Student name: Abao Zhang (张宝)

Student number: 12332459

Homework 2

Question 1

固定点迭代法无法用来寻找$x = g(x)$的解。

因为对于给定 $g(x) = x^2 + x - 4$ 在其作用域$(-\infty, \infty)$上，其导数$g'(x) = 2x + 1$的取值范围为$(-\infty, \infty)$，不符合 Lipschitz 连续性。

Question 2

欲证明 $|p_2 - p_1| < K|p_1 - p_0|$，将$p_2 = g(p_1)$和$p_1 = g(p_0)$代入不等式右边，则变为证明$|g(p_1) - g(p_0)| < K|p_1 - p_0|$，即证明$|\frac{g(p_1)-g(p_0)}{p_1-p_0}| < K$.

根据拉格朗日中值定理，存在p在p_0和p_1之间，使得$g'(p) = \frac{g(p_1)-g(p_0)}{p_1-p_0}$，而$|g'(p)| < K$成立，故$|\frac{g(p_1)-g(p_0)}{p_1-p_0}| < K$成立，命题得证。

Question 3

令$f(x) = e^x - 2 - x$，$f(a_0) = f(-2.4) \approx 0.49$，$f(b_0) = f(-1.6) \approx -0.20$，根据 False Position 公式

$$c = a - \frac{f(a) \cdot (b - a)}{f(b) - f(a)}$$

有

$$c_0 = a_0 - \frac{f(a_0) \cdot (b_0 - a_0)}{f(b_0) - f(a_0)} \approx -1.83$$

由于$f(c_0) \approx -0.01$，故$a_1 = a_0 = -2.4$，$b_1 = c_0$

同理$c_1 \approx -1.840925216033374$，$f(c_1) \approx -0.0004042303338573916$，$a_2 = a_1$，$b_2 = c_1$

$c_2 \approx -1.841385376445813$，$f(c_2) \approx -1.706703234161111e - 05$，$a_3 = a_1$，$b_3 = c_2$

$c_3 \approx -1.8414048042295494$

Question 4

(a)

当区间为$[3, 4]$时，$f(x) = \tan(x)$在$[3, 4]$上连续，且$f(3) \approx -0.1425465430742778$，$f(4) \approx 1.1578212823495775$，两者异号，当使用二分法进行求解时，能够通过不断迭代接近于解析解$x = \pi$。

(b)

当区间为$[1, 3]$时，$f(x) = \tan(x)$在$[1, 3]$上不连续，不符合使用二分法的条件，故不能使用二分法进行计算。

Question 5

对$\frac{|b-a|}{2^{n+1}} < \delta$两边取自然对数，有

$$\ln(|b-a|) - (n+1)\ln(2) < \ln(\delta)$$

整理可得

$$\frac{\ln(|b-a|) - \ln(\delta)}{\ln(2)} < n+1$$

由于迭代次数n为整数，我们需要取不等式左侧的整数上取整，因此迭代次数 N 应为

$$N = \left\lfloor \frac{\ln(b-a) - \ln(\delta)}{\ln(2)} \right\rfloor$$

Question 6

详情请见 `question-6.py`，使用 python 实现了相同的 `bisect` 和 `regula` 方法，调用两个方法并进行结果打印:

```
print(bisect(f, -2, 2, 0.00001))
print(regula(f, -2, 2, 0.00001, 0.00001, 100000))
```

输出如下:

```
python question-6.py

(-0.7034645080566406, 7.62939453125e-06, -5.542682600612192e-06)
(-0.7034654049217414, 0.6482672975391293, -3.837026751274397e-06)
```

Question 7

(a)

已知$f(x) = \cos(x)$, 则$f'(x) = -\sin(x)$，故牛顿迭代法公式

$$p_k = g(p_{k-1}) = p_{k-1} - \frac{f(p_{k-1})}{f'(p_{k-1})} = p_{k-1} + \frac{\cos(p_{k-1})}{\sin(p_{k-1})}$$

(b)

要找$p = \frac{3\pi}{2}$的根，不能使用用$p_0 = 3$的初始值，在第一次迭代时，由于$\sin(3)$比较小，而$\cos(3)$比较大，所以会作出比较大的跳跃，最终收敛于$-\frac{3\pi}{2}$

(c)

要找$p = \frac{3\pi}{2}$的根，可以使用用$p_0 = 5$的初始值，不会发生较大的跳跃，最终收敛于$\frac{3\pi}{2}$

Question 8

(a)

- 当 N 为奇数时 $\sqrt[N]{A}$为方程$f(x) = 0$的解
- 当 N 为偶数时：
  - 如果 A 为正数，$\sqrt[N]{A}$为方程$f(x) = 0$的解
  - 如果 A 为负数，方程$f(x) = 0$的在实数范围内没有解

(b)

已知$f(x)=x^N-A$，则有$f'(x)=Nx^{N-1}$

使用牛顿迭代法，

$$p_k = p_{k-1} - \frac{f(p_{k-1})}{f'(p_{k-1})} = p_{k-1} - \frac{p_{k-1}^N - A}{Np_{k-1}^{N-1}}$$

整理可得

$$p_k = \frac{(N-1)p_{k-1} + A/p_{k-1}^{N-1}}{N}$$

Question 9

(a)

当炮弹落地时，$y=0$，我们使用二分法对$y=f(t)=0$进行求解即可得到炮弹落地所需要的时间，调用 Question 6 中所实现的算法，代码如下：

```
def f(t):
    return 9600 * (1 - exp(-t / 15)) - 480 * t

print(bisect(f, 0, sys.float_info.max, 0.00000000001))
```

输出如下：

```
(9.08789966878612, 7.275957614183426e-12, -1.4642864698544145e-10)
```

其中第一个数值为时间t，第二个数值为误差，第三个值为该时刻对应的高度

(b)

将(a)中计算得到的t带入方程$x=r(t)$，计算可得x，具体代码如下

```
def r(t):
    return 2400 * (1 - exp(-t / 15))

print(r(bisect(f, 0, sys.float_info.max, 0.00000000001)[0]))
```

结果如下：

```
1090.5479602542978
```