TECHNION
Azrieli Continuing Education and External Studies Division

Python Exam

Mission 1: Multiple-choice Questions (30 pts)

Choose the correct answer and describe it with one line sentence.
Please submit the question number and the answer letter.

1.1. What is the output of the following Python code?

```
abc = "word213"
for i in abc:
    if abc.isdigit():
        print(i)
    else:
        break
```

a) 2 1 3
b) Nothing
c) w o r d 2 1 3
d) Error

1.2. What does **pip** stand for Python?

a) Unique pipeline module
b) Syntax checker
c) One of the debugging options
d) Preferred installer of modules

1.3. What is the output of the following Python code?

```
def notlikemath(a,b,c):
    return (a+b)*c

print(notlikemath(24,56,1))
```

a) Error - functions usually print out their values
b) Nothing - the function doesn't contain print functionality
c) 102
d) 80

1.4. What is the value stored inside variable **longs**?

```
longs = len(["hello",2, 4, 6])
```

a) 8
b) 5
c) 4
d) 1

1.5. What is the output of the following Python code?

```
word = "we dont fault exams"
print(word.replace("e","X",1)[8::])
```

a) wX dont fault
b) dont fault Xxams
c) fault exams
d) smaxX fluat t

1.6. What is the output of the following Python code?

```
wait = "indexes"
print(wait.index("e")+wait.index("e"))
```

a) 3
b) 8
c) 6
d) ee

Mission 2: Snippets (30 pts)

Provide a code snippet or script file that will follow the next tasks
Please submit the code in a txt file or python file - you could merge all scripts in the same file as well.

2.1. Write a program that takes the next list of numbers `[2, 68, 8, 10, 25, 24, 201, 23, 67]` and output the biggest number.

```
nums = [2, 68, 8, 10, 25, 24, 201, 23, 67]
nums.sort()
print(nums[-1])
```

2.2. Write a program that takes the next variable myword = "SearchingPath" and output a dictionary of two keys "upper" when the value is the number of uppercase letters in **myword** and "lower" when the value is the number of lowercase letters in **myword**.

```
myword = "SearchingPath"

upper = 0
lower = 0

for letter in myword:
  if letter.isupper():
    upper += 1
  else:
    lower += 1
```

```
print({"upper": upper, "lower": lower})
```

2.3. Write a program that takes the next variable number = 10 and print out the sum of the range of numbers until it.

Examples:
***number = 5** then the output should be **1+2+3+4+5 = 15**.*
***number = 2** then the output should be **1+2 = 3**.*

```
number = 10

x = 0

for n in range(number + 1):
  x += n

print(x)
```

Mission 3: Programming (40 pts)

In the next mission you will write a complicated program, please provide the full script file as answer.
Please submit the python script in .py or .txt file + the better1050.txt file that you are creating in advance.

Provided with the next passwords list:

https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/best1050.txt

Task:
Write a program that will make a new and more valuable wordlist from the given **best1050.txt** file.

- Name the new file "better1050.txt"
- The file will hold only a password with 8 characters. (Length = 8)
- The file will hold only a password that's not all its characters are digits. (Ex. "12346" = Not Suitable)
- The file will hold only a password that's not all its characters are letters. (Ex. "jenia" = Not Suitable)
  Suitable password example: "**password12**", "**123456aA**",etc.

- Add to the script a summary ability:
  a. The script will work with the new file "better1050.txt"
  b. When the script finish running it will output "I collect only 100 passwords!"
  c. When the script finish running it will output "The passwords classified contain: 81 letters, 10 digits and 1 special character"

Notes:
- Feel free to use any technique you like
- Functions could only help, but it's not a **demand**

- Before writing the script, plan it!
- Points for this mission will be calculate for **non-completed program** as well
- Hint: use dictionaries to make it easier
- Try to do your best

```
def read(file):
  with open(file, 'r') as f:
    x = f.read()
    return x.split()


best1050 = read('best1050.txt')

with open('better1050.txt', 'w') as b:
  for word in best1050:
    if len(word) == 8 and not word.isdigit() and not word.isalpha():
      b.write(word + "\n")

better = 'better1050.txt'

final_list = read(better)
legnth = len(final_list)
letters = 0
digits = 0
special = 0

for word in final_list:
  for letter in word:
    if letter.isalpha():
      letters += 1
    elif letter.isdigit():
      digits += 1
    else:
      special += 1


print(f"I collect only {legnth} passwords!")
print()
print(f"The passwords classified contain: {letters} letters, {digits} digits
 and {special} special character!")
print()
for word in final_list:
  print(word)
```