



Middle East Technical University, NCC
CNG111 - Introduction to Computer Engineering Concepts
Fall 2014 - 2015
Assignment I - 100 pts

Due Date: 23.11.14, midnight

1. (5 points) Write a script that takes a number from the user and prints out the number of digits of this number. The sign of a number should not change the number of its digits.

Sample runs (User input is shown in bold, the `#$ python X.py` line is where the execution begins.)

```
#$ python q1.py
```

```
Enter> 1231
```

```
4
```

```
#$ python q1.py
```

```
Enter> -90071
```

```
5
```

2. (10 points) Write a script that prints out the values the function $f(x) = x^2 + 4x - 4$ takes over a desired interval and with a specified step size.

Sample runs (User input is shown in bold, the `#$ python X.py` line is where the execution begins.)

```
#$ python q2.py
```

```
Enter the first value of x> 25
```

```
Enter the last value of x> 35
```

```
Enter the step size> 5
```

```
f(25) = 721
```

```
f(30) = 1016
```

```
f(35) = 1361
```

```
#$ python q2.py
```

```
Enter the first value of x> 1
```

```
Enter the last value of x> 2
```

```
Enter the step size> 0.28
```

```
f(1) = 1
```

```
f(1.28) = 2.7584
```

```
f(1.56) = 4.6736
```

```
f(1.84) = 6.7456
```

3. (10 points) Given a number, print all unique prime factors of that number.

Sample runs (User input is shown in **bold**, the **#\$ python X.py** line is where the execution begins.)

```
#$ python q3.py
```

```
Enter> 45
```

```
3
```

```
5
```

```
#$ python q3.py
```

```
Enter> 468225576
```

```
2
```

```
3
```

```
7
```

```
13
```

```
41
```

```
83
```

4. (15 points) Write a script to find the smallest number with at least 6 unique prime factors that gives a remainder of 3 when divided by 133.
5. (15 points) Write a script to find the first of the four smallest consecutive numbers with exactly four unique prime factors.
6. (15 points) A number is called non-increasing number if none of its digits is greater than any digit to the left of it. For example, 53, 222, 300, 732, 8653, 9220, 88644, 754330, and 766640 are all non-increasing numbers. Write a script that determines whether a given number is non-increasing or not.

Sample runs (User input is shown in **bold**, the **#\$ python X.py** line is where the execution begins.)

```
#$ python q5.py
```

```
Enter> 4561
```

```
4561 is NOT non-increasing
```

```
#$ python q5.py
```

```
Enter> 987000
```

```
987000 is non-increasing
```

```
#$ python q5.py
```

```
Enter> 76322
```

```
76322 is non-increasing
```

7. (15 points) Write a script that finds the smallest positive integer k such that the ratio of non-increasing numbers to all numbers between 1 and k (inclusive) is less than 0.001. For example, the smallest number for which this ratio goes below 0.01 is 302501.

8. (15 points) Write the following algorithm in Python. The algorithm calculates the square root of a value. Add print statements to your code to produce the same output for the given sample inputs below.

1. Let n be a number you get from the user.
2. Let i be 0.0.
3. Double the value of j until $j^2 > n$.
4. Let $g = \frac{i+j}{2}$ be our guess.
5. Let d be the absolute value of $g^2 - n$.
6. If $d < 0.001$, then the output is g .
7. If $g^2 < n$, let $i = g$.
8. If $g^2 > n$, let $j = g$.
9. Go to step 4.

Sample runs (User input is shown in **bold**, the **#\$ python X.py** line is where the execution begins.)

```
#$ python q4.py
```

```
Enter> 13
Seeking the square root of 13
For j = 4.0, square of j exceeds 13
i = 0.0, j = 4.0, g = 2.0
i = 2.0, j = 4.0, g = 3.0
i = 3.0, j = 4.0, g = 3.5
i = 3.5, j = 4.0, g = 3.75
i = 3.5, j = 3.75, g = 3.625
i = 3.5, j = 3.625, g = 3.5625
i = 3.5625, j = 3.625, g = 3.59375
i = 3.59375, j = 3.625, g = 3.609375
i = 3.59375, j = 3.609375, g = 3.6015625
The square root of 13 is 3.60546875
```

```
#$ python q4.py
```

```
Enter> 2025
Seeking the square root of 2025
For j = 64.0, square of j exceeds 2025
i = 0.0, j = 64.0, g = 32.0
i = 32.0, j = 64.0, g = 48.0
i = 32.0, j = 48.0, g = 40.0
i = 40.0, j = 48.0, g = 44.0
i = 44.0, j = 48.0, g = 46.0
The square root of 2025 is 45.0
```

How to submit

1. Write each individual answer in a separate file. Put your answer for the question 1 in q1.py and question 2 in q2.py...
2. Create a zip file containing all your answers and name this zip file as **CNG111-ASN1-ID-20141.zip** where **ID** is your COMPLETE (7-digit) student ID.
3. Upload your zip file via <http://dropitto.me/cbasaran>
4. You will need to enter the password: **Metu_NCC_Fall_2014** to submit your assignment.
5. The deadline is final, try to solve as much questions as you can until the deadline.
6. Plagiarism policy will be strictly enforced.

Happy Coding :)