Lecture 2, Computer Algebra 2014

```
a=[1,2,3]
b=a
b.append(4)
print a,b
```

```
[1, 2, 3, 4] [1, 2, 3, 4]
```

```
c=a[:]
c.append(5)
print a,c
```

```
[1, 2, 3, 4] [1, 2, 3, 4, 5]
```

```
for i in range(0,2):
    print i
a=[1,2]

b=[x*2 for x in a]
print b
#the previous option is easier than:
c=[]
for x in a:
    c.append(2*x)
print c
```

```
0
1
[2, 4]
[2, 4]
```

```
R.<X>=PolynomialRing(ZZ)
f=1+2*X+5*X^2
g=5+3*X+X^2+2*X^3
h=f*g
print h
```

```
10*X^5 + 9*X^4 + 19*X^3 + 32*X^2 + 13*X + 5
```

```
#Faulty implementation: performs (n+m+1)(n+1) sums instead of nm
def M(a,b):
   n=len(a)-1
   m=len(b)-1
   d=[]
   tmp=[]
   c=[0]*(n+m+1)
   print c
```

```
   for i in range(0,n+1):
       tmp=[k * a[i] for k in b]
       for k in range(0,i):
           tmp.insert(0,0) #shift i positions
       for k in range(0,n-i):
           tmp.append(0) #to get arrays of length m+n+1 (faulty
implementation)
       d.append(tmp)
       print d
   for j in range(0,n+m+1):
       for i in range(0,n+1):
           c[j]=c[j] + d[i][j]
   return c

print M([1,2,5],[5,3,1,2])
```

```
[0, 0, 0, 0, 0, 0]
[[5, 3, 1, 2, 0, 0]]
[[5, 3, 1, 2, 0, 0], [0, 10, 6, 2, 4, 0]]
[[5, 3, 1, 2, 0, 0], [0, 10, 6, 2, 4, 0], [0, 0, 25, 15, 5, 10]
[5, 13, 32, 19, 9, 10]
```

```
S.<T>=PolynomialRing(GF(3))
ff=1+2*T+5*T^2
gg=5+3*T+T^2+2*T^3
hh=ff*gg
print hh
```

```
T^5 + T^3 + 2*T^2 + T + 2
```

```
ff.gcd(gg)
```

```
1
```

```

```

```

```