Crash Course: September/October 2015 Solutions

As ever more efficient solutions could easily be possible!

1. Write a program that prints the coordinates of those points (a, b) with $1 < a < 10$ and $1 < b < 10$, a and b integers, which are inside the circle centred at (4,5) with radius 3.

```
for a in range(1,11):
    for b in range(1,11):
        if ((a-4)*(a-4)+(b-5)*(b-5))<9:
            print a,b
```

2. Assuming a, b, c and d are real numbers with $a \neq c$, write a program that takes inputs of a, b, c, d and t and prints to the screen the y-coordinate of the point on the straight line joining (a,b) and (c, d) with x-coordinate of t.

```
def line(a,b,c,d,t):
    gradient = (d-b)/(c-a)
    ycoord=b+gradient*(t-a)
    print ycoord
```

3. You can invest £1000 for one year in any of the following banks:

   Bank 1 pays you 100% interest once, at the end of the year.

   Bank 2 pays you 50% interest after six months, and then a further 50% interest after the second sixth months.

   Bank 3 pays you 33.3333…% interest after 4 months, then 33.3333…% after another 4 months and the final 33.3333…% after the last four months.

   and so on…

   Write a function which takes input of a positive integer n and returns the amount in your account at Bank n after one year should you invest your £1000 pounds there. What is the smallest value of n such that bank n returns more than £2700 after one year?

```
def amount(n):
    return 1000*((1.0+1/float(n))**n)

back = amount(1)
count = 1

while back<2700:
    count=count+1
    back=amount(count)

print count+1
```

4. $\sum_{r=1}^{\infty} \frac{1}{3^r} = 0.5$. How many terms, n, are needed in $\sum_{r=1}^{n} \frac{1}{3^r}$ so that its value differs from 0.5 by less than a) 0.1, b) 0.01, c) 0.001

```
def terms(close):
    sum = 0.0
    count = 1
    while sum<(0.5-close):
        sum = sum + (1.0/3.0)**count
        count = count + 1
    return count

print terms(0.1)
print terms(0.01)
print terms(0,001)
```

(

5. This question is related to definition of differentiation ('first principles').

   Here you'll find an approximation to the derivative of f(x) = x^2 at x = 3.

   Starting with h = 1, calculate $\frac{(3+h)^2 - 3^2}{h}$. Then continually halve h and recalculate. Stop doing this when the distance between two successive results is less than 0.00001 and print the value of the latest calculation to the screen.

```
def chord(h):
    return ((3.0+h)**2-9.0)/h

next = 1
prev = 0
h = 1.0
while abs(next-prev)>=0.00001:
    prev = next
    next = chord(h)
    h=h/2

print next
```

6. Write a function that takes a, b, n (a positive integer) as inputs, prints the estimate to $\int_a^b x^2 dx$ using the trapezium rule with n trapezia and then prints the difference between this value and $\frac{b^3}{3} - \frac{a^3}{3}$. What is the smallest value of n such that the difference between the estimate to $\int_1^2 x^2 dx$ using n trapezia and $\frac{2^3}{3} - \frac{1^3}{3}$ is less than 0.00001?

```
def estimate(a,b,n):
    sum=0.0
    gap = (b-a)/float(n)
    for i in range(1,n):
        sum = sum + 2*(a+i*gap)**2
    sum = sum + a**2 + b**2
    sum = (sum*gap)/2
    return sum


n=1
while abs(7.0/3.0-estimate(1.0,2.0,n))>=0.00001:
    n = n + 1

print n
```