

# 전산천문학 HW1 Solution

March 30, 2020

## 1.

```
[2]: x1, x2 = input('Input two Integers : ').split() # 두 정수를 입력 받음 / Space키를
      이용하여 두 정수를 구분
      x1, x2 = [int(x1), int(x2)] # 입력받은 데이터를 정수형(int)로 변환
      print ('Two Integers : %i %i' % (x1, x2))

      a,b = max([x1,x2]), min([x1,x2]) # 입력 받은 두 정수의 대소 비교

      quot=a//b # 몫 계산
      remain=a%b # 나머지 계산

      print ('Quotient : %i' % quot)
      print ('Remainder : %i' % remain)
```

```
Input two Integers : 78 1023
Two Integers : 78 1023
Quotient : 13
Remainder : 9
```

## 2.

```
[3]: size = input('Input the size : ') # H의 사이즈를 입력받음
      size = int(size) # 입력받은 데이터를 정수형(int)로 변환

      star,space='*', ' ' # H를 표시하는 데에 사용할 문자열 데이터
      for i in range(3*size):
          if i<size or i>=2*size:
              print (size*star+size*space+size*star) #H의 윗 부분과 아랫부분
          else:
              print ((3*size)*star) # H의 중간 부분
```

```
Input the size : 4
****      ****
****      ****
****      ****
```

```

****      ****
*****
*****
*****
*****
****      ****
****      ****
****      ****
****      ****

```

### 3.

```

[4]: def prime_check(n):
    if n<2: # 0,1은 소수가 아님
        return False
    elif n==2: # 2는 유일한 짝수 소수
        return True
    elif n%2==0: # 2를 제외한 나머지 짝수는 소수에서 제외
        return False
    for i in range(3,int(pow(n,0.5))+1,2): # sqrt(n)까지의 홀수에 대해서 나누어떨어
        지는지를 확인
        if n%i==0:
            return False
    return True

nmax = [100,1000,10000,100000]

for nmax_i in nmax:
    prime_count = 0
    for i in range(1,nmax_i+1):
        if prime_check(i)==True:
            prime_count+=1
    print ('Number of prime numbers within %i : %i' % (nmax_i,prime_count))

```

```

Number of prime numbers within 100 : 25
Number of prime numbers within 1000 : 168
Number of prime numbers within 10000 : 1229
Number of prime numbers within 100000 : 9592

```

### 4.

```

[5]: import numpy as np

data = np.loadtxt('hw1_p4.dat', delimiter='\t')
sorted_data = data[0::4] # 첫번째 행부터 시작하여 4번째 행마다 데이터를 따로 뽑아냄
→[start:stop:step]

```

```
np.savetxt('hw1_p4_fourth.dat', sorted_data, delimiter='\t') # 뽑아낸 데이터를 저장
```

```
[7]: import numpy as np
import matplotlib.pyplot as plt

data_fourth = np.loadtxt('hw1_p4_fourth.dat', delimiter = '\t')
x,a,b = data_fourth[:,0], data_fourth[:,1], data_fourth[:,2]

plt.figure(figsize=(10,6))
plt.plot(x,a*x,label='ax')
plt.plot(x,b*x,ls=':',label='bx')
plt.plot(x,a*b*x,ls='--',label='abx')
plt.xlabel('Time',fontsize=15)
plt.ylabel('Value',fontsize=15)
plt.xlim(0.0,20.0)
plt.ylim(-4.,4.)
plt.legend(loc=2,fontsize=14)
plt.tight_layout()
```

