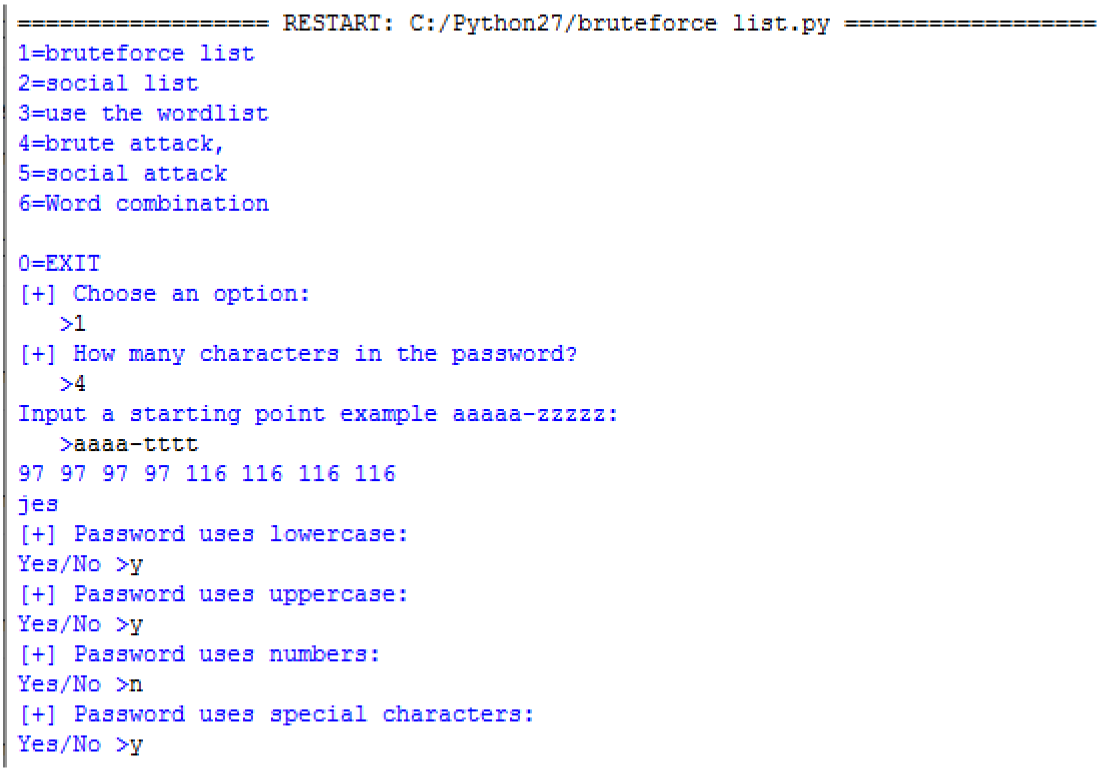
1. Bruteforce list option

You can create a simple bruteforce list by choosing option 1, then the program asks the length of the password in this case it is 4. After the length the program asks for the starting point and end point which are aaaa-tttt. Sadly at this point it ignores the ending point. Then the program ask if the password contains lowercase, uppercase, numbers and special characters. (Figure 1)

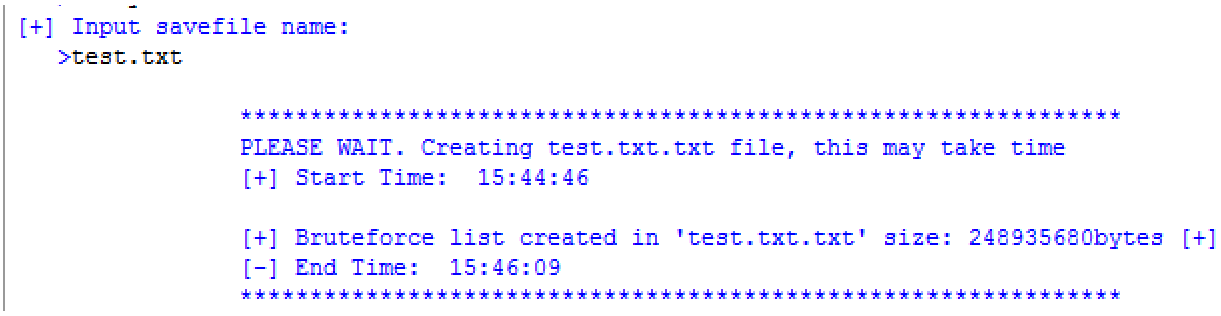


```
================== RESTART: C:/Python27/bruteforce list.py ==================
1=bruteforce list
2=social list
3=use the wordlist
4=brute attack,
5=social attack
6=Word combination

0=EXIT
[+] Choose an option:
   >1
[+] How many characters in the password?
   >4
Input a starting point example aaaaa-zzzzz:
   >aaaa-tttt
97 97 97 97 116 116 116 116
jes
[+] Password uses lowercase:
Yes/No >y
[+] Password uses uppercase:
Yes/No >y
[+] Password uses numbers:
Yes/No >n
[+] Password uses special characters:
Yes/No >y
```

Figure 1 program 1 part

Then it asks for the file name where the password will be saved and starts outputting the different combinations for the chosen parameters. The program shows starting time and the ending time. (Figure 2)



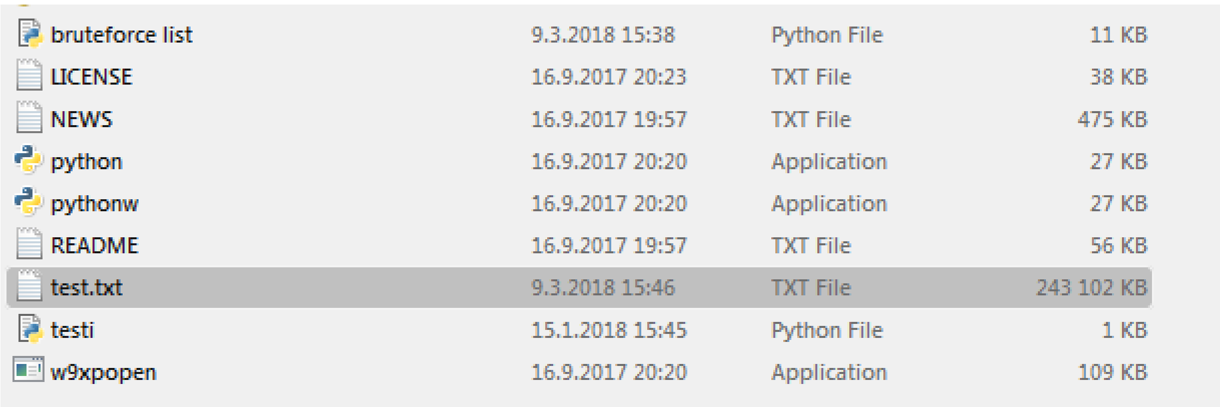
```
[+] Input savefile name:
   >test.txt

                ****************************************************************
                PLEASE WAIT. Creating test.txt.txt file, this may take time
                [+] Start Time:  15:44:46

                [+] Bruteforce list created in 'test.txt.txt' size: 248935680bytes [+]
                [-] End Time:  15:46:09
                ****************************************************************
```

Figure 2 program 2 part

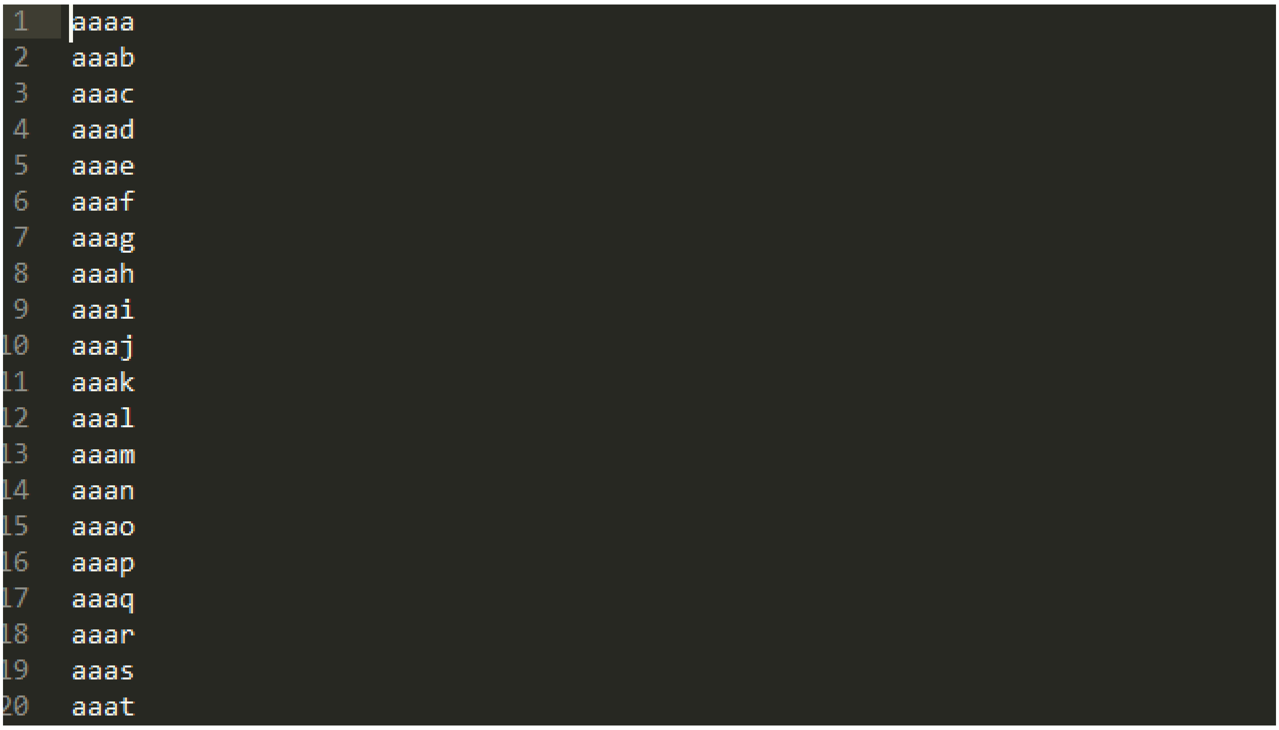
As seen below the program saved the file in to the same folder where the program was located. (Figure 3)



bruteforce list	9.3.2018 15:38	Python File	11 KB
LICENSE	16.9.2017 20:23	TXT File	38 KB
NEWS	16.9.2017 19:57	TXT File	475 KB
python	16.9.2017 20:20	Application	27 KB
pythonw	16.9.2017 20:20	Application	27 KB
README	16.9.2017 19:57	TXT File	56 KB
test.txt	9.3.2018 15:46	TXT File	243 102 KB
testi	15.1.2018 15:45	Python File	1 KB
w9xpopen	16.9.2017 20:20	Application	109 KB

Figure 3 created file

Looking inside the file shows all the different combinations (Figure 4)



```
aaaa
aaab
aaac
aaad
aaae
aaaf
aaag
aaah
aaai
aaaj
aaak
aaal
aaam
aaan
aaao
aaap
aaaq
aaar
aaas
aaat
```

Figure 4 file