

ICS 313 Homework 6

This shows the code and the outcome for both part 1 and part 2 of the assignment

```
In [1]: #assignment: ICS 313 Homework 6
#description: a random generator based on the command line argument
#author: Shin Saito

#import the packages necessary for the file
import sys
import random as rd
import string

if len(sys.argv) < 2:
    print(rd.randint(0,100))
    exit(0)
elif len(sys.argv) == 2:
    if not sys.argv[1].isdigit():
        print('error: must be -s, -c, integer or a blank input')
        exit(0)
    a = int(sys.argv[1])
    print(rd.randint(0,a))
    exit(0)
elif len(sys.argv) == 3:
    if sys.argv[1] == '-s':
        a = int(sys.argv[2])
        letters = [i for i in string.ascii_lowercase]
        letterOutput = rd.sample(letters, a)
        print(' '.join(letterOutput))
        exit(0)
    elif not sys.argv[1].isdigit():
        print('error: must be -s, -c, integer or a blank input')
        exit(0)
    elif sys.argv[1].isdigit():
        a = int(sys.argv[1])
        if sys.argv[2].isdigit():
            print(rd.randint(a, int(sys.argv[2])))
            exit(0)
        else:
            print('error: input must be an integer')
            exit(0)
elif sys.argv[1] == '-c':
    if sys.argv[2].isdigit():
        a = int(sys.argv[2])
        for c in range(a):
            if sys.argv[3] == '-s':
                b = int(sys.argv[4])
                letters = [i for i in string.ascii_lowercase]
                letterOutput = rd.sample(letters, b)
                print(' '.join(letterOutput))
            elif sys.argv[3].isdigit():
                if sys.argv[4].isdigit():
                    b = int(sys.argv[3])
                    d = int(sys.argv[4])
                    print(rd.randint(b,d))
                else:
                    print(rd.randint(0,100))
            elif not sys.argv[2].isdigit():
                print('Error: -c must follow with a correct format')
                exit(0)
        exit(0)
elif sys.argv[1] == '-p':
```

```
del sys.argv[1]
print(sys.argv)
inputList = sys.argv
rd.shuffle(inputList)
for i in range(len(inputList)):
    print(inputList[i])
exit(0)
```

error: must be -s, -c, integer or a blank input

Terminal Outcome

In [2]: `from IPython.display import Image`
`Image(url= "https://github.com/saitoshi/ICS313/blob/main/HW6/random313%20terminal.png")`

```
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py -p hello world I am a random program
['hello', 'world', 'I', 'am', 'a', 'random', 'program']
hello
am
program
a
world
I
random
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py -c 3 -s 5
q f v l b
z v c o n
e x p l v
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py -s 3
r u a
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py 1
0
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py
16
uhx02:/home/s/saitoshi/313/hw6% python3 random313.py 5 100
38
```

Part 2 of the Assignment: Create a random plotter bar graph

In [3]: `from collections import Counter`
`import numpy as np`
`import matplotlib.pyplot as plt`
`import random as rd`

```
count = 0
numbers = []
fig = plt.figure(figsize=(15, 15))
ax = fig.add_axes([0,0,1,1])

while (count < 1000):
    numbers.append(rd.randint(0,100))
    count = count + 1

lbl, vals = zip(*Counter(numbers).items())

index = np.arange(len(lbl), 1.0)
start = 0
end = 100
ax.xaxis.set_ticks(np.arange(start, end, 1.0))
ax.set_title('Frequency of Random Generator')
ax.set_xlabel('Number')
ax.set_ylabel('Count')
```

```
ax.bar(lbl,vals)
plt.show()
```

