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/submission
7 inflating: src/wordsearch.py
8
9
10 Running presubmit code tests...
11 Process Process-3:
12 Traceback (most recent call last):
13   File "/usr/lib/python3.7/multiprocessing/process.py", line 297, in _bootstrap
14     self.run()
15   File "/usr/lib/python3.7/multiprocessing/process.py", line 99, in run
16     self._target(*self._args, **self._kwargs)
17   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/autotest.py", line 74, in wrap
18     res=target(*args, **kwargs)
19   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/testrunners.py", line 39, in import_runner
20     code,res = peel(runners, modulename, fname, args, kwargs)
21   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/testrunners.py", line 7, in peel
22     return runners[-1](modulename, fname, args, kwargs,options,runners[:-1])
23   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/testrunners.py", line 17, in check_args
24     code,res = peel(runners, modulename, fname, args, kwargs)
25   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/testrunners.py", line 7, in peel
26     return runners[-1](modulename, fname, args, kwargs,options,runners[:-1])
27   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/lib/testrunners.py", line 12, in base_runner
28     return None,func(*args, **kwargs)
29   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/src/wordsearch.py", line 184, in find_words
30     check = from_letter_to_direction_str(matrix, directions)
31   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/src/wordsearch.py", line 169, in from_letter_to_dir
32     lst_of_strings_all_direct.extend(up_right_string(matrix))
33   File "/tmp/bodek.LeoVbi/intro2cse/ex5/daniel7895/presubmission/testdir/src/wordsearch.py", line 104, in up_right_string
34     word += matrix[i][j]
35 IndexError: list index out of range
36 --> BEGIN TEST INFORMATION
37 Test name: ex5_findwords
38 Module tested: wordsearch
39 Function call: find_words(['long', 'short', 'can', 'toe', 'poet', 'crop', 'dog', 'cat', 'ants', 'apple', 'cake'],[['a', 'p',
40 Expected return value: [('cat', 2), ('dog', 1)]
41 More test options: {}
42 --> END TEST INFORMATION
43 *****
44 ***** There is a problem:
45 ***** The test named 'ex5_findwords' failed.
46 *****
47 Test did not complete, exited with exitcode 1.
48 This probably means your code caused an exception to be raised.
49 result_code ex5_findwords exception 1
50 Traceback (most recent call last):
51   File "src/wordsearch.py", line 217, in <module>
52     main_func()
53   File "src/wordsearch.py", line 212, in main_func
54     result = find_words(words_list, matrix, lst1[4])
55   File "src/wordsearch.py", line 184, in find_words
56     check = from_letter_to_direction_str(matrix, directions)
57   File "src/wordsearch.py", line 169, in from_letter_to_direction_str
58     lst_of_strings_all_direct.extend(up_right_string(matrix))
59   File "src/wordsearch.py", line 104, in up_right_string
```

```

60     word += matrix[i][j]
61 IndexError: list index out of range
62 --> BEGIN TEST INFORMATION
63 Test name: ex5_program
64 Module tested: wordsearch
65 Function call: None('words1.txt','mat1.txt','out.txt','wl')
66 Expected return value: ['cat,2', 'dog,1']
67 More test options: {'timeout': 4}
68 --> END TEST INFORMATION
69 *****
70 *****          There is a problem:
71 *****          The test named 'ex5_program' failed.
72 *****
73 program exited with exitstatus 1
74 result_code    ex5_program    exitstatus    1
75 3 passed tests out of 5 in test set named 'ex5'.
76 result_code    ex5      3      1
77 Done running presubmit code tests
78
79 Finished running the presubmit tests
80
81 Additional notes:
82
83 Make sure to thoroughly test your code.
84

```

## 2 wordsearch.py

```
1 import sys
2
3
4 def read_wordlist(filename_words_list):
5     """get file with string of words, divided by \n \
6     and convert it to list of words (without the \n)"""
7     with open(filename_words_list) as file:
8         words_list = file.read().splitlines()
9     return words_list
10
11
12 def read_matrix(filename_mat):
13     """get file with lines of lettres, split the lines and then convert
14     every line to list, so we get list of lists. """
15     with open(filename_mat) as file:
16         line = file.read().splitlines()
17         matrix = []
18         for i in line:
19             x = list(i.split(','))
20             matrix.append(x)
21     return matrix
22
23
24 def up_string(matrix):
25     """get matrix, return list of string, every string\
26     is the letters in column, in up direction"""
27     if len(matrix[0]) > 0:
28         mat_to_str = []
29         for y in range(len(matrix[0])):
30             word = ""
31             for x in range(len(matrix)-1, -1, -1):
32                 word += matrix[x][y]
33             mat_to_str.append(word)
34         return mat_to_str
35     else:
36         return None
37
38
39 def down_string(matrix):
40     """get matrix, return list of string, every string\
41     is the letters in column, down up direction"""
42     if len(matrix) > 0:
43         mat_to_str = []
44         for y in range(len(matrix[0])):
45             word = ""
46             for x in range(len(matrix)):
47                 word += matrix[x][y]
48             mat_to_str.append(word)
49         return mat_to_str
50     else:
51         return ''
52
53
54 def right_string(matrix):
55     """get matrix, return list of string, every string\
56     is the letters in row, in right direction"""
57     mat_to_str = []
58     if len(matrix) > 0:
59         for y in range(len(matrix)):
```

```

60         word = ""
61         for x in range(len(matrix[0])):
62             word += matrix[y][x]
63         mat_to_str.append(word)
64     return mat_to_str
65 else:
66     return ''
67
68
69 def left_string(matrix):
70     """get matrix, return list of string, every string\
71     is the letters in row, in left direction"""
72     if len(matrix) > 0:
73         mat_to_str = []
74         for y in range(len(matrix)):
75             word = ""
76             for x in range(len(matrix[0])-1, -1, -1):
77                 word += matrix[y][x]
78             mat_to_str.append(word)
79     return mat_to_str
80 else:
81     return ''
82
83
84 def up_right_string(matrix):
85     """get matrix, return list of string, every string\
86     is the letters in the diagonal, up right direction"""
87     row_count = len(matrix)
88     col_count = len(matrix[0])
89     mat_to_str = []
90     for k in range(row_count):
91         word = ''
92         i = k
93         j = 0
94         while i >= 0 and j <= col_count - 1:
95             word += matrix[i][j]
96             i -= 1
97             j += 1
98         mat_to_str.append(word)
99     for k in range(col_count - 1):
100         word = ''
101         i = col_count
102         j = k
103         while i >= 0 and j <= (col_count - 1):
104             word += matrix[i][j]
105             i -= 1
106             j += 1
107         mat_to_str.append(word)
108     return mat_to_str
109
110
111 def up_left_string(matrix):
112     """get matrix, return list of string, every string\
113     is the letters in the diagonal, up left direction\
114     revers the strings from the down right"""
115     mat_to_str = down_right_string(matrix)
116     res = [i[::-1] for i in mat_to_str]
117     return res
118
119
120 def down_right_string(matrix):
121     """get matrix, return list of string, every string\
122     is the letters in the diagonal, down_right direction"""
123     row_count = len(matrix)
124     col_count = len(matrix[0])
125     mat_to_str = []
126     for k in range(row_count-1, 0, -1):
127         word = ''

```

```

128         i = k
129         j = 0
130         while i <= col_count and j >= 0:
131             word += matrix[i][j]
132             i += 1
133             j += 1
134         mat_to_str.append(word)
135     for k in range(col_count):
136         word = ''
137         i = 0
138         j = k
139         while j <= (col_count - 1):
140             word += matrix[i][j]
141             i += 1
142             j += 1
143         mat_to_str.append(word)
144     return mat_to_str
145
146
147 def down_left_string(matrix):
148     """get matrix, return list of string, every string\
149     is the letters in the diagonal, down_right direction"""
150     mat_to_str = up_right_string(matrix)
151     res = [i[::-1] for i in mat_to_str]
152     return res
153
154
155 def from_letter_to_direction_str(matrix, directions):
156     """if the char in the string of direction, call\
157     the function that provide the string of the direction\
158     and append to lst_of_strings_all_direct"""
159     lst_of_strings_all_direct = []
160     if 'u' in directions:
161         lst_of_strings_all_direct.extend(up_string(matrix))
162     if 'd' in directions:
163         lst_of_strings_all_direct.extend(down_string(matrix))
164     if 'r' in directions:
165         lst_of_strings_all_direct.extend(right_string(matrix))
166     if 'l' in directions:
167         lst_of_strings_all_direct.extend(left_string(matrix))
168     if 'w' in directions:
169         lst_of_strings_all_direct.extend(up_right_string(matrix))
170     if 'x' in directions:
171         lst_of_strings_all_direct.extend(up_left_string(matrix))
172     if 'y' in directions:
173         lst_of_strings_all_direct.extend(down_right_string(matrix))
174     if 'z' in directions:
175         lst_of_strings_all_direct.extend(down_left_string(matrix))
176     return lst_of_strings_all_direct
177
178
179 def find_words(words_list, matrix, directions):
180     """for every word in words_list, check if the word\
181     in the lst_of_strings_all_direct that was created,\
182     if the word in the string, count how many times. append the word to the with\
183     the numbers of appearance to the result list"""
184     check = from_letter_to_direction_str(matrix, directions)
185     results = []
186     for word in words_list:
187         counter = 0
188         if len(check) > 0:
189             for x in check:
190                 counter += sum(x[i:i+len(word)] == word for i in range(len(x)))
191             if counter > 0:
192                 result_tuple = (word, counter)
193                 results.append(result_tuple)
194         else:
195             return None

```

```

196     return results
197
198
199 def write_output(result, filename_results):
200     """creat file with the result\
201     Or overwrites an existing file"""
202     with open(filename_results, 'w') as file:
203         for pair in result:
204             file.write(str(pair[0]) + "," + str(pair[1]))
205             file.write("\n")
206
207
208 def main_func():
209     lst1 = sys.argv
210     words_list = read_wordlist(lst1[1])
211     matrix = read_matrix(lst1[2])
212     result = find_words(words_list, matrix, lst1[4])
213     write_output(result, lst1[3])
214
215
216 if __name__ == "__main__":
217     main_func()

```