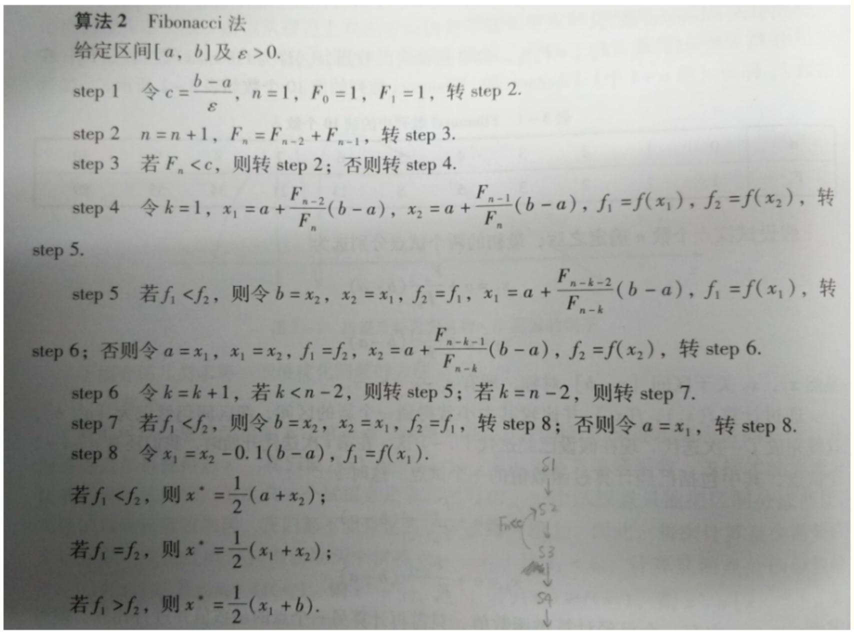
【task one】

Programming implementation algorithm "Fibonacci method "

Algorithm steps:



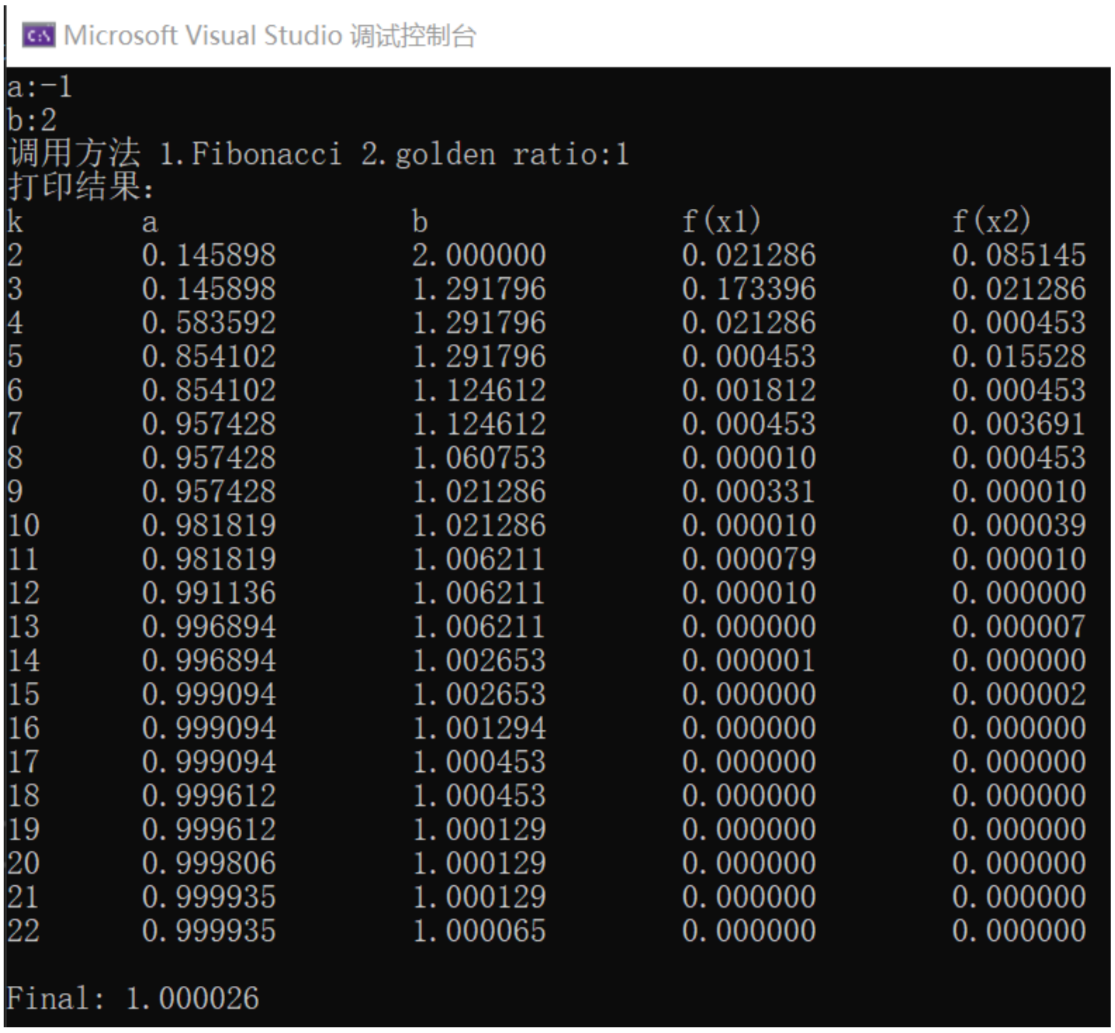
算法 2 Fibonacci 法

给定区间$[a, b]$及$\varepsilon > 0$.

```
step 1  令 $c = \frac{b-a}{\varepsilon}$, $n = 1$, $F_0 = 1$, $F_1 = 1$, 转 step 2.
step 2  $n = n + 1$, $F_n = F_{n-2} + F_{n-1}$, 转 step 3.
step 3  若 $F_n < c$, 则转 step 2; 否则转 step 4.
step 4  令 $k = 1$, $x_1 = a + \frac{F_{n-2}}{F_n}(b-a)$, $x_2 = a + \frac{F_{n-1}}{F_n}(b-a)$, $f_1 = f(x_1)$, $f_2 = f(x_2)$, 转 step 5.
step 5  若 $f_1 < f_2$, 则令 $b = x_2$, $x_2 = x_1$, $f_2 = f_1$, $x_1 = a + \frac{F_{n-k-2}}{F_{n-k}}(b-a)$, $f_1 = f(x_1)$, 转 step 6; 否则令 $a = x_1$, $x_1 = x_2$, $f_1 = f_2$, $x_2 = a + \frac{F_{n-k-1}}{F_{n-k}}(b-a)$, $f_2 = f(x_2)$, 转 step 6.
step 6  令 $k = k + 1$, 若 $k < n - 2$, 则转 step 5; 若 $k = n - 2$, 则转 step 7.
step 7  若 $f_1 < f_2$, 则令 $b = x_2$, $x_2 = x_1$, $f_2 = f_1$, 转 step 8; 否则令 $a = x_1$, 转 step 8.
step 8  令 $x_1 = x_2 - 0.1(b-a)$, $f_1 = f(x_1)$.
        若 $f_1 < f_2$, 则 $x^* = \frac{1}{2}(a + x_2)$;
        若 $f_1 = f_2$, 则 $x^* = \frac{1}{2}(x_1 + x_2)$;
        若 $f_1 > f_2$, 则 $x^* = \frac{1}{2}(x_1 + b)$.
```

$$\begin{tikzcd}
S_1 \arrow[d] \\
S_2 \arrow[d] \\
S_3 \arrow[u, bend left, "F_n<c"] \arrow[d] \\
S_4 \arrow[d] \\
S_5 \arrow[d] \\
S_6 \arrow[u, bend left, "k<n-2"] \arrow[d] \\
S_7 \arrow[d] \\
S_8
\end{tikzcd}$$

Screenshot of the experiment:



Microsoft Visual Studio 调试控制台

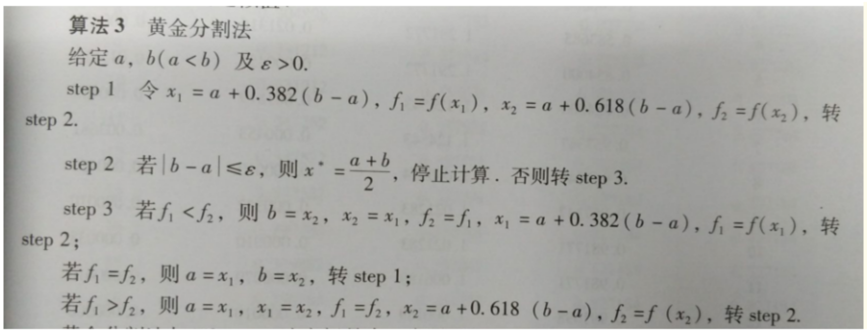
```
a:-1
b:2
调用方法 1.Fibonacci 2.golden ratio:1
打印结果:
k       a               b               f(x1)           f(x2)
2       0.145898        2.000000        0.021286        0.085145
3       0.145898        1.291796        0.173396        0.021286
4       0.583592        1.291796        0.021286        0.000453
5       0.854102        1.291796        0.000453        0.015528
6       0.854102        1.124612        0.001812        0.000453
7       0.957428        1.124612        0.000453        0.003691
8       0.957428        1.060753        0.000010        0.000453
9       0.957428        1.021286        0.000331        0.000010
10      0.981819        1.021286        0.000010        0.000039
11      0.981819        1.006211        0.000079        0.000010
12      0.991136        1.006211        0.000010        0.000000
13      0.996894        1.006211        0.000000        0.000007
14      0.996894        1.002653        0.000001        0.000000
15      0.999094        1.002653        0.000000        0.000002
16      0.999094        1.001294        0.000000        0.000000
17      0.999094        1.000453        0.000000        0.000000
18      0.999612        1.000453        0.000000        0.000000
19      0.999612        1.000129        0.000000        0.000000
20      0.999806        1.000129        0.000000        0.000000
21      0.999935        1.000129        0.000000        0.000000
22      0.999935        1.000065        0.000000        0.000000

Final: 1.000026
```

Task two:

Programming implementation algorithm " golden section method "

Algorithm steps:



算法 3 黄金分割法

给定 a, $b(a<b)$ 及 $\varepsilon>0$.

step 1 令 $x_1=a+0.382(b-a)$, $f_1=f(x_1)$, $x_2=a+0.618(b-a)$, $f_2=f(x_2)$, 转 step 2.

step 2 若 $|b-a|\leqslant\varepsilon$, 则 $x^*=\dfrac{a+b}{2}$, 停止计算. 否则转 step 3.

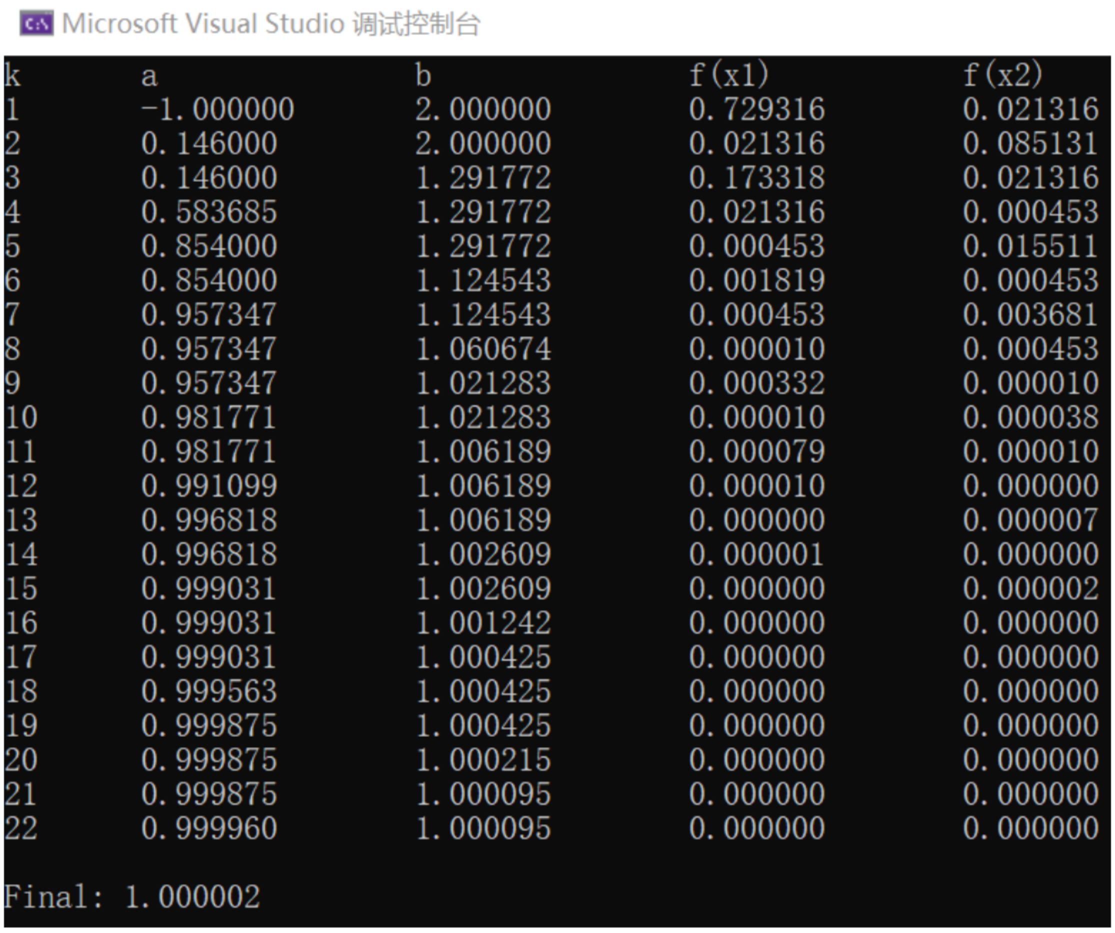
step 3 若 $f_1<f_2$, 则 $b=x_2$, $x_2=x_1$, $f_2=f_1$, $x_1=a+0.382(b-a)$, $f_1=f(x_1)$, 转 step 2;

若 $f_1=f_2$, 则 $a=x_1$, $b=x_2$, 转 step 1;

若 $f_1>f_2$, 则 $a=x_1$, $x_1=x_2$, $f_1=f_2$, $x_2=a+0.618\ (b-a)$, $f_2=f\ (x_2)$, 转 step 2.

$$\begin{tikzcd}
S_1 \arrow[d] \\
S_2 \arrow[d] \\
S_3 \arrow[u, bend right, "\begin{matrix} f_1<f_2 \\ f_1>f_2 \end{matrix}"'] \arrow[uu, bend left=60, "f_1=f_2"]
\end{tikzcd}$$

Screenshot of the experiment:


Microsoft Visual Studio 调试控制台

```
k       a               b               f(x1)           f(x2)
1       -1.000000       2.000000        0.729316        0.021316
2       0.146000        2.000000        0.021316        0.085131
3       0.146000        1.291772        0.173318        0.021316
4       0.583685        1.291772        0.021316        0.000453
5       0.854000        1.291772        0.000453        0.015511
6       0.854000        1.124543        0.001819        0.000453
7       0.957347        1.124543        0.000453        0.003681
8       0.957347        1.060674        0.000010        0.000453
9       0.957347        1.021283        0.000332        0.000010
10      0.981771        1.021283        0.000010        0.000038
11      0.981771        1.006189        0.000079        0.000010
12      0.991099        1.006189        0.000010        0.000000
13      0.996818        1.006189        0.000000        0.000007
14      0.996818        1.002609        0.000001        0.000000
15      0.999031        1.002609        0.000000        0.000002
16      0.999031        1.001242        0.000000        0.000000
17      0.999031        1.000425        0.000000        0.000000
18      0.999563        1.000425        0.000000        0.000000
19      0.999875        1.000425        0.000000        0.000000
20      0.999875        1.000215        0.000000        0.000000
21      0.999875        1.000095        0.000000        0.000000
22      0.999960        1.000095        0.000000        0.000000

Final: 1.000002
```

Experiment code:

```cpp
#include <iostream>
#include <math.h>
#include <iomanip>
using namespace std;

double fx( double x ){ // The desired function, take f(x)=x^2-2x+1 as an example
	return ( x * x - 2 * x + 1);
}
int printAll( int k , double a , double b , double f1 , double f2 ) { // print the result
	cout << k << "\t" << setiosflags( ios ::fixed) << a << "\t" << b << "\t"
		<< f1 << "\t"    << f2 << endl;
	return 0;
}
// Golden section method algorithm
int gold( double a , double b , double e , double ratio ) {
	double x1, x2, f1, f2, xx;
	int flag=1, k=1;
```

```
for ( int i = 0; ; i++) {
    //step1 :
    if (flag == 1) {
        x1 = a + (1- ratio ) * ( b - a );
        f1 = fx(x1);
        x2 = a + ratio * ( b - a );
        f2 = fx(x2);

        flag = 2;
    }
    //step2
    if (flag == 2) {
        if (abs( b - a ) <= e ) {
            xx = ( a + b ) / 2;
            break ;
        }
        else {
            flag = 3;
        }
    }

    printAll(k++, a , b , f1, f2);

    //step3
    if (flag == 3) {
        if (f1 < f2) {
            b = x2;
            x2 = x1;
            f2 = f1;
            x1 = a + 0.382 * ( b - a );
            f1 = fx(x1);

            flag = 2;
        }
        else if (f1 == f2) {
            a = x1;
            b = x2;

            flag = 1;
        }
        else if (f1 > f2) {
            a = x1;
            x1 = x2;
            f1 = f2;
```

```
                x2 = a + 0.618 * ( b - a );
                f2 = fx(x2);

                flag = 2;
            }
        }
    }
    cout << endl << "Final: " << xx << endl;
    return 0;
}
//Fibonacci method algorithm
int Fibonacci( double a , double b , double e ) {
    double c,x1,x2,xx,f1,f2;
    int F[40], n=1, flag=2, k;
    //Step1
    c = ( b - a ) / e ;
    n = 1;
    F[0] = F[1] = 1;
    do {
        //Step2
        n = n + 1;
        F[n] = F[n - 2] + F[n - 1];
    } while (F[n] < c); //Step3


    //Step4
    k = 1;
    x1 = a + ( double (F[n - 2]) / double (F[n])) * ( b - a );
    x2 = a + ( double (F[n - 1]) / double (F[n])) * ( b - a );
    f1 = fx(x1);
    f2 = fx(x2);
    //Step5 Step6
    do {
        if (f1 < f2) {
            b = x2;
            x2 = x1;
            f2 = f1;
            x1 = a + ( double (F[n - k - 2]) / double (F[n - k])) * ( b - a );
            f1 = fx(x1);
        }
        else {
            a = x1;
            x1 = x2;
            f1 = f2;
```

```
                x2 = a + ( double (F[n - k - 1]) / double (F[n - k])) * ( b - a );
                f2 = fx(x2);
            }
            k++;
            printAll(k, a , b , f1, f2);

        } while (k < n - 2);
        //Step7
        if (f1 < f2) {
            b = x2;
            x2 = x1;
            f2 = f1;
        }
        else {
            a = x1;
        }
        //Step8
        x1 = x2 - 0.1 * ( b - a );
        f1 = fx(x1);
        if (f1 < f2) {
            xx = 0.5 * ( a + x2);
        }
        else if (f1 == f2) {
            xx = 0.5 * (x1 + x2);
        }
        else if (f1 > f2) {
            xx = 0.5 * (x1 + b );
        }
        k++;
        printAll(k, a , b , f1, f2);
        cout << endl << "Final: " << xx << endl;
        return 0;
    }

    int main() {
        double a, b, e=0.0001, ratio=0.618; //e is Epsilon
        int choose = 0;
        cout << " Range [a,b]" << endl;
        cout << "a:" ;
        cin >> a;
        cout << "b:" ;
        cin >> b;
        cout << " Call method 1.Fibonacci 2.golden ratio:" ;
        cin >> choose;
```

```
	if (choose != 1 && choose != 2) {
		return 1;
	}
	cout << " Print result: " << endl << "k\ta\t\tb\t\tf(x1)\t\tf(x2)\t" << endl;
	if (choose == 1) {
		Fibonacci(a, b, e);
	}
	else if (choose == 2) {
		gold(a, b, e, ratio);
	}
}
```