Homework 1
H. Altuğ Yıldırım
090100252

1) Firstly, I took the course mostly not for the analysis methods but for learning Python itself. I think that the libraries that Python have are very powerful tools that can replace the closed source tools like MatLab etc. Also Python is a sufficient language for writing simple yet efficient programs and even games.

2) Using Fortran Matrix Multiplication:

Program:

```
program xmatmul
  implicit none

  integer i,j
  integer, parameter :: M=2,N=3

  real :: a(M,N) = reshape((/1.0,4.0,2.0,5.0,3.0,6.0/),(/M,N/))
  real :: b(N,M) = reshape((/1.0,2.0,3.0,4.0,5.0,6.0/),(/N,M/))
  real :: c(M,M)
  integer :: clock0, clock1, clockmax, clockrate, ticks
  real    :: secs

  call system_clock(count_max=clockmax, count_rate=clockrate)
  call system_clock(clock0)

  write(*,*) 'Matrix [a]'
  do i=1,M
    write(*,1000) (a(i,j),j=1,N)
  enddo
  write(*,*)

  write(*,*) 'Matrix [b]'
  do i=1,N
    write(*,1000) (b(i,j),j=1,M)
  enddo
  write(*,*)

  c = matmul(a, b)
  write(*,*) 'Matrix [c] = [a] x [b]'
  do i = 1,M
    write(*,1000) (c(i,j),j=1,M)
  enddo
  write(*,*)
  call system_clock(clock1)
  ticks = clock1-clock0
  ticks = mod(ticks+clockmax, clockmax)
  secs = float(ticks)/float(clockrate)
```

```
      write(*,*) 'Code took ', secs, ' seconds'
1000  FORMAT(1x,1P10E14.6)
     end program xmatmul
!###########################################################################
!Sources:
!http://www.nr.com/forum/showthread.php?t=1434
!http://www.hpcx.ac.uk/support/documentation/UserGuide/HPCxuser/Porting_Codes.html
!###########################################################################
```

```
Output:
 Matrix [a]
   1.000000E+00  2.000000E+00  3.000000E+00
   4.000000E+00  5.000000E+00  6.000000E+00

 Matrix [b]
   1.000000E+00  4.000000E+00
   2.000000E+00  5.000000E+00
   3.000000E+00  6.000000E+00

 Matrix [c] = [a] x [b]
   1.400000E+01  3.200000E+01
   3.200000E+01  7.700000E+01

 Code took   -1.00000005E-03  seconds
```

Same program with using Python:

```
from numpy import *
import time
start = time.clock()
a=array([[1.,2.,3.],[4.,5.,6.]])
b=a.T
c=dot(a,b)
print 'Matrix [a]'
print a
print 'Matrix [b]'
print b
print 'Matrix [c] = [a] x [b]'
print c
print 'Code took ', time.clock() - start, ' seconds'
######################
######Sources########
#https://stackoverflow.com/questions/14452145/ -->
#--> how-to-measure-time-taken-between-lines-of -->
#--> -code-in-python
```

```
Output:
Matrix [a]
[[ 1.  2.  3.]
 [ 4.  5.  6.]]
Matrix [b]
[[ 1.  4.]
 [ 2.  5.]
 [ 3.  6.]]
Matrix [c] = [a] x [b]
[[ 14.  32.]
 [ 32.  77.]]
Code took  0.001763  seconds
```

I can not able to understand the negative sign of the code time in FORTRAN program but, if we neglect the negative sign Python is slower than the FORTRAN(0.001763 > 0.001). But as can be seen in the code, it is much more shorter than the FORTRAN equivalent of it.