S173102 朴 東勺

No. . . ( )

§14.

(1).
```
x = int(input("整数を入力してください:"))
y = 2*x - 1

for i in range(0, x):
    print(y - 2*i)
```

(2) (a) STEP1: $a = \boxed{1}$, $b = \boxed{2}$, $c = (a+b)/2$ とする。STEP2へ進む。

STEP2: $c^2+c-5 \neq 0$ かつ $b-a > 0.000002$ ならば STEP3へ進み

そうでなければ STEP4へ進む。

STEP3: $c^2+c+5>0$ ならば $\boxed{b = c\ (b\text{にcの値を代入})}$ とし

$c^2+c+5<0$ ならば $\boxed{a = c\ (a\text{にcの値を代入})}$ とする。

そして $c = (a+b)/2$ として STEP2へ進む。

STEP4: cと c^2+c-5 の値を表示し、終了する。

(b)
```
def f(x):
        return x*x + x - 5
    i=1
    a=1
    b=2
    c=(a+b)/2
    while 2*e <= b-a:
        if f(c) > 0:
            b = c
        else:
            a = c
        c = (a+b)/2
        i = i+1
    print(c, f(c))
```

(c) 二分法より a と b の間の距離($|b-a|$)が 0.00001(0) に近接(収束)する。
よって STEP2 と STEP3 を繰り返すことで $b-a \leq 0.000002$ をみたすことになる。
したがって、STEP4へ進め、アルゴリズムは必ず終了する。

(d) (c)の過程を通じて、a と b の間の距離($|b-a|$)が 0.00001(0) に近接(収束)する。
これは $|\beta - c|$ が 0.00001(0) に近接(収束)する過程と同じ形である。
よって、アルゴリズムを通じて $|\beta - c| \leq 0.000001$ がみたされるようになる。

No. 　・　・　・（　　）

(3) (a) <マクローリンの定理>

関数$f(x)$: 0を含むある開区間 $I=(a,b)$でC^{n+1}級関数

このとき、$^\forall x \in (a,b)$に対して、

$$^\exists \theta \in (0,1) \ \text{s.t.}\ f(x)=f(0)+f'(0)\cdot x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n)}(0)}{n!}x^n+\frac{f^{(n+1)}(\theta x)}{(n+1)!}x^{n+1}$$

(b) マクローリンの定理より

$$e^x=1+x+\frac{1}{2!}x^2+\cdots+\frac{1}{n!}x^n+\frac{e^{\theta x}}{(n+1)!}\cdot x^{n+1}$$

このとき、$x=-2$とすると

$$e^{-2}=1-2+\frac{2^2}{2!}-\frac{2^3}{3!}+\cdots+\frac{(-2)^n}{n!}+\frac{e^{-2\theta}}{(n+1)!}\cdot(-2)^{n+1}$$

ここで

$$\left|e^{-2}-\left\{1-2+\frac{2^2}{2!}-\frac{2^3}{3!}+\cdots+\frac{(-2)^n}{n!}\right\}\right|=\left|\frac{(-2)^{n+1}}{(n+1)!\,e^{2\theta}}\right|\leq\frac{2^{n+1}}{(n+1)!}\leq 2.$$

であるから、$2<\varepsilon$に対して、$^\forall n\in\mathbb{N}$:

$$\left|e^{-2}-\left\{1-2+\frac{2^2}{2!}-\frac{2^3}{3!}+\cdots+\frac{(-2)^n}{n!}\right\}\right|<\varepsilon.$$

よって $\left|e^{-2}-\left\{1-2+\frac{2^2}{2!}-\frac{2^3}{3!}+\cdots+\frac{(-2)^n}{n!}\right\}\right|$ は0に収束する

このとき、$n=13$と定めると

$$\left|e^{-2}-\left\{1-2+\frac{2^2}{2!}-\frac{2^3}{3!}+\cdots-\frac{2^{13}}{13!}\right\}\right|<1.65679\times10^{-7}<0.000001.$$

(c)
```
n=0
 p=1
 q=1
 sum=1
 while(abs(2*p/(n+1))>0.000001):
     n=n+1
     p=2*p/n
     q=-q
     sum=sum+p*q
 print(n,sum)
```

No. 　　　　(　　　)

(4) (a) $y - f(x_k) = f'(x_k)(x - x_k)$

このとき，$(x, y) = (x_{k+1}, 0)$ を代入すると，

$0 - f(x_k) = f'(x_k)(x_{k+1} - x_k)$

$\Rightarrow x_k f'(x_k) - f(x_k) = f'(x_k) \cdot x_{k+1}$

$\therefore x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)} \quad (\because f'(x_k) \neq 0)$

(b). (a)より $x_k > x_{k+1}$.

よって．$x_1 > x_2 > x_3 > \cdots > x_k > x_{k+1} > \cdots$

したがって得られた数列 $\{x_k\}$ が $f(x) = 0$ の解に収束することが期待される．

(5) a = [0]*3 より配列 [0.0.0] が生成される．

[0.0.0]出力 → a[2] = a[2]+1 $_{0}$ → [0.0.1]出力 → a[2] = a[2]+1 $_{1}$ → [0.0.2]出力
→ a[2] = a[2]+1 より [0.0.3] → while文：{[0.0.0] → k = 1 → a[1] = a[1]+1 → [0.1.0]} $_{2}$
→ [0.1.0]出力 → a[2] = a[2]+1 $_{3}$ → [0.1.1]出力 → a[2] = a[2]+1 $_{4}$ → [0.1.2]出力
→ a[2] = a[2]+1 より [0.1.3] → while文：{[0.1.0] → k = 1 → a[1] = a[1]+1 → [0.2.0]} $_{5}$
→ [0.2.0]出力 → a[2] = a[2]+1 $_{6}$ → [0.2.1]出力 → a[2] = a[2]+1 $_{7}$ → [0.2.2]出力
→ a[2] = a[2]+1 より [0.2.3] → while文：{[0.2.0] → k = 1 → a[1] = a[1]+1 → [0.3.0]
→ [0.0.0] → k = 0 → a[0] = a[0]+1 → [1.0.0]} $_{8}$ → [1.0.0]出力 → a[2] = a[2]+1 $_{9}$

[結果]
[0.0.0]
[0.0.1]
[0.0.2]
[0.1.0]
[0.1.1]
[0.1.2]
[0.2.0]
[0.2.1]
[0.2.2]
[1.0.0]