This is a smaller (35-point) exam for a 4.5-credit course. This is why it has 2 questions only.

Q1

As a short reminder, the Newton-Raphson method can be used to calculate solutions to an equation $f(x) = 0$ as long as the function has a derivative (and some other assumptions).

- You start with an initial guess x_0
- Then you calculate iterative values using this formula: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
- You continue calculating new values until the difference is small enough

For our specific example, the iterative formula for calculating the cube root is: $x_{n+1} = \frac{1}{3}(2x_n + r/x_n^2)$

(If you want you can convince yourself about this by derivating the function $f(x) = x^3 - r$, but you don't have to.)

Your task is to write a function that calculates the cube root using the Newton-Raphson method. It should take an additional argument which gives the desired accuracy. At each iteration it should print the current iteration number, and the current approximation. The function should stop when the difference between two approximations (x_n and x_{n+1}) is smaller than the given accuracy. Use 1.0 for the initial guess.

```
>>> cube_root(123456, 0.001)
Iteration 0 : 1.0
Iteration 1 : 41152.666666666664
Iteration 2 : 27435.11113541048
Iteration 3 : 18290.07414494723
Iteration 4 : 12193.382886313704
Iteration 5 : 8128.9222009941195
Iteration 6 : 5419.282090095583
Iteration 7 : 3612.8561279539495
Iteration 8 : 2408.573904719865
Iteration 9 : 1605.7230301501113
Iteration 10 : 1070.4979807170064
Iteration 11 : 713.7012307952664
Iteration 12 : 475.8816106133979
Iteration 13 : 317.43612304754566
Iteration 14 : 212.03247499854152
Iteration 15 : 142.27033121515026
Iteration 16 : 96.88000396181768
Iteration 17 : 68.97119532155453
Iteration 18 : 54.63158034697543
Iteration 19 : 50.209121658040324
Iteration 20 : 49.79671442213922
Iteration 21 : 49.793280083624325
```

Q2

Implement a function that prints a histogram using asterisks (*). The function should have the following signature:

```
histogram(max_bar, labels, counts)
```

Where the parameters are the following:

- `max_bar` : the number of "*" symbols for the value with the highest count
- `labels` : a list of the labels that the histogram should be printed for
- `counts` : a dictionary mapping labels to their counts

This function can e.g. be used to print the word counts from lab 1. Here's an example run:

```
counts = {"pig": 19, "cat":57, "wolf":42, "horse":26}
labels = ["horse", "pig", "cat"]
histogram(20,labels,counts)
horse     *********
pig       ******
cat       ********************
>>>
```

Requirements:

- the histogram should be printed in the same order as the labels in the list
- the largest count (57 in the example) should be printed with `max_bar` asterisks (20 in the example)
- the other counts should be printed with a proportionate number of asterisks
- the histogram bars should all start in the same column, column 10
- you can assume that all labels have at most 10 characters