

Cプログラミング入門

(基幹5クラス)

第9回 関数・数学ライブラリ

演習①

- ▶ int型変数 x を端末から入力し, $y = x^3 + 1$ の結果 y を標準出力するプログラム func1.c を作成せよ。
- ▶ ただし, main関数を以下の通り記述した上で, 自作の関数 **func** を #includeとmain関数の間に記述すること

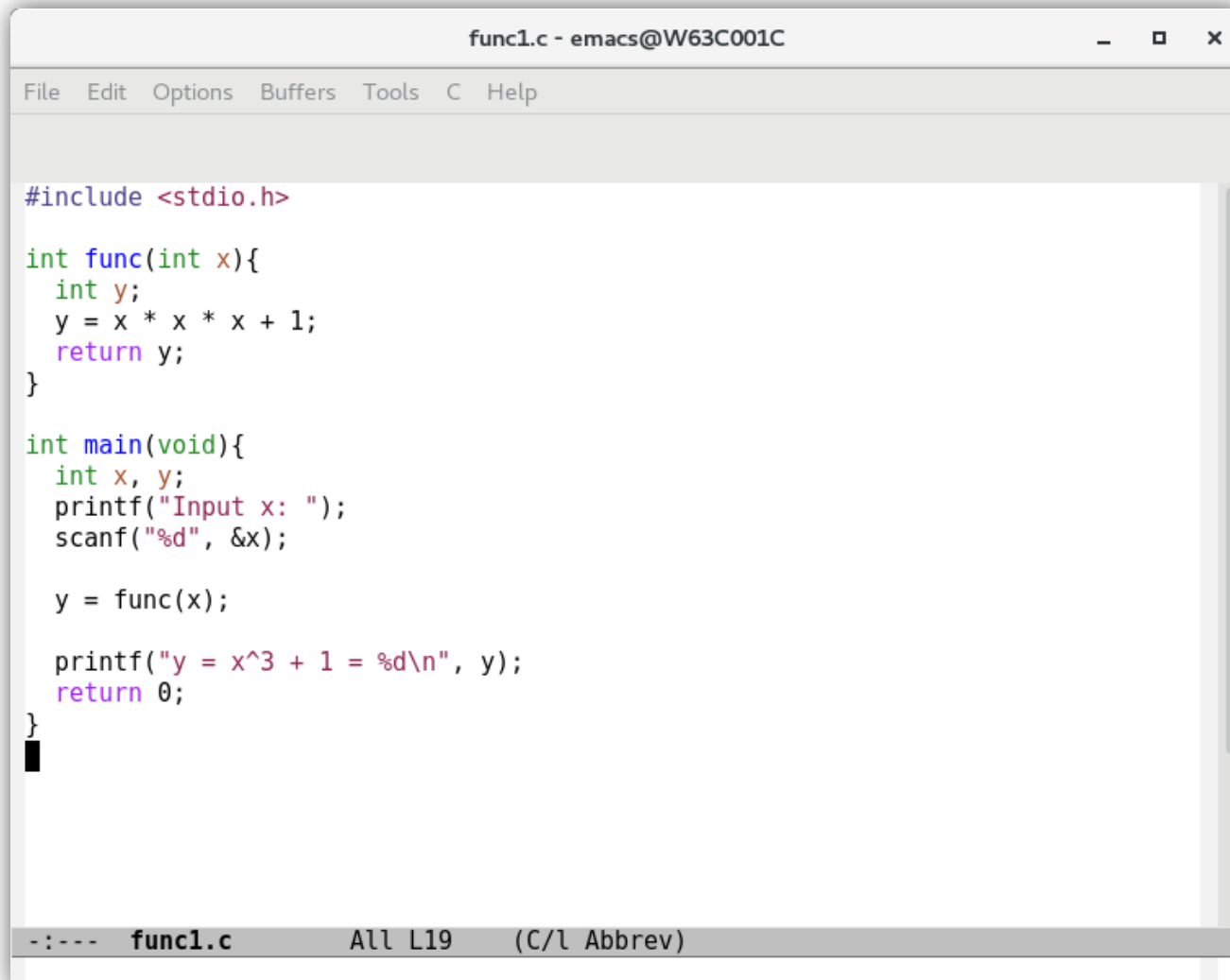
```
#include <stdio.h>

int main(void){
    int x, y;
    printf("Input x: ");
    scanf("%d", &x);

    y = func(x);

    printf("y = x^3 + 1 = %d¥n", y);
    return 0;
}
```

演習① ~プログラム例~



```
#include <stdio.h>

int func(int x){
    int y;
    y = x * x * x + 1;
    return y;
}

int main(void){
    int x, y;
    printf("Input x: ");
    scanf("%d", &x);

    y = func(x);

    printf("y = x^3 + 1 = %d\n", y);
    return 0;
}
```

-:--- func1.c All L19 (C/l Abbrev)

演習②

- ▶ 以下を参考に，端末から入力した3つの整数の**最大値**を標準出力するプログラム func2.c を作成せよ。
- ▶ **cp /share/func2.c ./** ... 講義中のみ実行できます

```
#include <stdio.h>

int max3(int a, int b, int c){
    

?

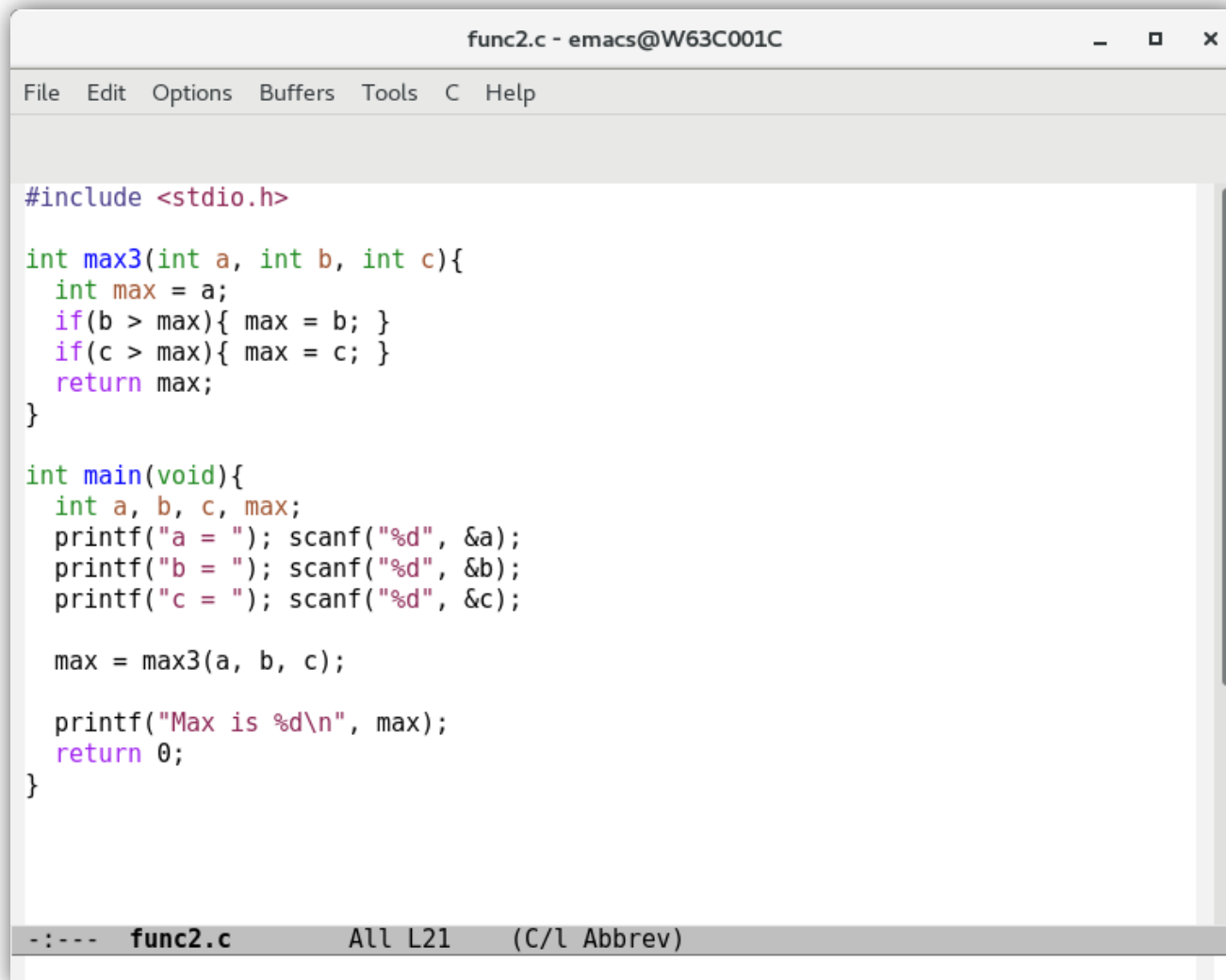

}

int main(void){
    int a, b, c, max;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);
    printf("c = "); scanf("%d", &c);

    max = max3(a, b, c);

    printf("Max is %d\n", max);
    return 0;
}
```

演習② ~プログラム例~



```
func2.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>

int max3(int a, int b, int c){
    int max = a;
    if(b > max){ max = b; }
    if(c > max){ max = c; }
    return max;
}

int main(void){
    int a, b, c, max;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);
    printf("c = "); scanf("%d", &c);

    max = max3(a, b, c);

    printf("Max is %d\n", max);
    return 0;
}

-:--- func2.c All L21 (C/l Abbrev)
```

演習③

- ▶ 以下を参考に，端末から入力した2つの整数の**差の絶対値**を標準出力するプログラム func3.c を作成せよ。
- ▶ **cp /share/func3.c ./** ... 講義中のみ実行できます

```
#include <stdio.h>

int diff2(int a, int b){
    

?


}

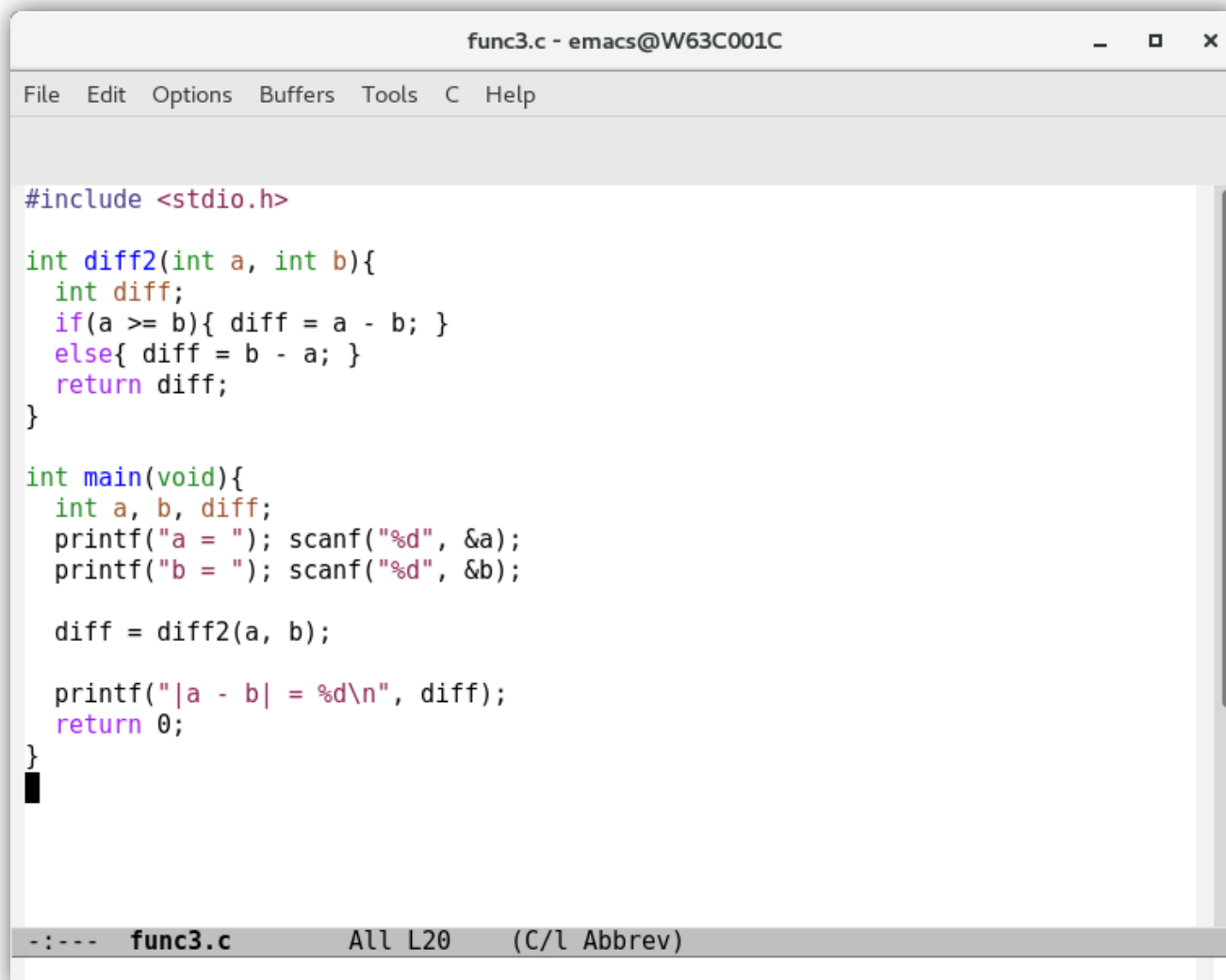
int main(void){
    int a, b, diff;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    diff = diff2(a, b);

    printf("|a - b| = %d\n", diff);
    return 0;
}
```

数学ライブラリ (math.h) を
使用しないで書くこと

演習③ ~プログラム例~



```
func3.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>

int diff2(int a, int b){
    int diff;
    if(a >= b){ diff = a - b; }
    else{ diff = b - a; }
    return diff;
}

int main(void){
    int a, b, diff;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    diff = diff2(a, b);

    printf("|a - b| = %d\n", diff);
    return 0;
}

-:--- func3.c All L20 (C/l Abbrev)
```

演習④

- ▶ 以下を参考に，端末から入力した2つの整数の**平均値**を標準出力するプログラム func4.c を作成せよ。
- ▶ **cp /share/func4.c ./** ... 講義中のみ実行できます

```
#include <stdio.h>

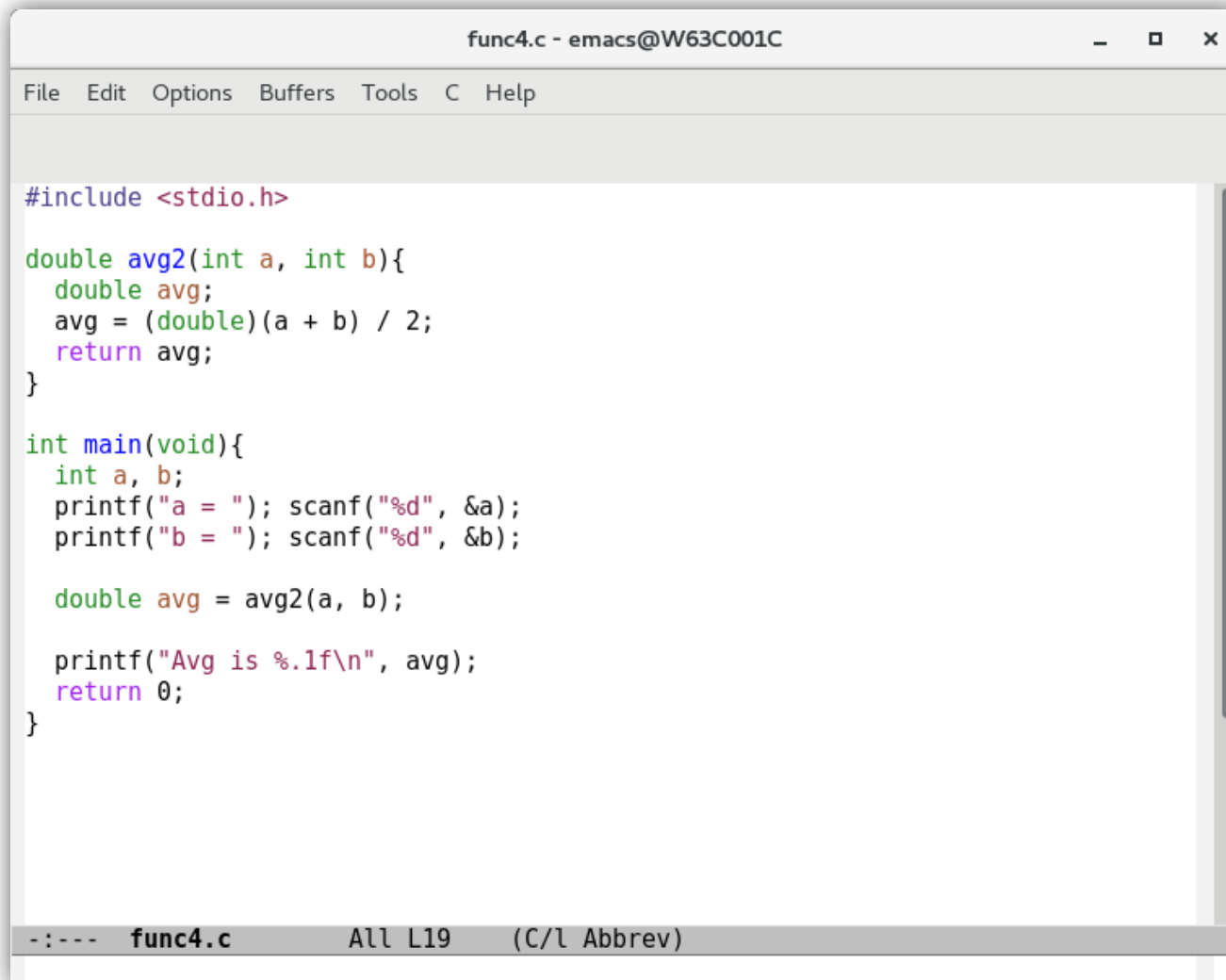
? avg2(int a, int b){
    ?
}

int main(void){
    int a, b;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    double avg = avg2(a, b);

    printf("Avg is %.1f¥n", avg);
    return 0;
}
```


演習④ ~プログラム例~



The image shows a screenshot of an Emacs editor window titled "func4.c - emacs@W63C001C". The window contains a C program that calculates the average of two integers. The code is as follows:

```
#include <stdio.h>

double avg2(int a, int b){
    double avg;
    avg = (double)(a + b) / 2;
    return avg;
}

int main(void){
    int a, b;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    double avg = avg2(a, b);

    printf("Avg is %.1f\n", avg);
    return 0;
}
```

The status bar at the bottom of the window displays "func4.c", "All L19", and "(C/l Abbrev)".

演習⑤

- ▶ 端末から入力した整数 x, y ($y > 0$ とする) にもとづいて x^y を計算するプログラム func5.c を作成せよ。
- ▶ **cp /share/func5.c ./** ... 講義中のみ実行できます

```
#include <stdio.h>

int pow2(int x, int y){
    

?


}

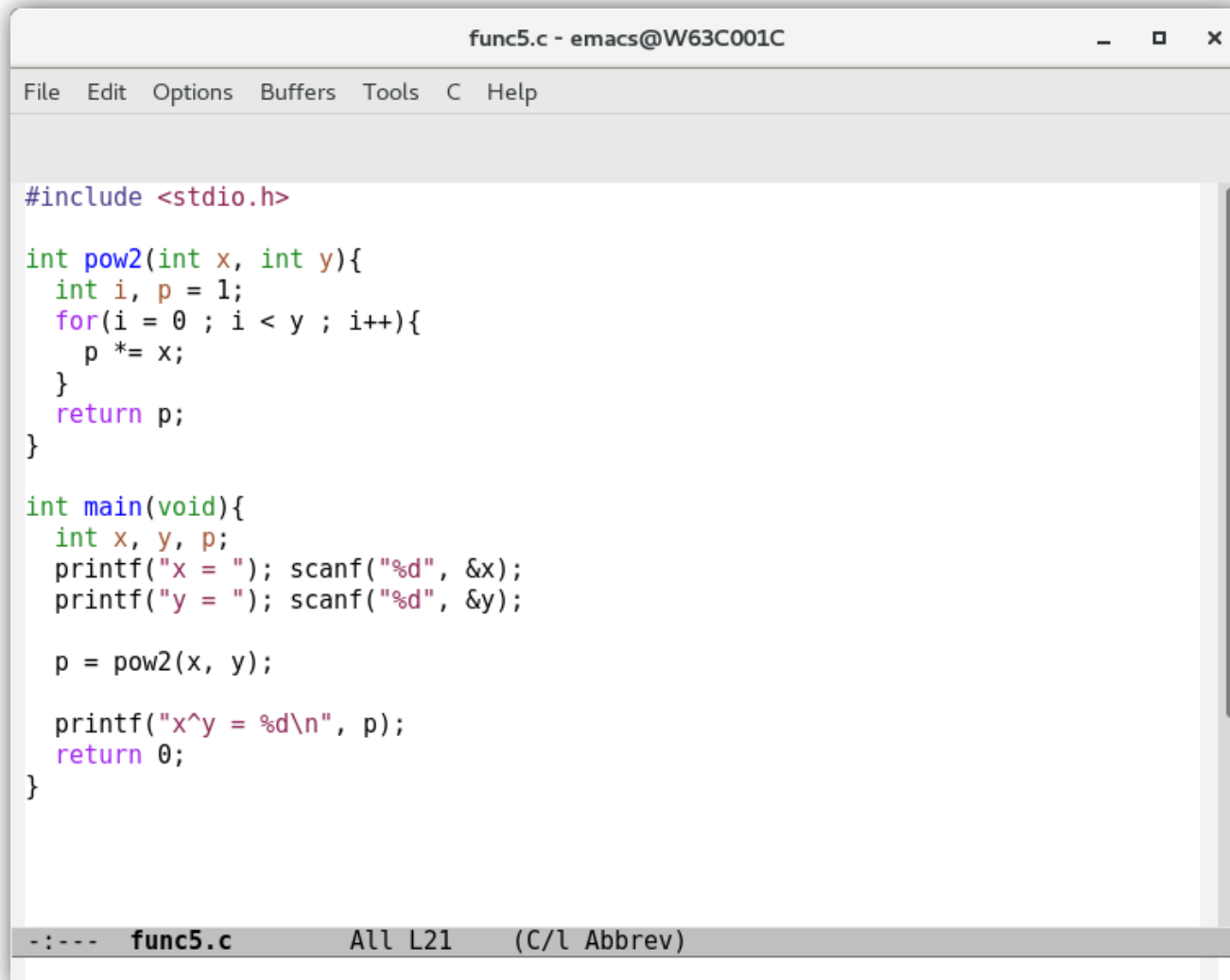
int main(void){
    int x, y, p;
    printf("x = "); scanf("%d", &x);
    printf("y = "); scanf("%d", &y);

    p = pow2(x, y);

    printf("x^y = %d\n", p);
    return 0;
}
```

数学ライブラリ (math.h) を
使用しないで書くこと

演習⑤ ~プログラム例~



```
func5.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>

int pow2(int x, int y){
    int i, p = 1;
    for(i = 0 ; i < y ; i++){
        p *= x;
    }
    return p;
}

int main(void){
    int x, y, p;
    printf("x = "); scanf("%d", &x);
    printf("y = "); scanf("%d", &y);

    p = pow2(x, y);

    printf("x^y = %d\n", p);
    return 0;
}

-:--- func5.c      All L21      (C/l Abbrev)
```

演習⑥

- ▶ 端末から入力した2つの整数の**差の絶対値**を標準出力するプログラム math1.c を数学ライブラリ内の関数 fabs を用いて作成せよ。(演習③と同じ動作)

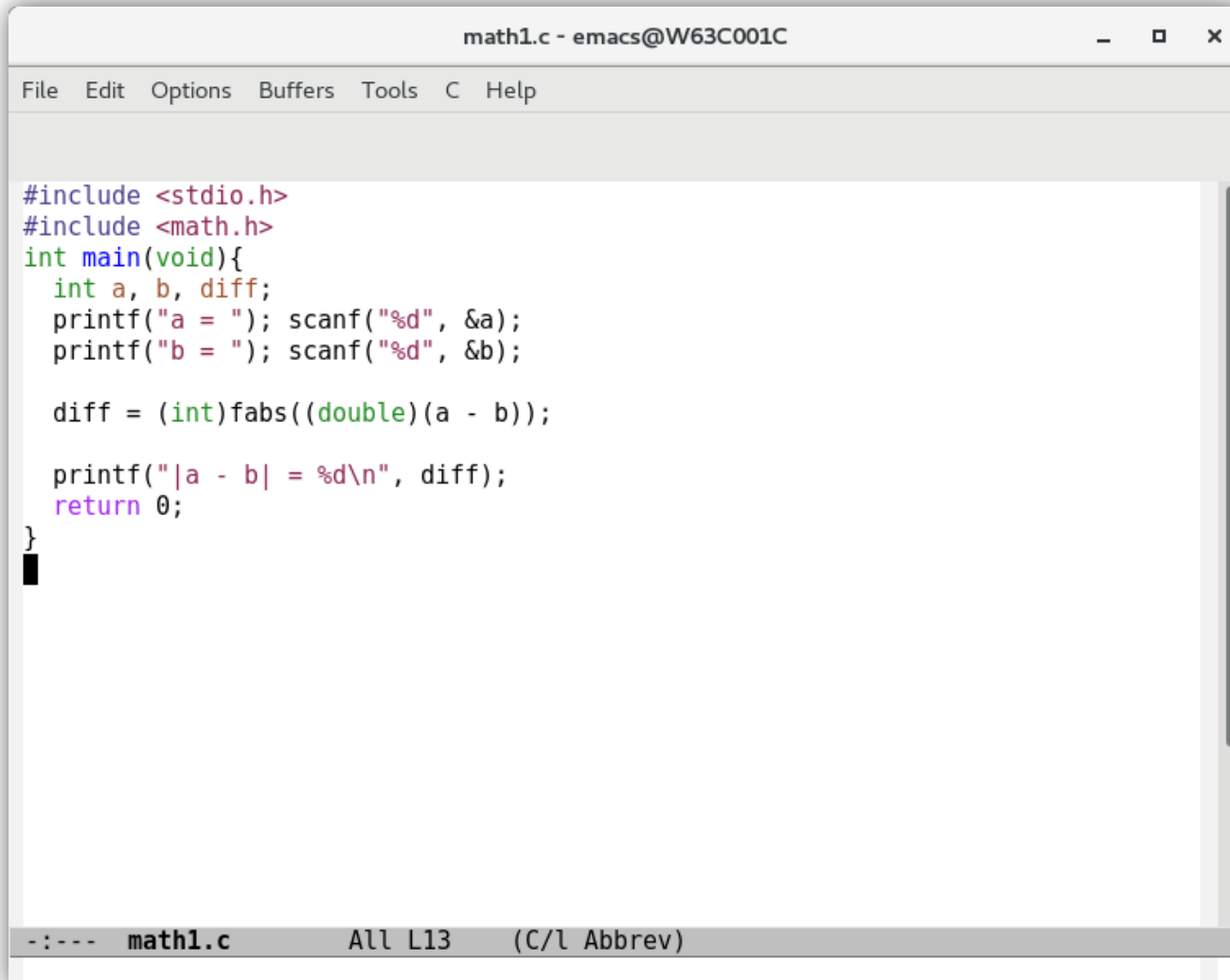
```
#include <stdio.h>
#include <math.h>
int main(void){
    int a, b, diff;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    diff = 

    printf("|a - b| = %d\n", diff);
    return 0;
}
```



演習⑥ ~プログラム例~



```
math1.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#include <math.h>
int main(void){
    int a, b, diff;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);

    diff = (int)fabs((double)(a - b));

    printf("|a - b| = %d\n", diff);
    return 0;
}
```

--- math1.c All L13 (C/l Abbrev)

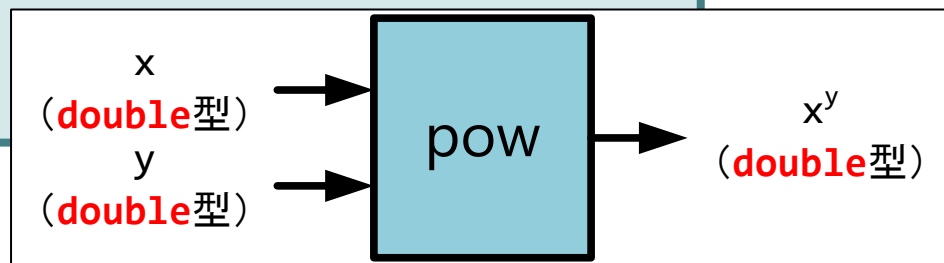
演習⑦

- ▶ 端末から入力した整数 x, y ($y > 0$ とする) にもとづいて x^y を計算するプログラム math2.c を数学ライブラリ内の関数 pow を用いて作成せよ。(演習⑤と同じ動作)

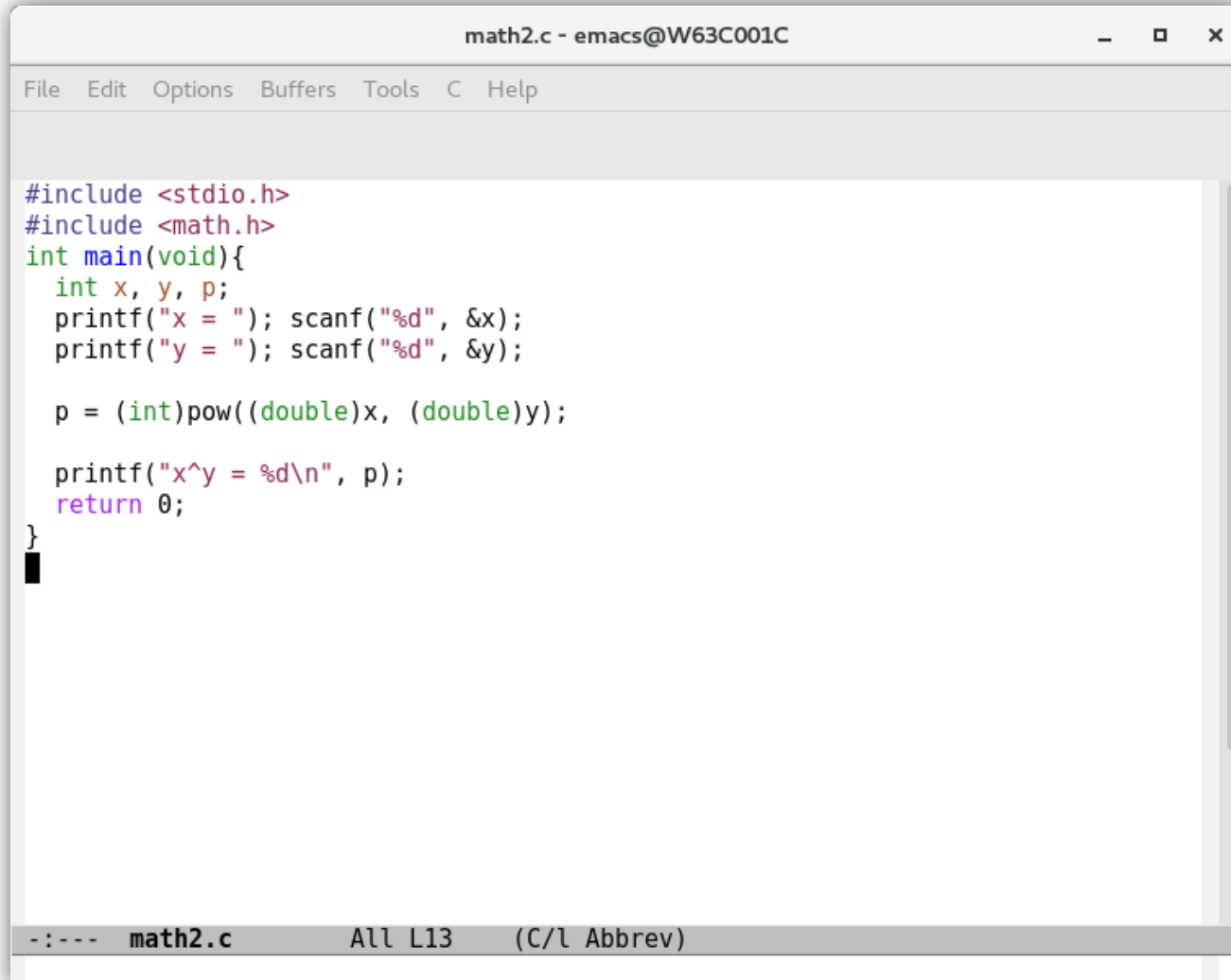
```
#include <stdio.h>
#include <math.h>
int main(void){
    int x, y, p;
    printf("x = "); scanf("%d", &x);
    printf("y = "); scanf("%d", &y);

    p = 

    printf("x^y = %d\n", p);
    return 0;
}
```



演習⑦ ~プログラム例~



The image shows a screenshot of an Emacs editor window titled "math2.c - emacs@W63C001C". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The main editing area contains the following C code:

```
#include <stdio.h>
#include <math.h>
int main(void){
    int x, y, p;
    printf("x = "); scanf("%d", &x);
    printf("y = "); scanf("%d", &y);

    p = (int)pow((double)x, (double)y);

    printf("x^y = %d\n", p);
    return 0;
}
```

The status bar at the bottom of the window displays "--:--- math2.c All L13 (C/l Abbrev)".

演習⑧（発展問題）

- ▶ 円周率の値をプログラムで求めるための方法のひとつとして、**ライプニッツの公式**を用いる方法がある。

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \lim_{N \rightarrow \infty} \sum_{i=0}^N \frac{(-1)^i}{2i+1} = \frac{\pi}{4}$$

- ▶ 本公式を用いて近似的に円周率の値を求め、それを数学ライブラリ内の定数M_PIと比較することで誤差が 10^{-5} 未満となるような最小の*N*を求めたい。
- ▶ *N*を求めるプログラム Leibniz.c を作成せよ。

演習⑧－ 1

$$4 \times \sum_{i=0}^N \frac{(-1)^i}{2i+1}$$

- ▶ 端末から入力した N をもとに、円周率の近似値を求め、誤差を計算するプログラムを作成せよ。
- ▶ ここでは、近似値と定数M_PIの差の絶対値を誤差と定義する

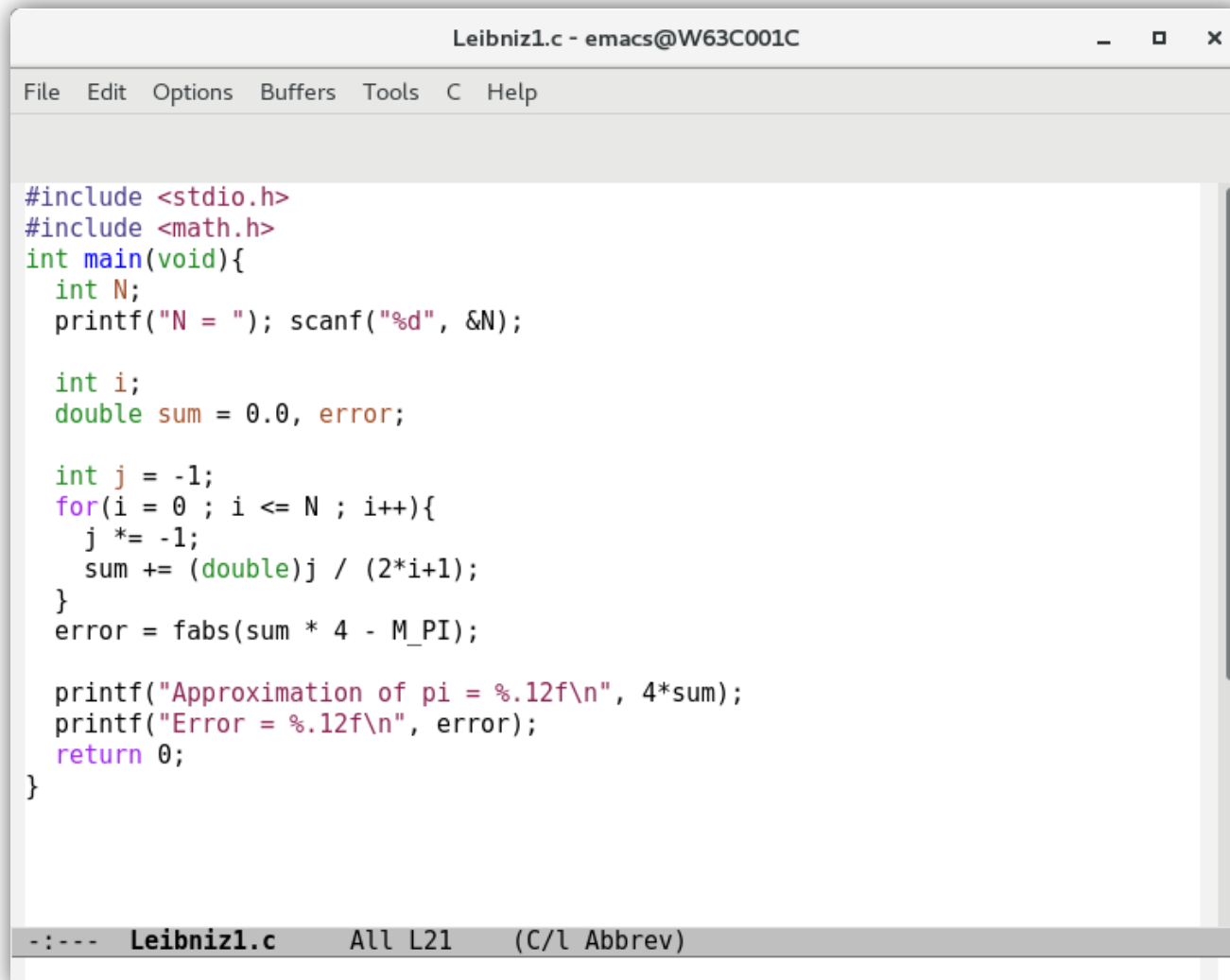
```
#include <stdio.h>
#include <math.h>
int main(void){
    int N;
    printf("N = "); scanf("%d", &N);

    int i;
    double sum = 0.0; error;

    printf("Approximation of pi = %.12f¥n", 4*sum);
    printf("Error = %.12f¥n", error);
    return 0;
}
```

演習⑧ – 1 ~プログラム例~



```
Leibniz1.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#include <math.h>
int main(void){
    int N;
    printf("N = "); scanf("%d", &N);

    int i;
    double sum = 0.0, error;

    int j = -1;
    for(i = 0 ; i <= N ; i++){
        j *= -1;
        sum += (double)j / (2*i+1);
    }
    error = fabs(sum * 4 - M_PI);

    printf("Approximation of pi = %.12f\n", 4*sum);
    printf("Error = %.12f\n", error);
    return 0;
}
```

--- Leibniz1.c All L21 (C/l Abbrev)

演習⑧ – 2

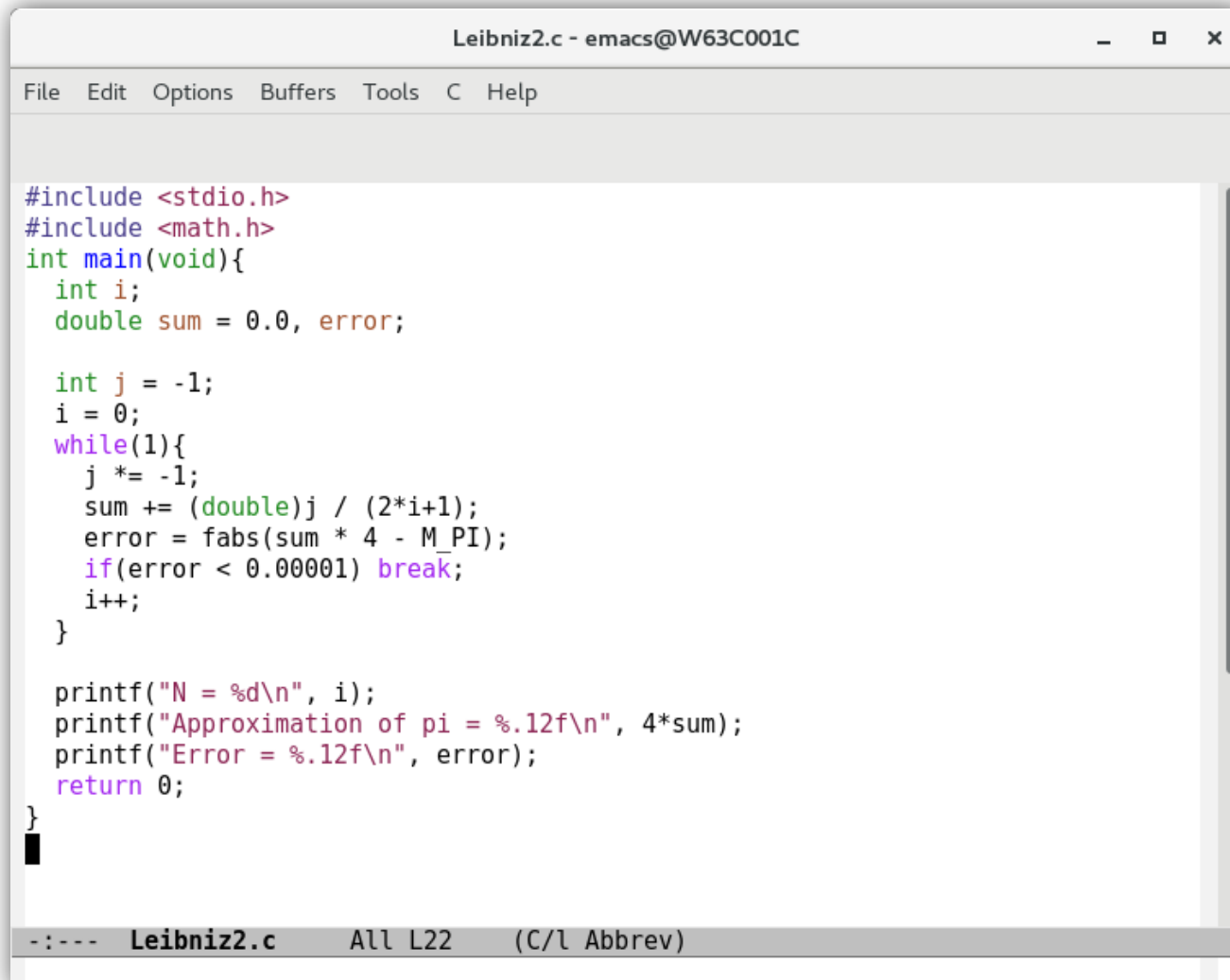
- ▶ 演習⑧ – 1 のプログラムを書き換え，誤差が 10^{-5} 未満となるような最小の N を求めるプログラムを作成せよ。

```
#include <stdio.h>
#include <math.h>
int main(void){
    int i;
    double sum = 0.0; error;

    ?

    printf("N = %d\n", ? );
    printf("Approximation of pi = %.12f\n", 4*sum);
    printf("Error = %.12f\n", error);
    return 0;
}
```

演習⑧ – 2 ~プログラム例~



```
Leibniz2.c - emacs@W63C001C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#include <math.h>
int main(void){
    int i;
    double sum = 0.0, error;

    int j = -1;
    i = 0;
    while(1){
        j *= -1;
        sum += (double)j / (2*i+1);
        error = fabs(sum * 4 - M_PI);
        if(error < 0.00001) break;
        i++;
    }

    printf("N = %d\n", i);
    printf("Approximation of pi = %.12f\n", 4*sum);
    printf("Error = %.12f\n", error);
    return 0;
}

-:--- Leibniz2.c All L22 (C/l Abbrev)
```