INTRODUCTION TO PROGRAMMING FOR SCIENTISTS

Lecture 3:
Files and Standard Libraries

Prof. Steven Ludtke
N421, sludtke@bcm.edu

HOMEWORK REVIEW

- Counting Letters

```
a=raw_input("Enter Text: ")
b=a.lower()
length=len(b)
letters = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q",
"r", "s", "t", "u", "v", "w", "x", "y", "z"]
for i in xrange(0,26):
        if (b.count(letters[i], 0, length) > 0):
              print letters[i], ":", b.count(letters[i], 0, length)
```

HOMEWORK REVIEW

```
Flag = 'y'
while(Flag == 'y'):
    Astring = raw_input("Please enter a string: ")
    D = {}

    for word in Astring:
        word = word.lower()
        if word.isalpha():
            if not D.has_key(word): D[word] = 1
            else: D[word] = D[word]+ 1

    for key in D.keys():
        print key, ': ', D[key]

    Flag = raw_input("Continue? (y/n): ")
```

HOMEWORK REVIEW

```
a=raw_input("Enter letters:")
b=sorted(a)
c="".join(b)
x=[(i, c.count(i)) for i in c]
y=set(x)
z=list(y)
m=dict(z)
print m
```

HOMEWORK REVIEW

```
a=raw_input("Words?")
b=a.replace(' ','')
c=b.upper()
d=list(c);d.sort()
e=''.join(d)

f=list(set(e));f.sort()
g=''.join(f)

for x in g:
   k=e.count(x)
   if k >= 1:
      print x, k
```

HOMEWORK REVIEW

```
sentence =raw_input()
letters = sentence.replace(' ', '')
loletters=letters.lower()
lletters= list(loletters)
lletters.sort()

d = {}
for i in lletters:
   d[i]= lletters.count(i)

for i in d:
        print i,":",d[i]
```

HOMEWORK REVIEW

```
import string
s=raw_input("Enter string: ")
s=s.lower()

for l in string.ascii_lowercase:
        n=s.count(l)
        if n>0 : print l,n
```

HOMEWORK REVIEW

```
s=raw_input(“Enter string: “)
s=s.lower()
let={}

for i in s:
   if i.isalpha() : let[i]=let.setdefault(i,0)+1

for i in sorted(let.keys()):
   print i,let[i]
```

HOMEWORK REVIEW

```
#include <stdio.h>
#include <ctype.h>

main() {
    int let[26];
    for (int i=0; i<26; i++) let[i]=0;
    for (int c=getc(stdin); c!=EOF; c=getc(stdin)) {
        c=toupper(c);
        if (c>64 && c<91) let[c-65]++;
    }
    for (int i=0; i<26; i++) {
        if (let[i]) printf("%d %c\n",let[i],i+65);
    }
}
```

PROGRAMMING

- How do we represent the data ?
- Break the task into small pieces
- Code each of the pieces

FIND THE FIRST N PRIMES

- Check to see if each number is divisible by every other number

```
for i in range(10000):
    for j in range(2,i/2):
        if i%j==0 : break
    else: print i
```

FIND THE FIRST N PRIMES

- What if we skip even numbers, which are divisible by 2, as well as even factors, which cannot be prime

```
for i in range(3,10000,2):
      for j in range(3,i/2,2):
            if i%j==0 : break
      else: print i
```

FIND THE FIRST N PRIMES

- Actually, we just need to find the FIRST prime factor to show that it isn't prime. The first prime factor, if it exists, is guaranteed to be <sqrt (number). Think about it...

```
for i in range(3,10000,2):
    for j in range(3,int(i**.5)+1,2):
        if i%j==0 : break
    else: print i
```

FIND THE FIRST N PRIMES

- The only factors we need to check are themselves prime numbers.

```
primes=[3]
for i in range(5,10000,2):
    for j in primes:
        if i%j==0: break
    else: primes.append(i)

for i in primes: print i
```

FIND THE FIRST N PRIMES

```
primes=[3]
m=1
for i in range(5,10000,2):
   while primes[m-1]**2<i : m+=1
   for j in primes[:m]:
      if i%j==0: break
   else: primes.append(i)
```

FIND THE FIRST N PRIMES

- A 1-line program which does the same thing. Not the fastest, and certainly not easy to follow.

```
[i for i in range(3,10000,2) if len([j for j in
              range(3,int(i**.5)+1,2) if i%j==0])
==0]
```

OBFUSCATED C CONTEST

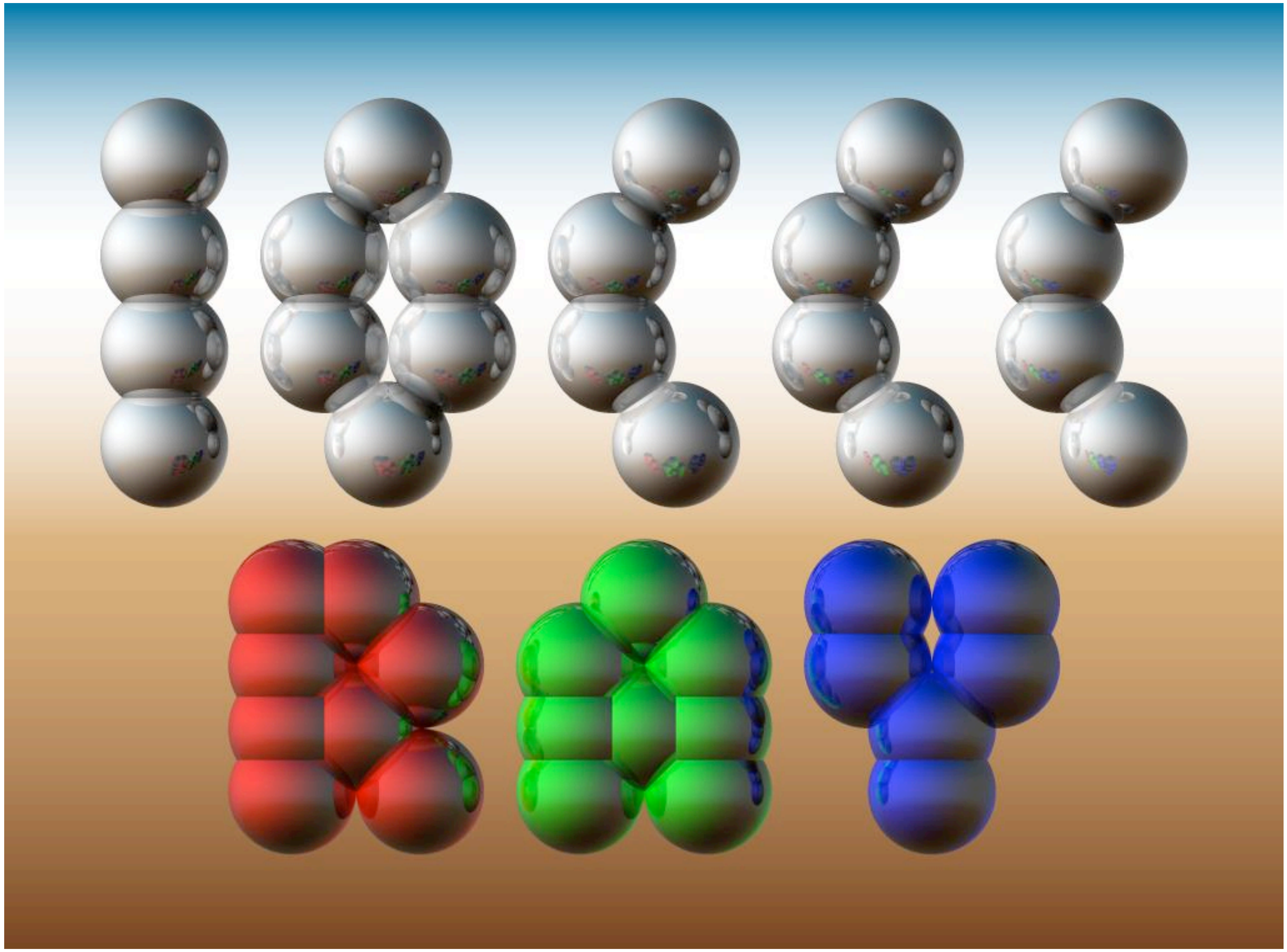
Goal of the competition is to produce a C program which does something interesting but is as difficult to read as possible, and may be internally complicated. This is much like the poetry of programming. Here is an example of a recent winner:

OBFUSCATED C CONTEST

```c
                               X=1024; Y=768; A=3;

J=0;K=-10;L=-7;M=1296;N=36;O=255;P=9;_=1<<15;E;S;C;D;F(b){E="1""111886:6:??AAF"
"FHHMMOO55557799@@>>>BBBGGIIKK"[b]-64;C="C@=::C@@==@=:C@=:C@=:C5""31/513/5131/"
"31/531/53"[b ]-64;S=b<22?9:0;D=2;}I(x,Y,X){Y?(X^=Y,X*X>x?(X^=Y):0,  I (x,Y/2,X
)):(E=X);       }H(x){I(x,   _,0);}p;q(         c,x,y,z,k,l,m,a,          b){F(c
);x-=E*M     ;y-=S*M           ;z-=C*M         ;b=x*        x/M+         y*y/M+z
*z/M-D*D    *M;a=-x               *k/M    -y*l/M-z        *m/M;    p=((b=a*a/M-
b)>=0?(I    (b*M,_      ,0),b     =E,      a+(a>b      ?-b:b)):     -1.0);}Z;W;o
(c,x,y,     z,k,l,    m,a){Z=!     c?      -1:Z;c     <44?(q(c,x         ,y,z,k,
l,m,0,0     ),(p>      0&&c!=      a&&        (p<W         ||Z<0)          )?(W=
p,Z=c):     0,o(c+         1,     x,y,z,        k,l,          m,a)):0     ;}Q;T;
U;u;v;w    ;n(e,f,g,             h,i,j,d,a,    b,V){o(0      ,e,f,g,h,i,j,a);d>0
&&Z>=0? (e+=h*W/M,f+=i*W/M,g+=j*W/M,F(Z),u=e-E*M,v=f-S*M,w=g-C*M,b=(-2*u-2*v+w)
/3,H(u*u+v*v+w*w),b/=D,b*=b,b*=200,b/=(M*M),V=Z,E!=0?(u=-u*M/E,v=-v*M/E,w=-w*M/
E):0,E=(h*u+i*v+j*w)/M,h-=u*E/(M/2),i-=v*E/(M/2),j-=w*E/(M/2),n(e,f,g,h,i,j,d-1
,Z,0,0),Q/=2,T/=2,       U/=2,V=V<22?7:  (V<30?1:(V<38?2:(V<44?4:(V==44?6:3))))
,Q+=V&1?b:0,T                +=V&2?b         :0,U+=V    &4?b:0)     :(d==P?(g+=2
,j=g>0?g/8:g/     20):0,j    >0?(U=      j    *j/M,Q      =255-    250*U/M,T=255
-150*U/M,U=255    -100    *U/M):(U     =j*j     /M,U<M           /5?(Q=255-210*U
/M,T=255-435*U            /M,U=255    -720*      U/M):(U       -=M/5,Q=213-110*U
/M,T=168-113*U    /       M,U=111                -85*U/M)      ),d!=P?(Q/=2,T/=2
,U/=2):0);Q=Q<    0?0:      Q>O?      O:          Q;T=T<0?    0:T>O?O:T;U=U<0?0:
U>O?O:U;}R;G;B    ;t(x,y     ,a,     b){n(M*J+M    *40*(A*x   +a)/X/A-M*20,M*K,M
*L-M*30*(A*y+b)/Y/A+M*15,0,M,0,P,  -1,0,0);R+=Q    ;G+=T;B   +=U;++a<A?t(x,y,a,
b):(++b<A?t(x,y,0,b):0);}r(x,y){R=G=B=0;t(x,y,0,0);x<X?(printf("%c%c%c",R/A/A,G
/A/A,B/A/A),r(x+1,y)):0;}s(y){r(0,--y?s(y),y:y);}main(){printf("P6\n%i %i\n255"
                                "\n",X,Y);s(Y);}
```

Obfuscated C Contest



ARGUMENTS

- Can’t do this if you start by clicking on the program
- *myprogram.py file1.txt file2.txt*
- *python myprogram.py file1.txt file2.txt*

- from sys import argv
- len(argv) -> 3
- argv[0] -> myprogram.py
- argv[1] -> file1.txt

STRING FORMATTING

- Uses ‘%’ in a string context
- “%d %f %s”%(1,2.0,”test”) -> '1 2.000000 test'
- Operators:
  - %*n*d -> integer with *n* characters
  - %*n.m*f -> floating point with *n* characters and *m* sig. figs.
  - %*n*s -> string with at least *n* characters
- Other options: ‘-’ to left justify, ‘0’ to zero-pad

FILES

- infile=file(“test.txt”,”r”)
- line=file.readline()
- lines=file.readlines()
- data=file.read(bytes)

- outfile=file(“test.txt”,”w”)
- outfile.write(“a test string\n”)
- outfile.close()

USEFUL STANDARD LIBS

- pprint
- os
- datetime
- time
- Queue
- random
- hashlib
- urllib

Homework #3

- BEGIN THINKING ABOUT WHAT TO DO FOR YOUR CLASS PROJECT

- Write a simplified amortization program, that is, a program that keeps track of how much you still owe on a loan. We will simplify the problem a bit. Assume that each month you are charged 1/12 of the annual percentage rate on the remaining balance of the loan. The amount of the monthly payment will be constant.

  You should prompt the user for the amount of the loan, the annual interest rate, and the payment amount. WRITE THE OUTPUT TO A FILE ‘result.txt’, NOT TO THE SCREEN (no ‘print’). For each month, write out the payment number, interest for the month, and the remaining balance on the loan after the payment. Continue to write out new months until the loan is payed off.

  Send by email the program (.py file) and the output from a run of the program with a $150,000 loan at 5% interest and a $1500 monthly payment

  Note: you are welcome to talk to other people about any of the homework problems, myself included, but the actual code should be your own, not copied.