

第4回課題

課題1(08-Newton.pdfの課題14)

08-Newton.pdfのスライド37ページを参考に、ニュートン法で2次方程式の解の1つを求めるプログラムを作成してください。

入力形式

求める2次方程式を $ax^2+bx+c=0$ とすると、入力は以下の形式で与えられます。

(aの値)
(bの値)
(cの値)
(xの初期値)
(計算精度)
(繰り返し回数)

入力は**小数**でも受け付けられるようにしてください。 **倍精度浮動小数点**の範囲("double"型の範囲)で受け付けられるようにしてください。

出力形式

計算精度を満たす解が求められた時

(解の値)

計算精度を満たす解が求められた時(重解の場合)

(解の値)(重解)

繰り返し上限になり、解が求められなかった時

繰り返し上限

出力桁数は**浮動小数点数型**の出力範囲("%f"で出力できる範囲)までで行ってください。

具体例

入力例1

```
1
2
1
10
0.000001
100
```

出力例1

-0.999999 (重解)

入力例2

```
2
-2
-12
5
0.000001
100
```

出力例2

3.000000

入力例3

```
1
-0.3
-0.1
100
0.000001
10
```

出力例3

繰り返し上限

注意事項

入力形式が**倍精度浮動小数点**になっていることに注意してください. ("**double型**"の範囲)

必ずプログラムのmain関数より前に"#include <math.h>"と記載し, mathライブラリをincludeしてください.

コピペ用プログラム

スライド37ページのプログラムをコピーできるようにしたものです。課題に合わせて、不要な出力を削っています。

```
#include <stdio.h>
#include <math.h>

/* f(x)の出力は不要です */

/*f(x)*/
double f(double x)
{
    return /* f(x) */;
}

/*f(x)の導関数*/
double g(double x)
{
    return /* f(x)の導関数 */;
}

int main()
{
    double x, new_x, eps;
    int number,i;

    /*データの取り込み(print文は課題の出力の関係で削除してます) */

    scanf(/* 初期値xの値の型は? */, &x);
    scanf(/* 計算精度の値の型は? */, &eps);
    scanf(/* 繰り返し回数の型は? */, &number);

    /* ニュートン法による解の導出 */
    for(/* forの繰り返し条件は? */){
        new_x = /* 適切な計算式は? */;

        if(/* 計算精度の判定の条件式は? */){
            printf("%f\n", new_x);
            break;
        }

        /* 収束条件、重解の判定をし、重解を出力 */
        if(/* 重解の時の計算精度の判定の条件式は? */){
            printf("%f(重解)\n", new_x);
            break;
        }
        x = new_x;
    }
    if(i == number) printf("繰り返し上限\n");
}
```

課題2(09-Integration.pdfの課題16)

09-Integration.pdfを参考に台形則による近似で関数の数値積分を行うプログラムを作成してください。

入力形式

(積分区間の値1)
(積分区間の値2)
(分割数)

入力は**小数**でも受け付けられるようにしてください。 倍精度浮動小数点の範囲("double"型の範囲)で受け付けられるようにしてください。

出力形式

(積分結果の値)

出力桁数は**浮動小数点数型の出力範囲("%f"で出力できる範囲)**までで行ってください。

具体例

入力例1

0
1
10000

出力例1

0.746824

注意事項

入力形式が**倍精度浮動小数点**になっていることに注意してください。 ("double型"の範囲)

必ずプログラムのmain関数より前に"#include <math.h>"と記載し、 mathライブラリをincludeしてください。

コピペ用プログラム

スライド5ページのプログラムをコピーできるようにしたものです。課題に合わせて、不要な出力を削っています。

```
#include<stdio.h>
#include<math.h>

/* f(x) */
double f(double x)
{
    return /* f(x)の関数 */;
}

int main()
{
    double a, b, x, h, sum = 0.0;
    int n, i;

    //積分区間
    scanf("%lf %lf", &a, &b);
    //分割数
    scanf("%d", &n);

    // 台形の面積の計算
    h = (b-a)/n;
    for (i=0; i<n; i++) {
        x = a + i*h;
        sum = sum + h*f(x);/* ここは台形則に書き換える必要がある */
    }
    printf("%f\n",sum);
}
```