Rar password brute-force cracker in Python

Group Members

S/N	**NAME**	**MATRICULE**
1	Wepngong Ngeh Benaiah	FE12A191
2	Esambe Elvis Njume	FE12A056
3	Nkeangnyi Tonia	FE12A138
4	Kingue Patrick	FE12A087
5	Takoungang Dieudonne	FE12A172
6	Naoussi Martial	FE12A1

Synopsis

In cryptography, a **brute-force attack**, or **exhaustive key search**, is a **cryptanalytic** attack that can, in theory, be used against any encrypted data. Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier. It consists of systematically checking all possible **keys** or **passwords** until the correct one is found. In the worst case, this would involve **traversing the entire search space**.

When password guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because of the time a brute-force search takes.

Brute force attacks work by calculating every possible combination that could make up a password and testing it to see if it is the correct password. As the password’s length increases, the amount of time, on average, to find the correct password increases exponentially. This means short passwords can usually be discovered quite quickly, but longer passwords may take decades.

The **pyrar_cracker.py** uses a brute-force attack where all possible keys from the search space

(**1234567890aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ**) are tested against the file. When a key matches, it displays it on the screen, together with the running time and exits. For the search space given with 52 characters, in the worst case could take

52!*password_length.

System Requirements

Hardware Requirements

- 4Gb RAM
- 3GHz or higher
- 3GB hard drive

Software Requirements

- **Python2.7 or higher**
- **unrar**
- **pzip**

How to run pyrar_cracker.py

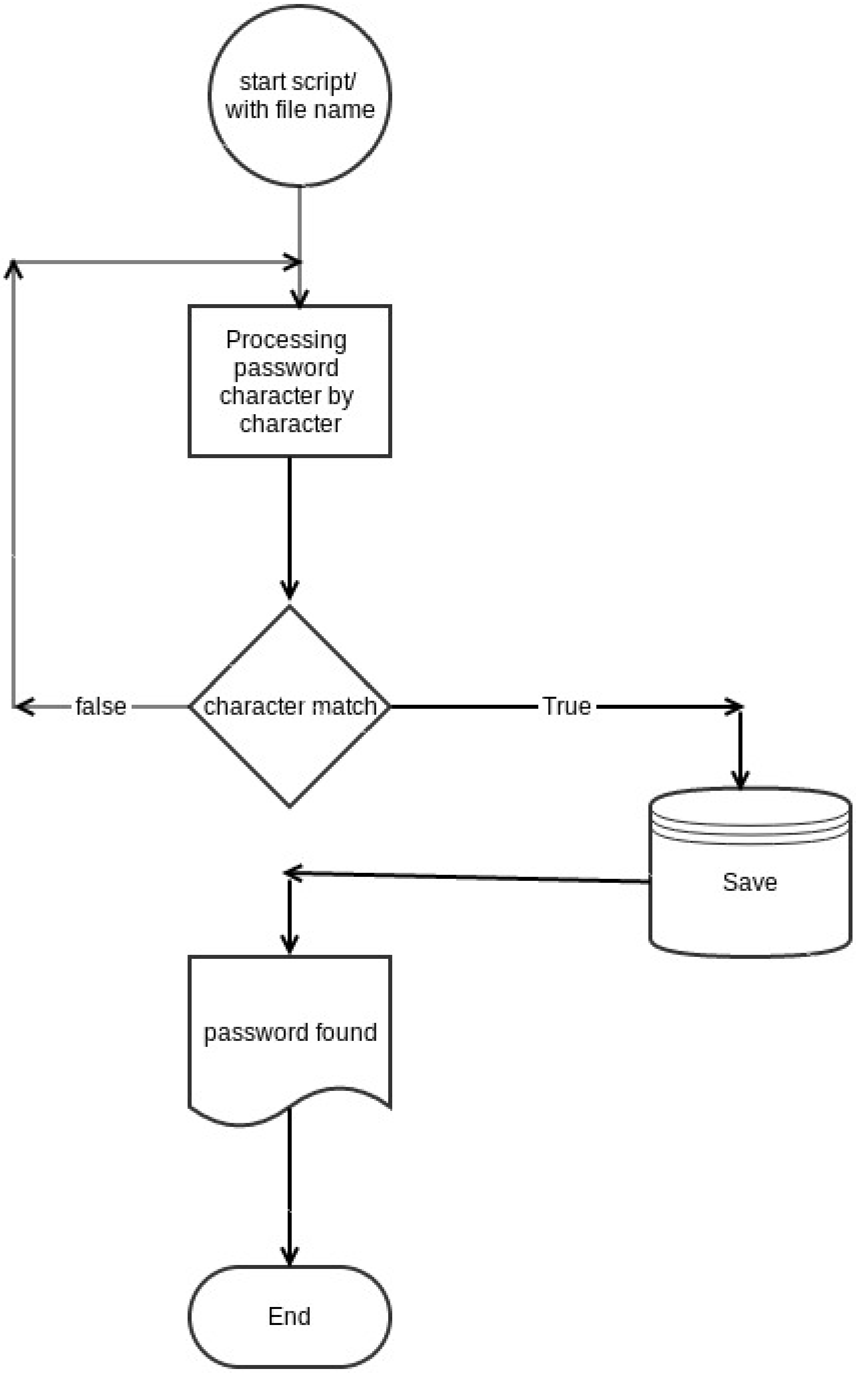
Open a terminal window in any *nix distribution and type the following command.

```
bwepngong@bwepngong-s-A665D:$ python3.4  pyrar_cracker.py <<file to crack.rar>>
```

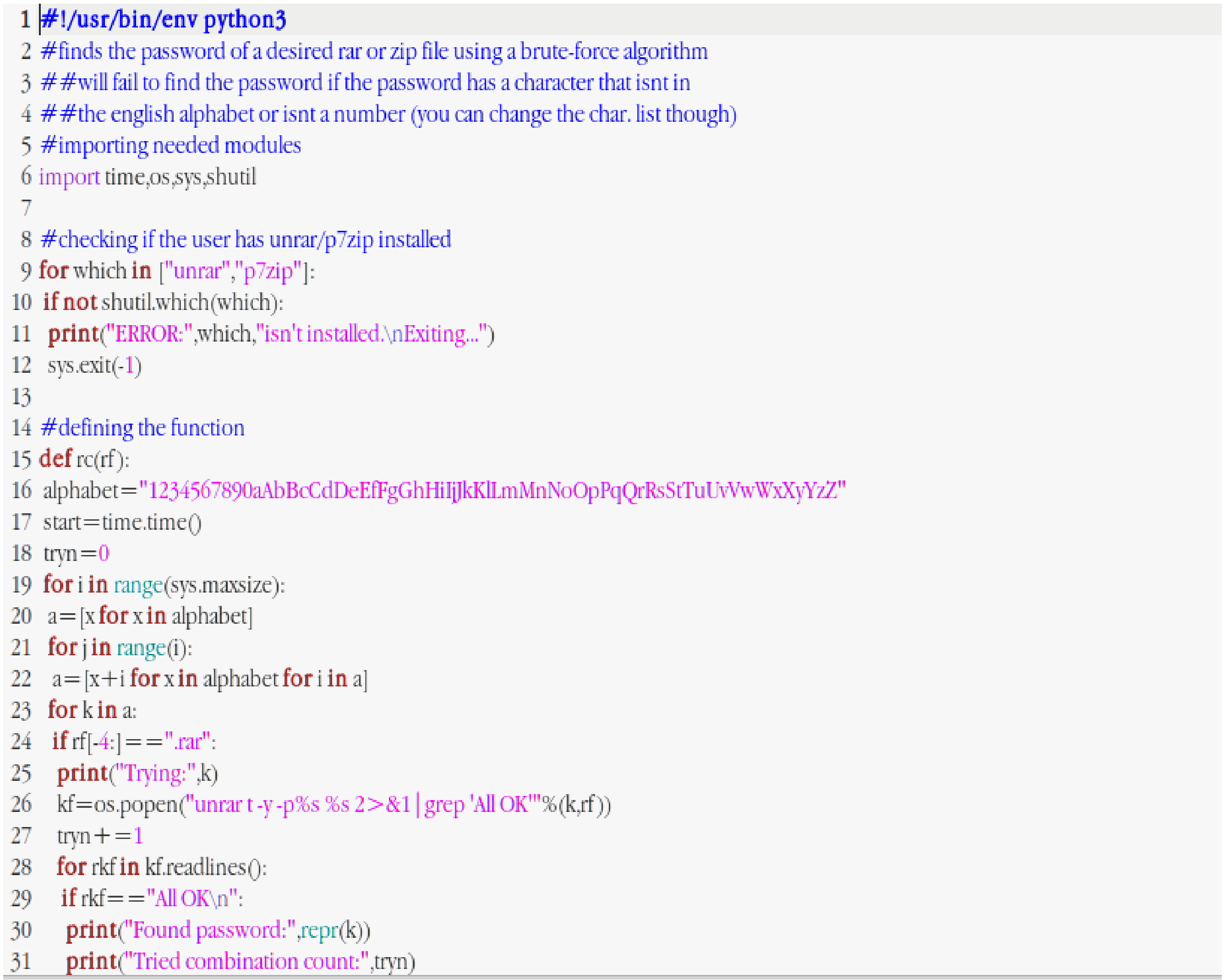
Have a cup of coffee or 20 of them and wait for it to crack your password.

As earlier said, the cracking process could take from hours to years to finish.

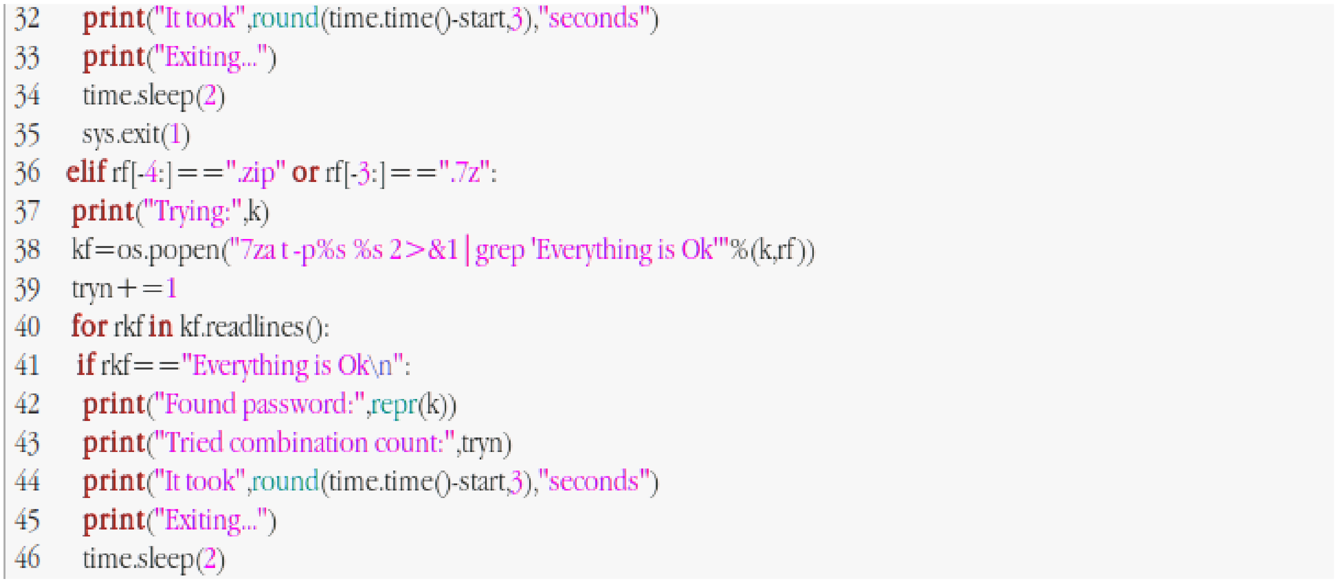
Flow Chart


start script/ with file name
Processing password character by character
character match
false
True
Save
password found
End

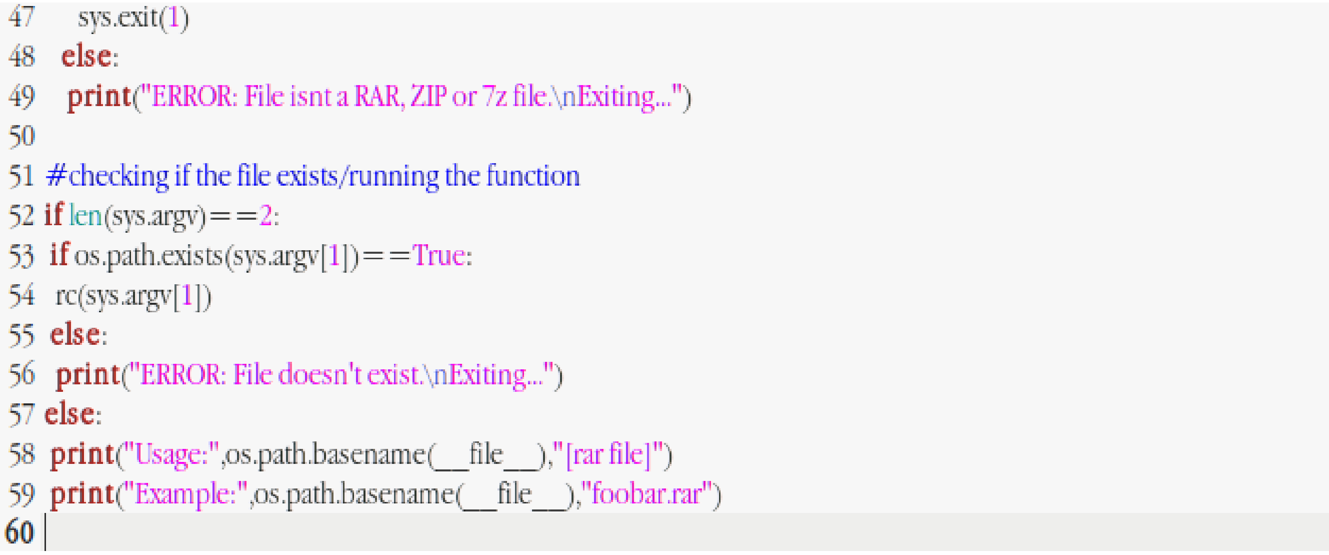
Source Code



```
#!/usr/bin/env python3
#finds the password of a desired rar or zip file using a brute-force algorithm
##will fail to find the password if the password has a character that isnt in
##the english alphabet or isnt a number (you can change the char. list though)
#importing needed modules
import time,os,sys,shutil

#checking if the user has unrar/p7zip installed
for which in ["unrar","p7zip"]:
 if not shutil.which(which):
  print("ERROR:",which,"isn't installed.\nExiting...")
  sys.exit(-1)

#defining the function
def rc(rf):
 alphabet="1234567890aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ"
 start=time.time()
 tryn=0
 for i in range(sys.maxsize):
  a=[x for x in alphabet]
  for j in range(i):
   a=[x+i for x in alphabet for i in a]
  for k in a:
   if rf[-4:]==".rar":
    print("Trying:",k)
    kf=os.popen("unrar t -y -p%s %s 2>&1 | grep 'All OK'"%(k,rf))
    tryn+=1
    for rkf in kf.readlines():
     if rkf=="All OK\n":
      print("Found password:",repr(k))
      print("Tried combination count:",tryn)
```



```
      print("It took",round(time.time()-start,3),"seconds")
      print("Exiting...")
      time.sleep(2)
      sys.exit(1)
   elif rf[-4:]==".zip" or rf[-3:]==".7z":
    print("Trying:",k)
    kf=os.popen("7za t -p%s %s 2>&1 | grep 'Everything is Ok'"%(k,rf))
    tryn+=1
    for rkf in kf.readlines():
     if rkf=="Everything is Ok\n":
      print("Found password:",repr(k))
      print("Tried combination count:",tryn)
      print("It took",round(time.time()-start,3),"seconds")
      print("Exiting...")
      time.sleep(2)
```

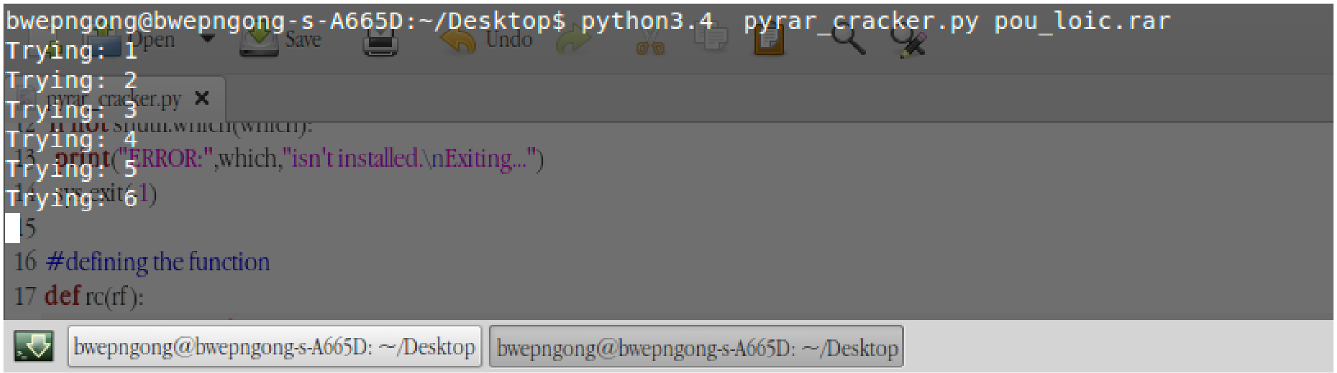


```
    sys.exit(1)
  else:
   print("ERROR: File isnt a RAR, ZIP or 7z file.\nExiting...")

#checking if the file exists/running the function
if len(sys.argv)==2:
 if os.path.exists(sys.argv[1])==True:
  rc(sys.argv[1])
 else:
  print("ERROR: File doesn't exist.\nExiting...")
else:
 print("Usage:",os.path.basename(__file__),"[rar file]")
 print("Example:",os.path.basename(__file__),"foobar.rar")

```

Output

Run from an Ubuntu 14.04 terminal, this is the output

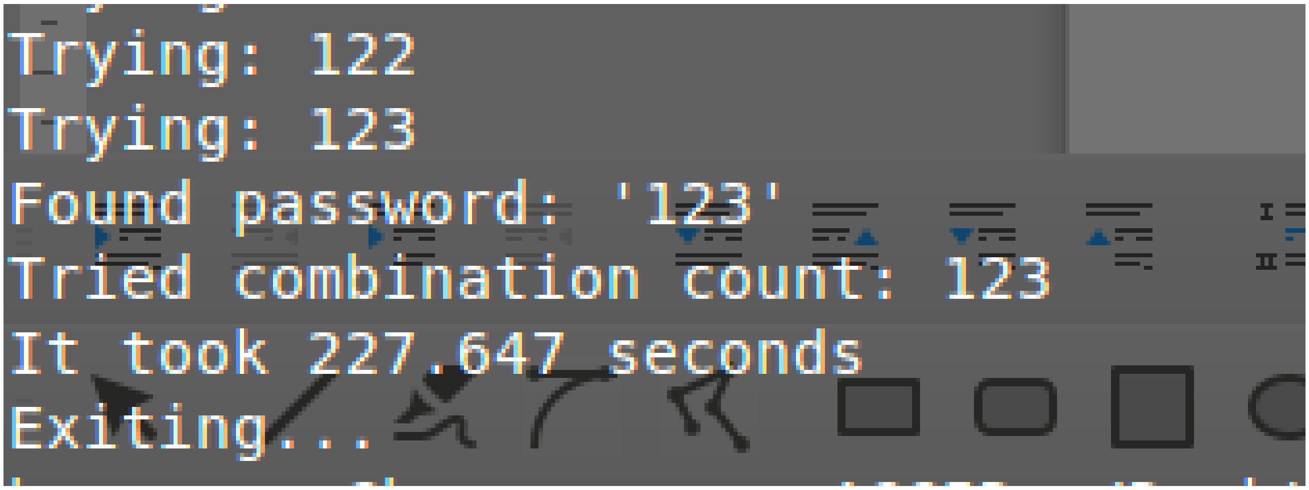
At the beginning



```
bwepngong@bwepngong-s-A665D:~/Desktop$ python3.4  pyrar_cracker.py pou_loic.rar
Trying: 1
Trying: 2
Trying: 3
Trying: 4
Trying: 5
Trying: 6
```

Open
Save
Undo
pyrar_cracker.py
bwepngong@bwepngong-s-A665D: ~/Desktop
bwepngong@bwepngong-s-A665D: ~/Desktop

Password found after **227.69 seconds** as 123. For testing purposes, the alphabet was reduced to **‘1234567890’.**



```
Trying: 122
Trying: 123
Found password: '123'
Tried combination count: 123
It took 227.647 seconds
Exiting...
```