Dawn's Lawn

Dawn's lawn

Dawn has a magical lawnmower that she uses to mow her square lawn.
As soon as she trims the grass, it starts growing quickly.
Once the grass grows tall enough, it turns into a flower.
Dawn has a lawn that has flowers, grass, and dirt.
More details can be found here:

https://mega.nz/#!bTwAgS5K!_7Ct7i5rPzsTwxB-QXuM0C751gYppdItpfGl3Cr-RaY

You can find Dawn's actual lawn (your input) here:

https://mega.nz/#!OGxQDQgA!pQdDXyto9sqdoprz53541HTyPMr9aLFBKP29Rutqom8

How many flowers are in Dawn's lawn after she mows it completely?

Flag: CTFlearn{194}

Writeup:

created python program to count flowers on the lawn and ran command python lawn.py dawn2.txt to get flag of 194

```
import sys
import os
import time
from colorama import Style, Fore, Back

VALUES = ['*', '|', '/', '-', '\\', '_', '.']

class Grass:
    def __init__(self, init_value, life_span):
        assert init_value in VALUES

        self.value = init_value
        self.trimmed = False
        self.life_length = 0
        self.life_span = life_span

    def trim(self):
        self.trimmed = True
        self.life_length = 0
        i = VALUES.index(self.value) + 2
        i = i if i < len(VALUES) else len(VALUES) - 1
        self.value = VALUES[i]

    def grow(self):
        if self.value in (VALUES[-1], VALUES[0]) or not self.trimmed:
            return
        self.life_length += 1
        if self.life_length >= self.life_span:
            self.life_length = 0
            i = VALUES.index(self.value) - 1
            self.value = VALUES[i]

    def __repr__(self):
        return self.value


class Lawn:
    def __init__(self, lines):
        self.lawn = []
        lifespan = self.get_lifespan(lines)
        for line in lines:
            row = []
            for c in line:
                row.append(Grass(c, lifespan))
            self.lawn.append(row)
        self.rows = len(self.lawn)
        self.cols = len(self.lawn[0])
        self.x = 0
        self.y = 0

    def get_lifespan(self, lines):
        return len(lines)

    def step(self):
        if self.x >= self.cols:
            return True
        current_grass = self.lawn[self.y][self.x]
        current_grass.trim()
        self.y += 1
        if self.y == self.rows:
            self.y = 0
            self.x += 1
        flag = False
        for row in range(self.rows):
            for col in range(self.cols):
                grass = self.lawn[col][row]
                grass.grow()
                flag = not grass.trimmed
                if flag:
```

```
                    break
            if flag:
                break
        return False

    def count_flowers(self):
        flowers = 0
        for row in self.lawn:
            for grass in row:
                if grass.value == VALUES[0]:
                    flowers += 1
        return flowers

    def __repr__(self):
        lawn_str = ''
        for rowi, row in enumerate(self.lawn):
            for coli, grass in enumerate(row):
                if grass.value == VALUES[-1]:
                    color = Fore.BLACK
                elif grass.value == VALUES[0]:
                    color = Fore.MAGENTA
                elif grass.trimmed == False:
                    color = Fore.WHITE
                else:
                    color = {
                        "|": Fore.GREEN,
                        "/": Fore.CYAN,
                        "-": Fore.YELLOW,
                        "\\": Fore.LIGHTRED_EX,
                        "_": Fore.RED,
                    }.get(grass.value)
                lawn_char = color + repr(grass) + Style.RESET_ALL + \
                    ('0' if rowi == self.y and coli == self.x else " ")
                lawn_str += lawn_char
            lawn_str.strip()
            lawn_str += "\n"
        return lawn_str


if __name__ == '__main__':
    with open(sys.argv[1], "r") as f:
        lines = f.readlines()
    for i, line in enumerate(lines):
        line = line.strip()
        lines[i] = line
    lawn = Lawn(lines)
    while True:
        flag = lawn.step()
        time.sleep(.05)
        os.system("clear")
        print(lawn)
        print(Fore.MAGENTA, lawn.count_flowers(), Style.RESET_ALL)
        if flag == True:
            break
```

files