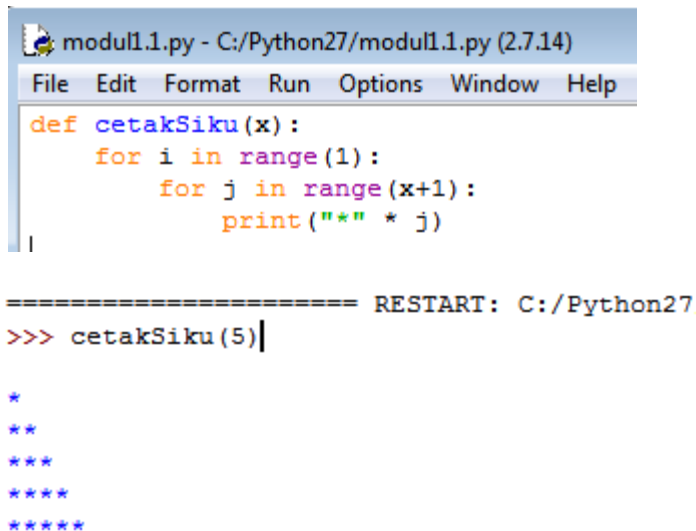


MODUL 1	
NAMA	DIMAS FAJAR SAPUTRO
NIM	L200160090
KELAS	D

SOAL

1. Membuat suatu fungsi cetakSiku(x) dengan nilai x menunjukkan segitiga itu dan gunakan perulangan dua kali.



```

modul1.1.py - C:/Python27/modul1.1.py (2.7.14)
File Edit Format Run Options Window Help

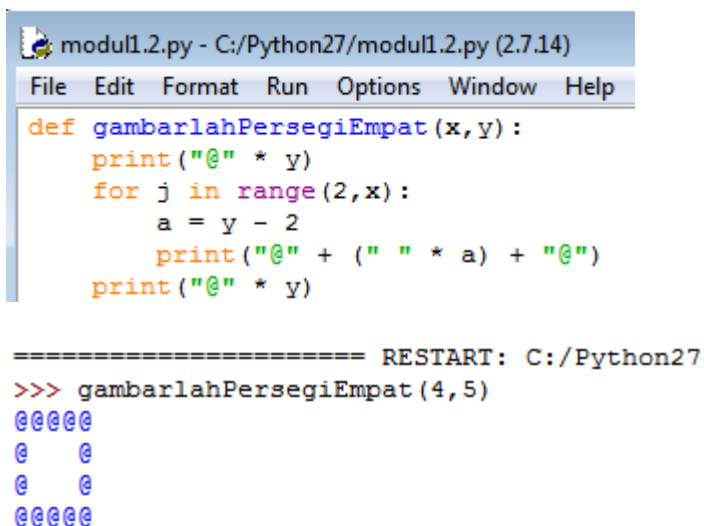
def cetakSiku(x):
    for i in range(1):
        for j in range(x+1):
            print("*" * j)

===== RESTART: C:/Python27
>>> cetakSiku(5)

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

```

2. Membuat sebuah fungsi yang menerima dua integer positif, yang akan menggunakan bentuk persegiempat.



```

modul1.2.py - C:/Python27/modul1.2.py (2.7.14)
File Edit Format Run Options Window Help

def gambarlahPersegiEmpat(x,y):
    print("@" * y)
    for j in range(2,x):
        a = y - 2
        print("@" + (" " * a) + "@")
    print("@" * y)

===== RESTART: C:/Python27
>>> gambarlahPersegiEmpat(4,5)
@@@@@
@    @
@    @
@    @
@@@@@

```

3. Membuat 2 buah fungsi:

- Fungsi menerima sebuah fungsi untuk menghitung jumlah semua huruf dan huruf vokal saja dengan keluaran berupa list.

```
modul1.31.py - C:/Python27/modul1.31.py (2.7.14)
File Edit Format Run Options Window Help

def jumlahHurufVokal(x):
    vokal = ["A", "I", "U", "E", "O"]
    a = ""
    for i in x.upper():
        if i in vokal:
            a += i
    listHasil = (len(x), len(a))
    print(listHasil)
```

```
===== RESTART: C:/Pytl
>>> jumlahHurufVokal("Surakarta")
(9, 4)
>>> jumlahHurufVokal("UMS")
(3, 1)
```

- Fungsi menerima sebuah fungsi untuk menghitung jumlah semua huruf dan huruf konsonan saja dengan keluaran berupa list

```
modul1.31.py - C:/Python27/modul1.31.py (2.7.14)
File Edit Format Run Options Window Help

def jumlahHurufKonsonan(x):
    konsonan = ["q", "w", "r", "t", "y", "p", "s",
                "d", "f", "g", "h", "j", "k", "l",
                "z", "x", "c", "v", "b", "n", "m"]
    a = ""
    for i in x.lower():
        if i in konsonan:
            a += i
    listHasil = (len(x), len(a))
    print(listHasil)
```

```
===== RESTART: C:/Pytl
>>> jumlahHurufKonsonan("Surakarta")
(9, 5)
>>> jumlahHurufKonsonan("UMS")
(3, 2)
```

4. Membuat sebuah fungsi menghitung rerata sebuah array

```
modul1.4.py - C:/Python27/modul1.4.py (2.7.14)
File Edit Format Run Options Window Help

def rerata(x):
    a = 0
    for i in range(len(x)):
        a += x[i]
    hasil = a / len(x)
    print(hasil)
```

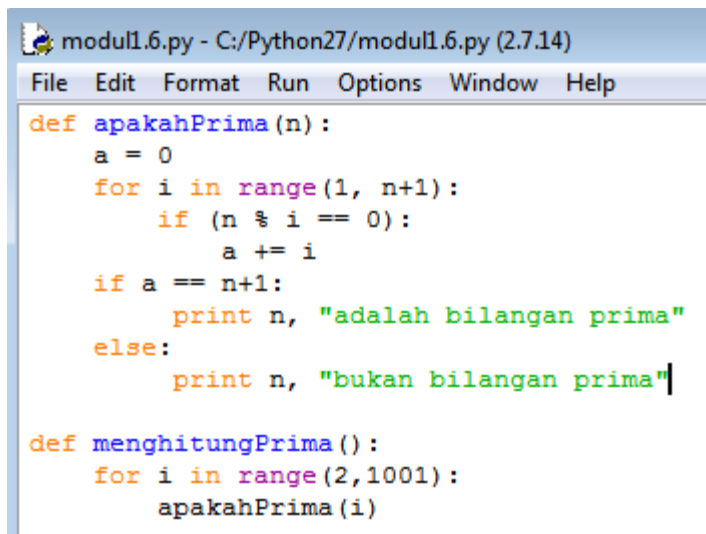
```
===== RESTART: C:/F
>>> rerata([1,2,3,4,5,6,7,8,9,10])
5
```

5. Membuat sebuah fungsi untuk menentukan apakah suatu bilangan merupakan bilangan prima atau bukan.

```
def apakahPrima(n):
    a = 0
    for i in range(1, n+1):
        if (n % i == 0):
            a += i
    if a == n+1:
        print n, "adalah bilangan prima"
    else:
        print n, "bukan bilangan prima"
```

```
===== RESTART: C:
>>> apakahPrima(200)
200 bukan bilangan prima
>>> apakahPrima(23)
23 adalah bilangan prima
```

6. Membuat suatu program untuk mencetak semua bilangan prima dari 2 sampai 1000 dengan memanfaatkan apakahPrima(n)



```
modul1.6.py - C:/Python27/modul1.6.py (2.7.14)
File Edit Format Run Options Window Help

def apakahPrima(n):
    a = 0
    for i in range(1, n+1):
        if (n % i == 0):
            a += i
    if a == n+1:
        print n, "adalah bilangan prima"
    else:
        print n, "bukan bilangan prima"

def menghitungPrima():
    for i in range(2,1001):
        apakahPrima(i)
```

```

>>> menghitungPrima()
2 adalah bilangan prima
3 adalah bilangan prima
4 bukan bilangan prima
5 adalah bilangan prima
6 bukan bilangan prima
7 adalah bilangan prima
8 bukan bilangan prima
9 bukan bilangan prima
10 bukan bilangan prima
11 adalah bilangan prima
12 bukan bilangan prima
13 adalah bilangan prima
14 bukan bilangan prima
15 bukan bilangan prima
16 bukan bilangan prima
17 adalah bilangan prima
18 bukan bilangan prima
19 adalah bilangan prima
20 bukan bilangan prima
21 bukan bilangan prima
22 bukan bilangan prima
23 adalah bilangan prima
24 bukan bilangan prima
25 bukan bilangan prima
26 bukan bilangan prima
27 bukan bilangan prima
28 bukan bilangan prima
29 adalah bilangan prima
30 bukan bilangan prima
31 adalah bilangan prima
971 adalah bilangan prima
972 bukan bilangan prima
973 bukan bilangan prima
974 bukan bilangan prima
975 bukan bilangan prima
976 bukan bilangan prima
977 adalah bilangan prima
978 bukan bilangan prima
979 bukan bilangan prima
980 bukan bilangan prima
981 bukan bilangan prima
982 bukan bilangan prima
983 adalah bilangan prima
984 bukan bilangan prima
985 bukan bilangan prima
986 bukan bilangan prima
987 bukan bilangan prima
988 bukan bilangan prima
989 bukan bilangan prima
990 bukan bilangan prima
991 adalah bilangan prima
992 bukan bilangan prima
993 bukan bilangan prima
994 bukan bilangan prima
995 bukan bilangan prima
996 bukan bilangan prima
997 adalah bilangan prima
998 bukan bilangan prima
999 bukan bilangan prima
1000 bukan bilangan prima

```

7. Membuat suatu program yang mencetak faktorisasi prima dari suatu bilangan.

```

modul1.7.py - C:/Python27/modul1.7.py (2.7.14)
File Edit Format Run Options Window Help

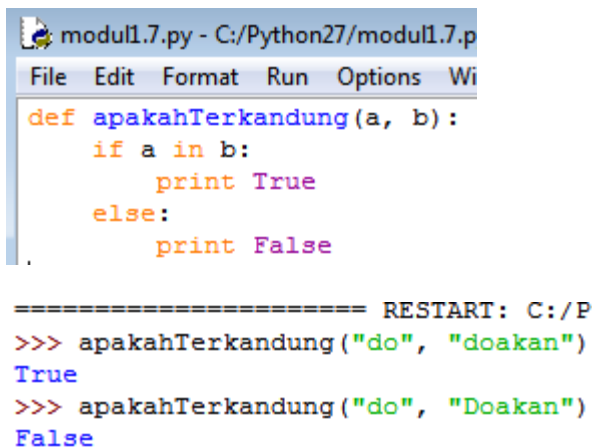
def faktorPrima(x):
    a = []
    b = 2
    while b <= x:
        if x % b == 0:
            a.append(b)
            x /= b
        else:
            b += 1
    print a

===== ]

>>> faktorPrima(20)
[2, 2, 5]\
>>> faktorPrima(21)
[3, 7]
>>> faktorPrima(50)
[2, 5, 5]
>>> faktorPrima(100)
[2, 2, 5, 5]

```

8. Membuat suatu fungsi yang menerima 2 buah input berupa string dan apakah input a berada di dalam input b.

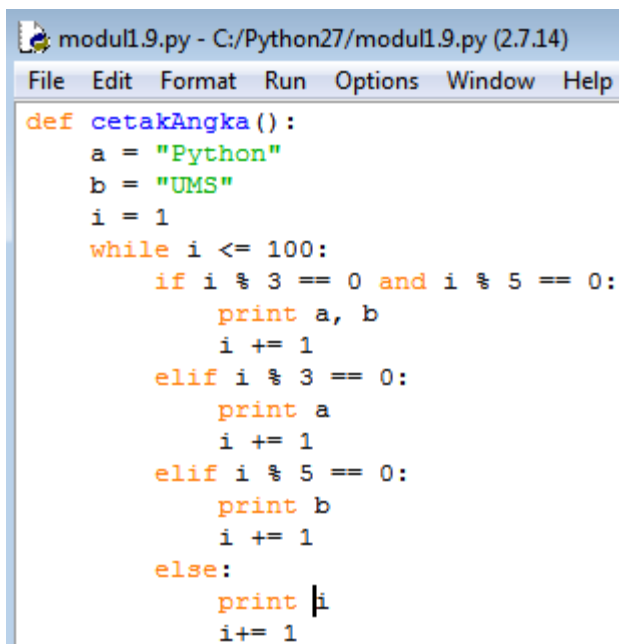


```
modul1.7.py - C:/Python27/modul1.7.p
File Edit Format Run Options Wi

def apakahTerkandung(a, b):
    if a in b:
        print True
    else:
        print False

===== RESTART: C:/P
>>> apakahTerkandung("do", "doakan")
True
>>> apakahTerkandung("do", "Doakan")
False
```

9. Membuat program untuk mencetak angka dari 1 sampai 100 dengan output jika angka kelipatan 3, cetak "Python". Kalau kelipatan 5, cetak "UMS". Kalau pas kelipatan 3 sekaligus kelipatan 5, cetak "Python UMS".



```
modul1.9.py - C:/Python27/modul1.9.py (2.7.14)
File Edit Format Run Options Window Help

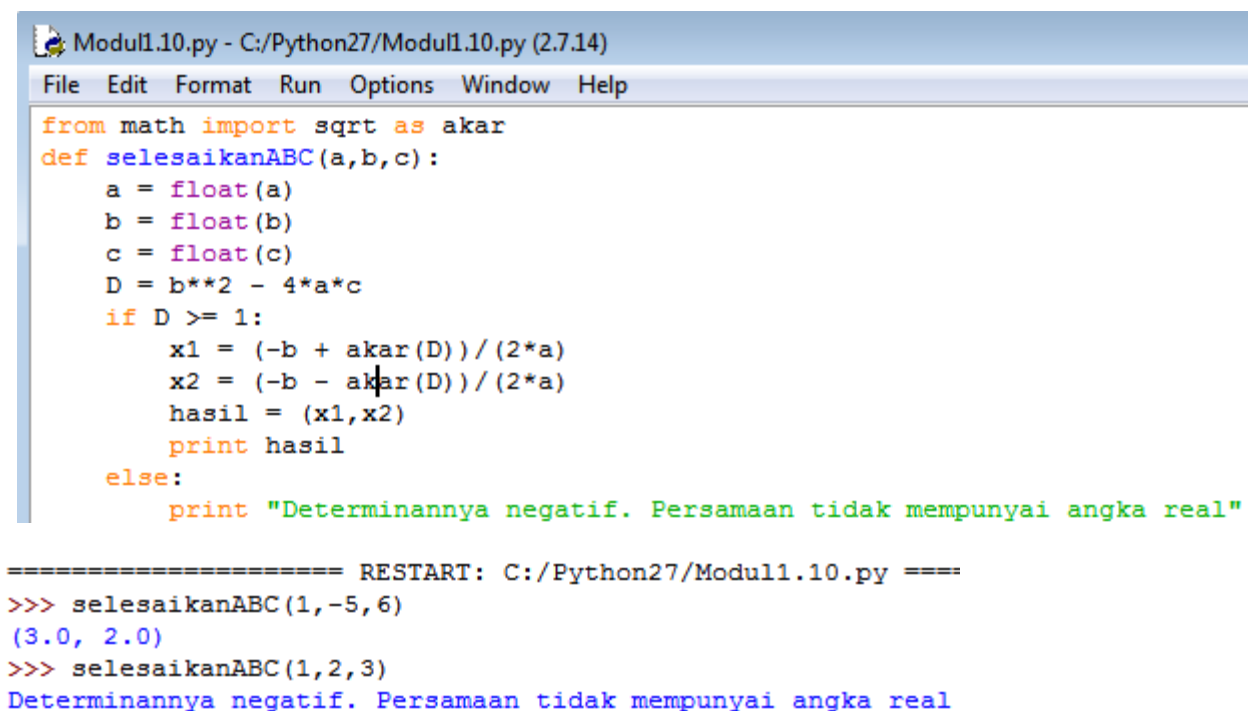
def cetakAngka():
    a = "Python"
    b = "UMS"
    i = 1
    while i <= 100:
        if i % 3 == 0 and i % 5 == 0:
            print a, b
            i += 1
        elif i % 3 == 0:
            print a
            i += 1
        elif i % 5 == 0:
            print b
            i += 1
        else:
            print i
            i += 1
```

```

>>> cetakAngka()
1      Python
2      73
Python 74
4      Python UMS
UMS    76
Python 77
7      Python
8      79
Python UMS
UMS    Python
11     82
Python 83
13     Python
14     UMS
Python UMS 86
16     Python
17     88
Python    89
19     Python UMS
UMS      91
Python   92
22     Python
23     94
Python  UMS
UMS     Python
26     97
Python  98
28     Python
29     UMS
>>> |

```

10. Membuat modifikasi pada contoh 1.4, agar bisa menangkap kasus dimana determinannya kurang dari nol akan menampilkan pesan peringatan.



```

Modul1.10.py - C:/Python27/Modul1.10.py (2.7.14)
File Edit Format Run Options Window Help

from math import sqrt as akar
def selesaikanABC(a,b,c):
    a = float(a)
    b = float(b)
    c = float(c)
    D = b**2 - 4*a*c
    if D >= 1:
        x1 = (-b + akar(D))/(2*a)
        x2 = (-b - akar(D))/(2*a)
        hasil = (x1,x2)
        print hasil
    else:
        print "Determinannya negatif. Persamaan tidak mempunyai angka real"

===== RESTART: C:/Python27/Modul1.10.py =====
>>> selesaikanABC(1,-5,6)
(3.0, 2.0)
>>> selesaikanABC(1,2,3)
Determinannya negatif. Persamaan tidak mempunyai angka real

```