

Cプログラミング入門

(基幹5クラス)

第10回 ポインタ

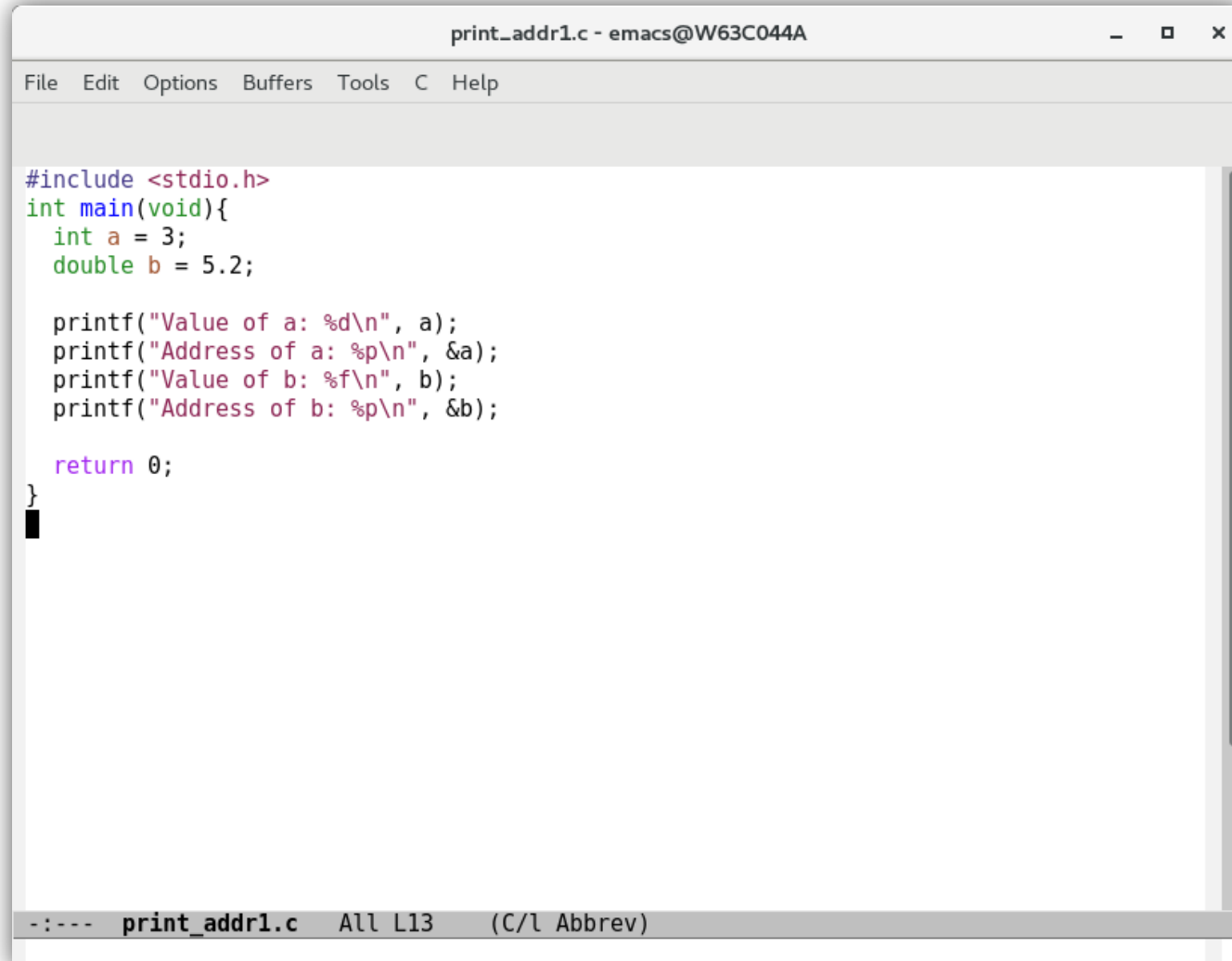
演習①

- ▶ int型変数a, double型変数bを宣言し, それぞれの値とアドレスを表示するプログラム print_addr1.c を作成せよ。

```
#include <stdio.h>

int main(void){
    int a = 3;
    double b = 5.2;
    printf("Value of a: [?] %n", [?]);
    printf("Address of a: [?] %n", [?]);
    printf("Value of b: [?] %n", [?]);
    printf("Address of b: [?] %n", [?]);
    return 0;
}
```

演習① ~プログラム~



```
print_addr1.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>
int main(void){
    int a = 3;
    double b = 5.2;

    printf("Value of a: %d\n", a);
    printf("Address of a: %p\n", &a);
    printf("Value of b: %f\n", b);
    printf("Address of b: %p\n", &b);

    return 0;
}

-:--- print_addr1.c  All L13  (C/l Abbrev)
```

演習②

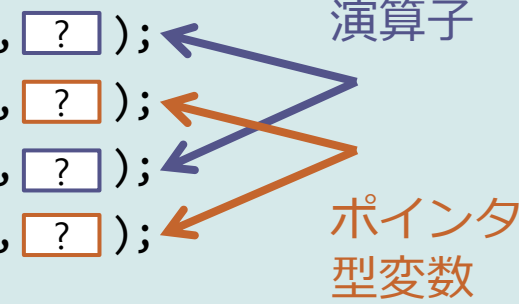
- ▶ int型変数a, double型変数bを宣言し, それぞれのアドレスを表示するプログラム print_addr2.c を作成せよ。
- ▶ 次スライドを参考に, 二通りの方法でアドレスを表示させてみる。
 - ▶ 変数a, bを用いた表示 (演習①と同じ方法)
 - ▶ ポインタ型変数pa, pbを用いた表示

演習②

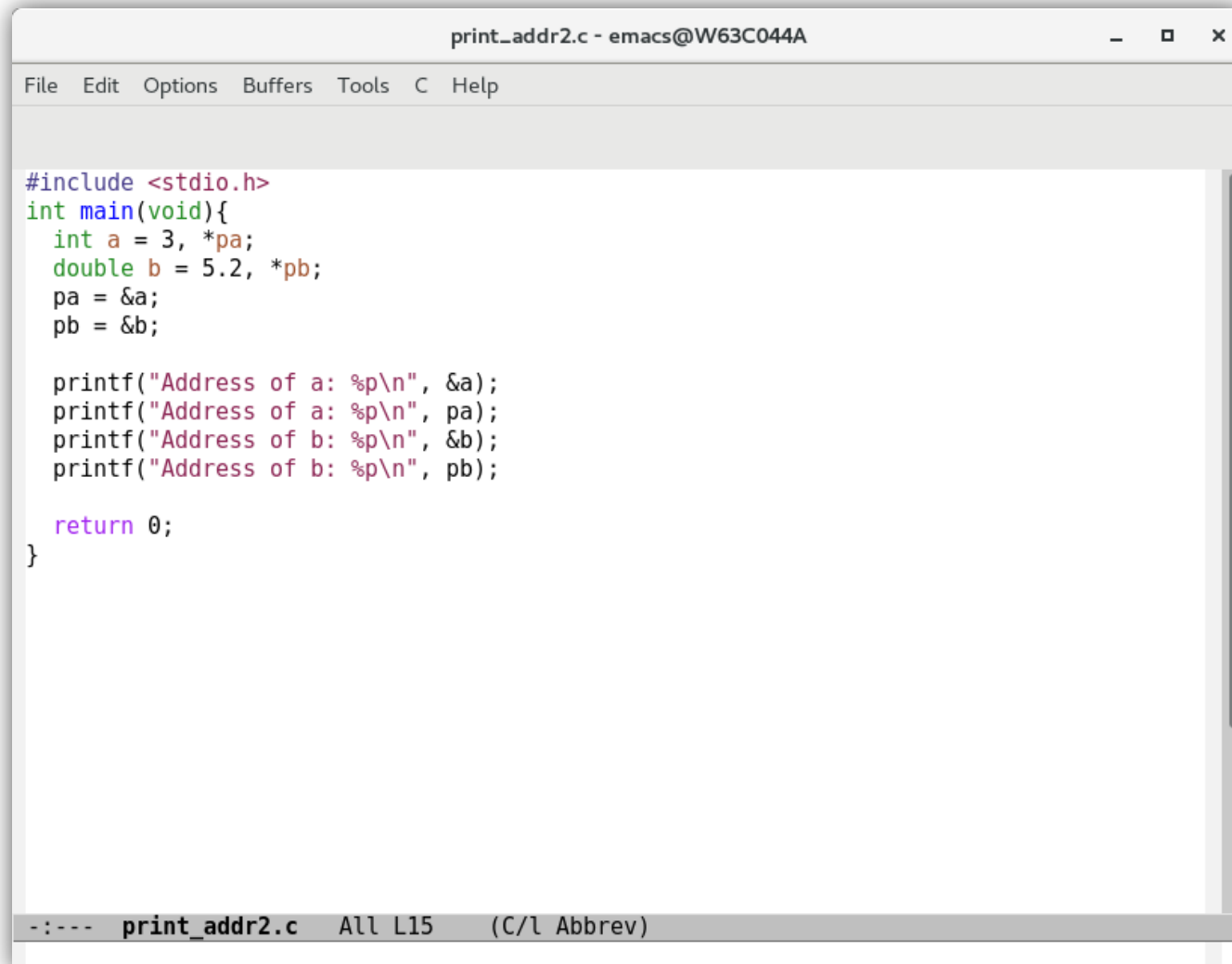
```
#include <stdio.h>
int main(void){
    int a = 3, *pa;
    double b = 5.2, *pb;
    pa = &a;
    pb = &b;
    printf("Address of a: [?] ¥n", [?] );
    printf("Address of a: [?] ¥n", [?] );
    printf("Address of b: [?] ¥n", [?] );
    printf("Address of b: [?] ¥n", [?] );
    return 0;
}
```

アドレス演算子

ポインタ型変数



演習② ~プログラム~



```
print_addr2.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>
int main(void){
    int a = 3, *pa;
    double b = 5.2, *pb;
    pa = &a;
    pb = &b;

    printf("Address of a: %p\n", &a);
    printf("Address of a: %p\n", pa);
    printf("Address of b: %p\n", &b);
    printf("Address of b: %p\n", pb);

    return 0;
}

-:--- print_addr2.c  All L15  (C/l Abbrev)
```

演習③

- ▶ int型変数a, double型変数bを宣言し, それぞれの値を表示するプログラム print_value.c を作成せよ。
- ▶ 次スライドを参考に, 二通りの方法で値を表示させてみる。
 - ▶ 変数a, bを用いた表示 (演習①と同じ方法)
 - ▶ ポインタ型変数pa, pbと間接参照演算子を用いた表示

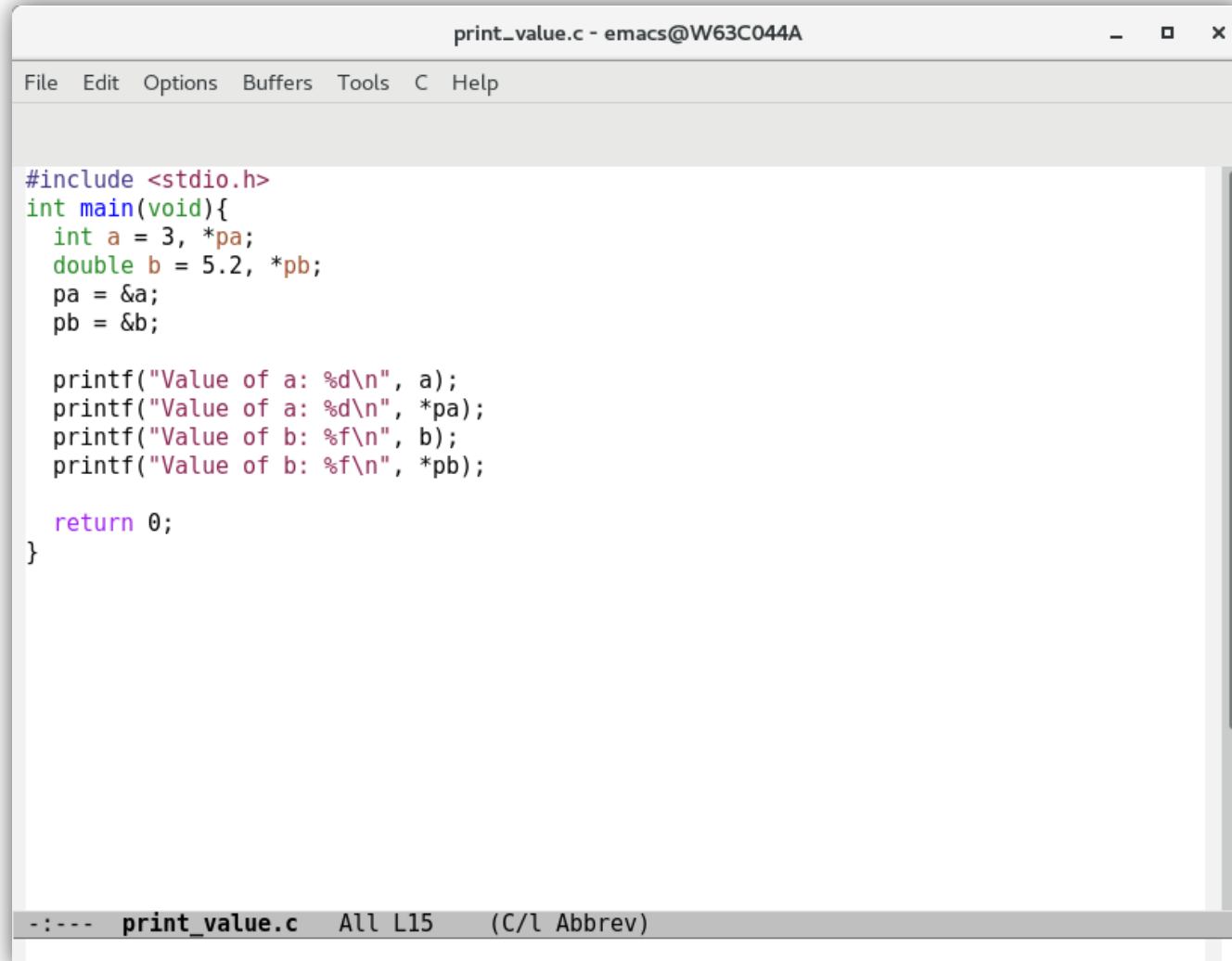
演習③

```
#include <stdio.h>
int main(void){
    int a = 3, *pa;
    double b = 5.2, *pb;
    pa = &a;
    pb = &b;
    printf("Value of a: [?] ¥n", [?]);
    printf("Value of a: [?] ¥n", [?]);
    printf("Value of b: [?] ¥n", [?]);
    printf("Value of b: [?] ¥n", [?]);
    return 0;
}
```

変数

ポインタ型変数と
間接参照演算子

演習③ ~プログラム~



```
print_value.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>
int main(void){
    int a = 3, *pa;
    double b = 5.2, *pb;
    pa = &a;
    pb = &b;

    printf("Value of a: %d\n", a);
    printf("Value of a: %d\n", *pa);
    printf("Value of b: %f\n", b);
    printf("Value of b: %f\n", *pb);

    return 0;
}

-:--- print_value.c  All L15  (C/l Abbrev)
```

演習④

- ▶ 端末から入力した 3 つの整数の**最大値と最小値**を標準出力するプログラム pfunc1.c を作成せよ。

```
#include <stdio.h>

void comp3(int a, int b, int c, int *pmax, int *pmin){
    

?

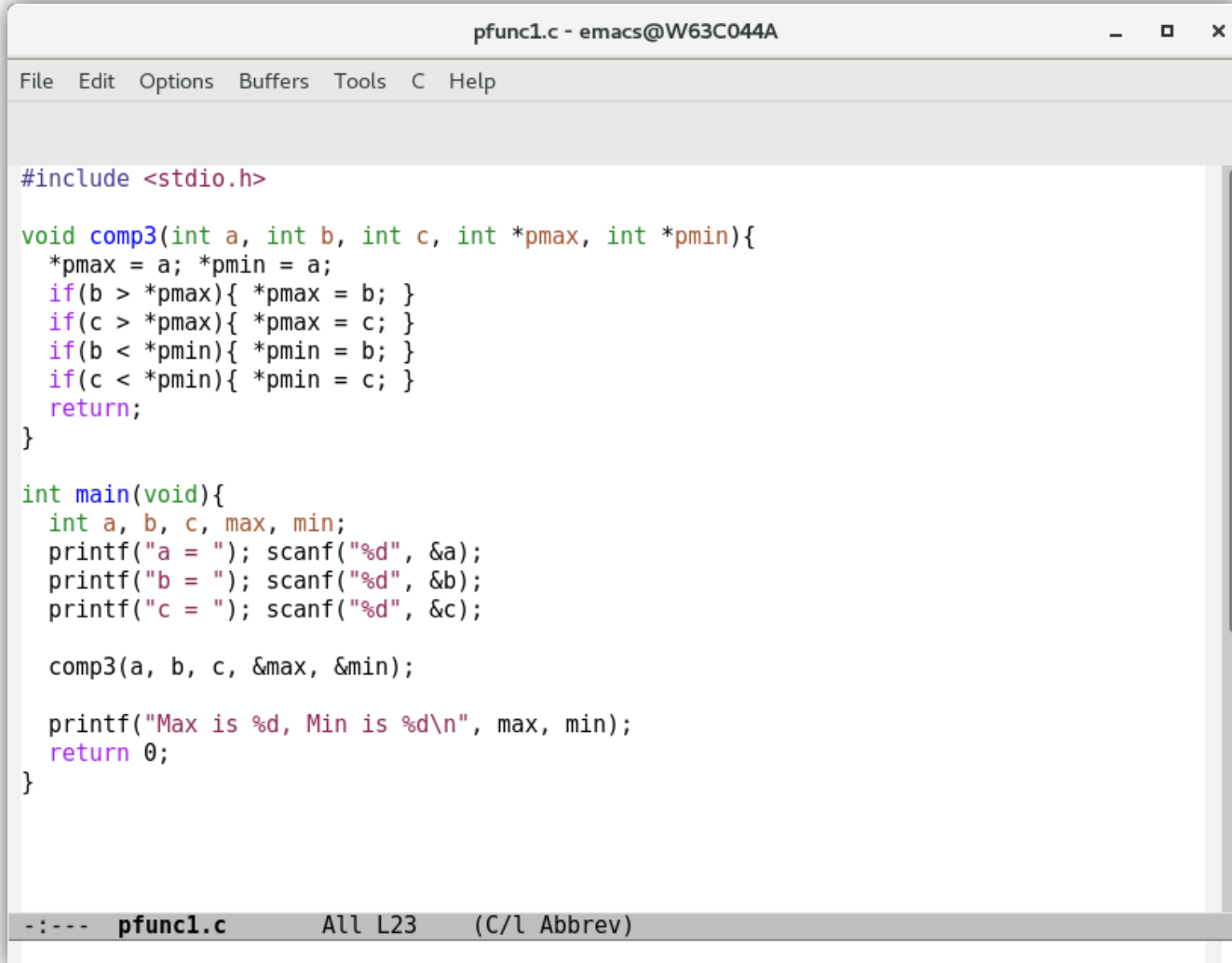

    return;
}

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

    comp3(a, b, c, &max, &min);

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

演習④ ~プログラム例~



```

pfunc1.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>

void comp3(int a, int b, int c, int *pmax, int *pmin){
    *pmax = a; *pmin = a;
    if(b > *pmax){ *pmax = b; }
    if(c > *pmax){ *pmax = c; }
    if(b < *pmin){ *pmin = b; }
    if(c < *pmin){ *pmin = c; }
    return;
}

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

    comp3(a, b, c, &max, &min);

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

-:--- pfunc1.c      All L23      (C/l Abbrev)
```

演習⑤

- ▶ 端末から入力した 2 つの整数を**入れ替えて**標準出力するプログラム pfunc2.c を作成せよ。

```
#include <stdio.h>

void swap(int *pa, int *pb){
    

?

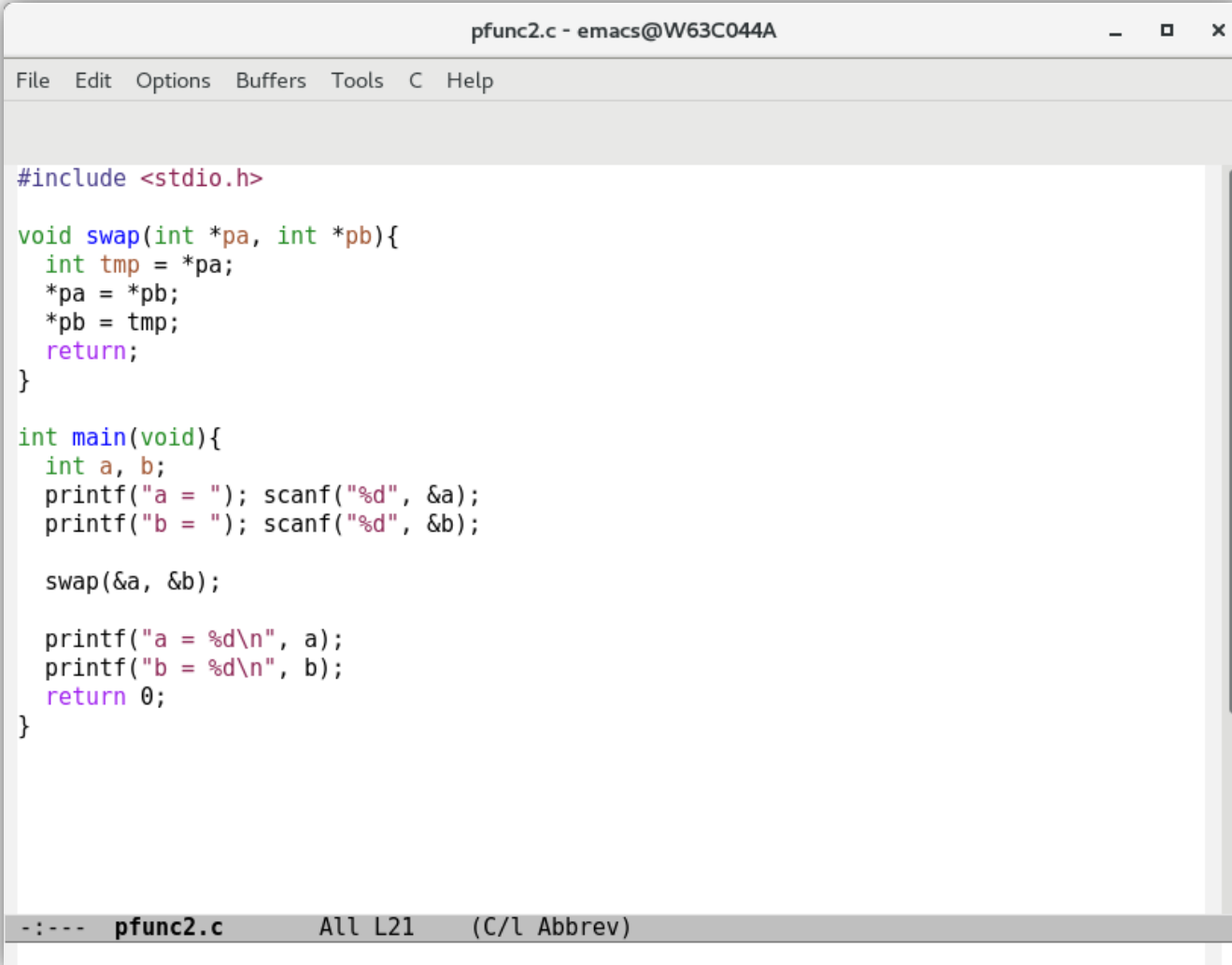

    return;
}

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

    swap(&a, &b);

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

演習⑤ ~プログラム例~



The screenshot shows an Emacs editor window titled "pfunc2.c - emacs@W63C044A". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The code is as follows:

```
#include <stdio.h>

void swap(int *pa, int *pb){
    int tmp = *pa;
    *pa = *pb;
    *pb = tmp;
    return;
}

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

    swap(&a, &b);

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

The status bar at the bottom displays "-:--- pfunc2.c All L21 (C/l Abbrev)".

演習⑥

- ▶ 端末から入力した 3 つの整数を**降順に並び替えて**標準出力するプログラム pfunc3.c を作成せよ。

```
#include <stdio.h>

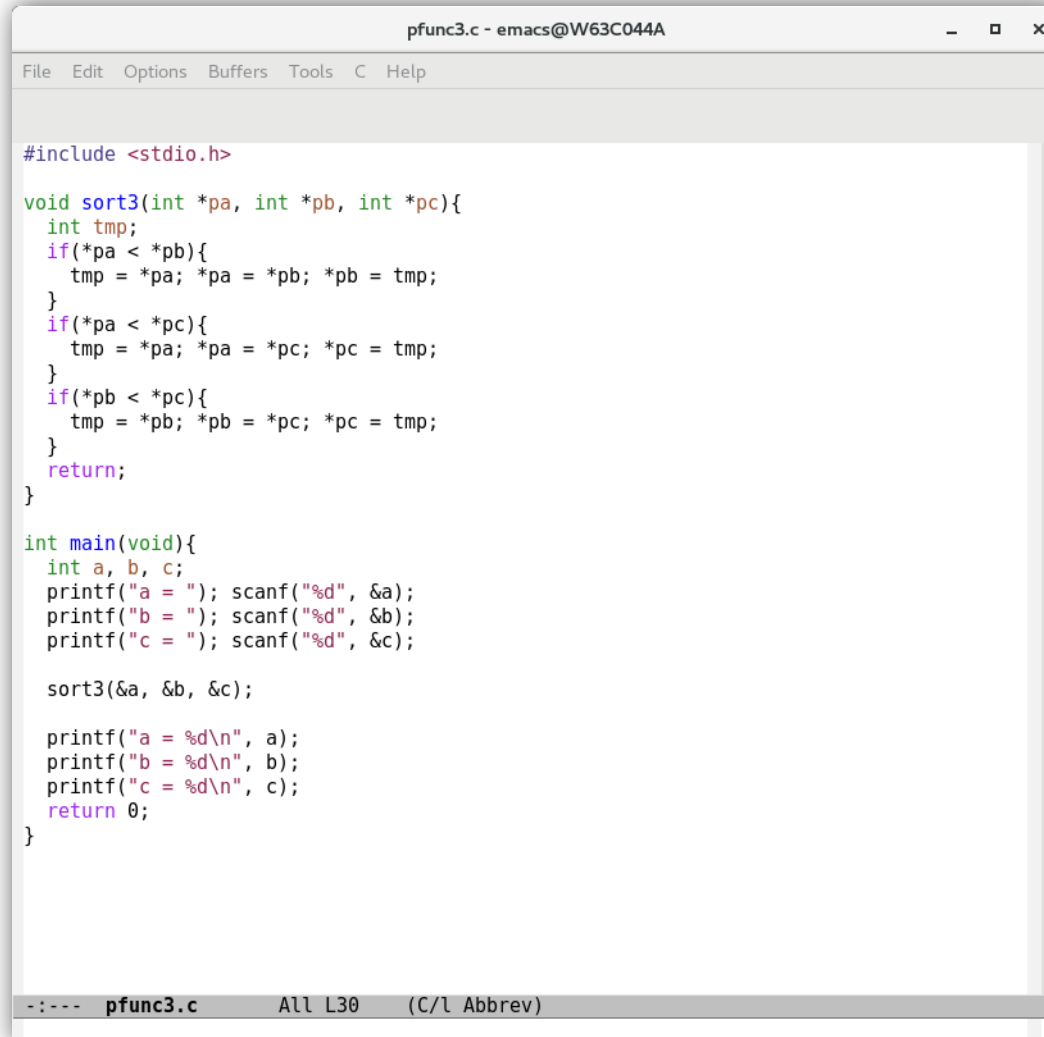
void sort3(){
    
}

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

    sort3(&a, &b, &c);

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

演習⑥ ~プログラム例 1~



```

pfunc3.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>

void sort3(int *pa, int *pb, int *pc){
    int tmp;
    if(*pa < *pb){
        tmp = *pa; *pa = *pb; *pb = tmp;
    }
    if(*pa < *pc){
        tmp = *pa; *pa = *pc; *pc = tmp;
    }
    if(*pb < *pc){
        tmp = *pb; *pb = *pc; *pc = tmp;
    }
    return;
}

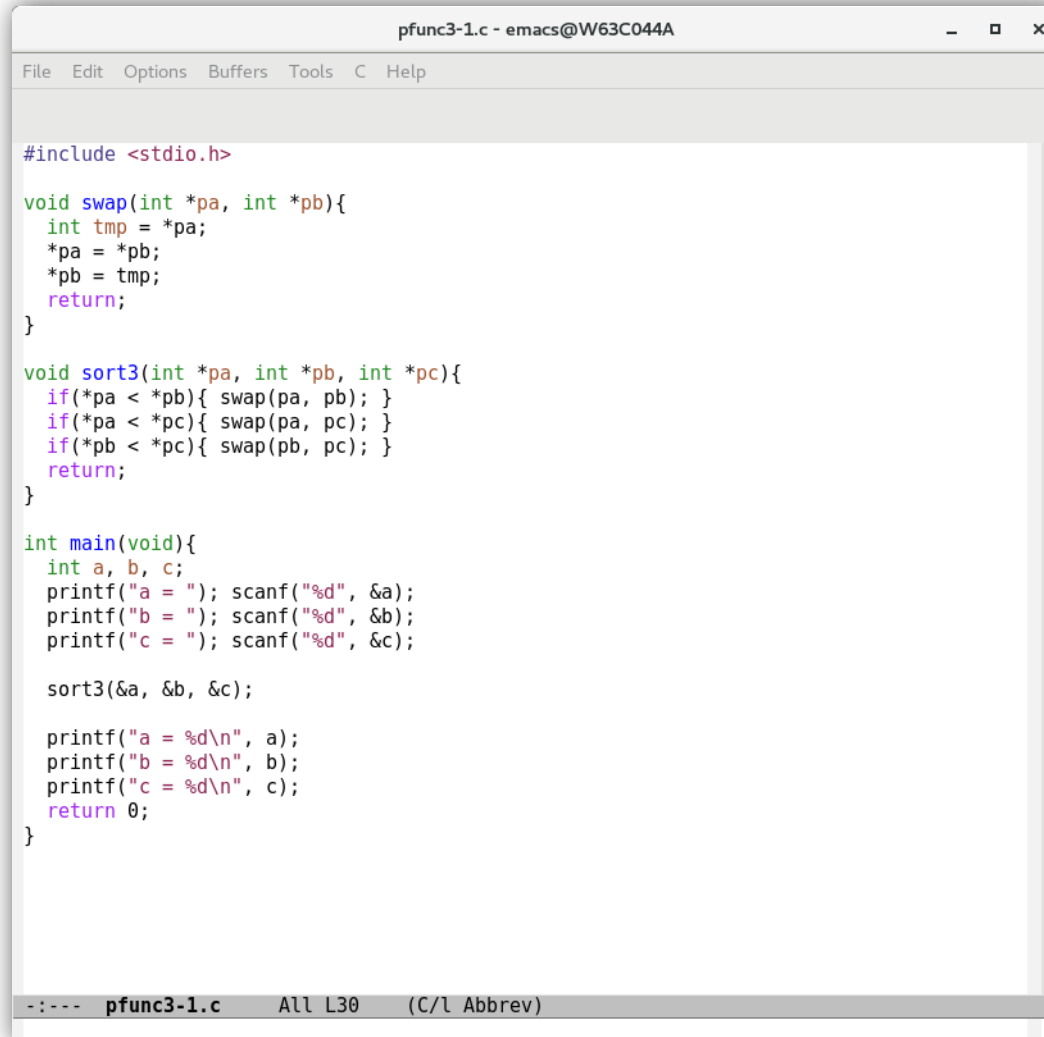
int main(void){
    int a, b, c;
    printf("a = "); scanf("%d", &a);
    printf("b = "); scanf("%d", &b);
    printf("c = "); scanf("%d", &c);

    sort3(&a, &b, &c);

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

-:--- pfunc3.c All L30 (C/l Abbrev)
```

演習⑥ ~プログラム例 2~



```
pfunc3-1.c - emacs@W63C044A
File Edit Options Buffers Tools C Help

#include <stdio.h>

void swap(int *pa, int *pb){
    int tmp = *pa;
    *pa = *pb;
    *pb = tmp;
    return;
}

void sort3(int *pa, int *pb, int *pc){
    if(*pa < *pb){ swap(pa, pb); }
    if(*pa < *pc){ swap(pa, pc); }
    if(*pb < *pc){ swap(pb, pc); }
    return;
}

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

    sort3(&a, &b, &c);

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

-:--- pfunc3-1.c All L30 (C/l Abbrev)
```


演習⑦－ 1（余力がある人向け）

- ▶ 端末から入力したdouble型変数 $a \geq 0$ の平方根 \sqrt{a} を求めるプログラムを数学ライブラリを用いて作成せよ。

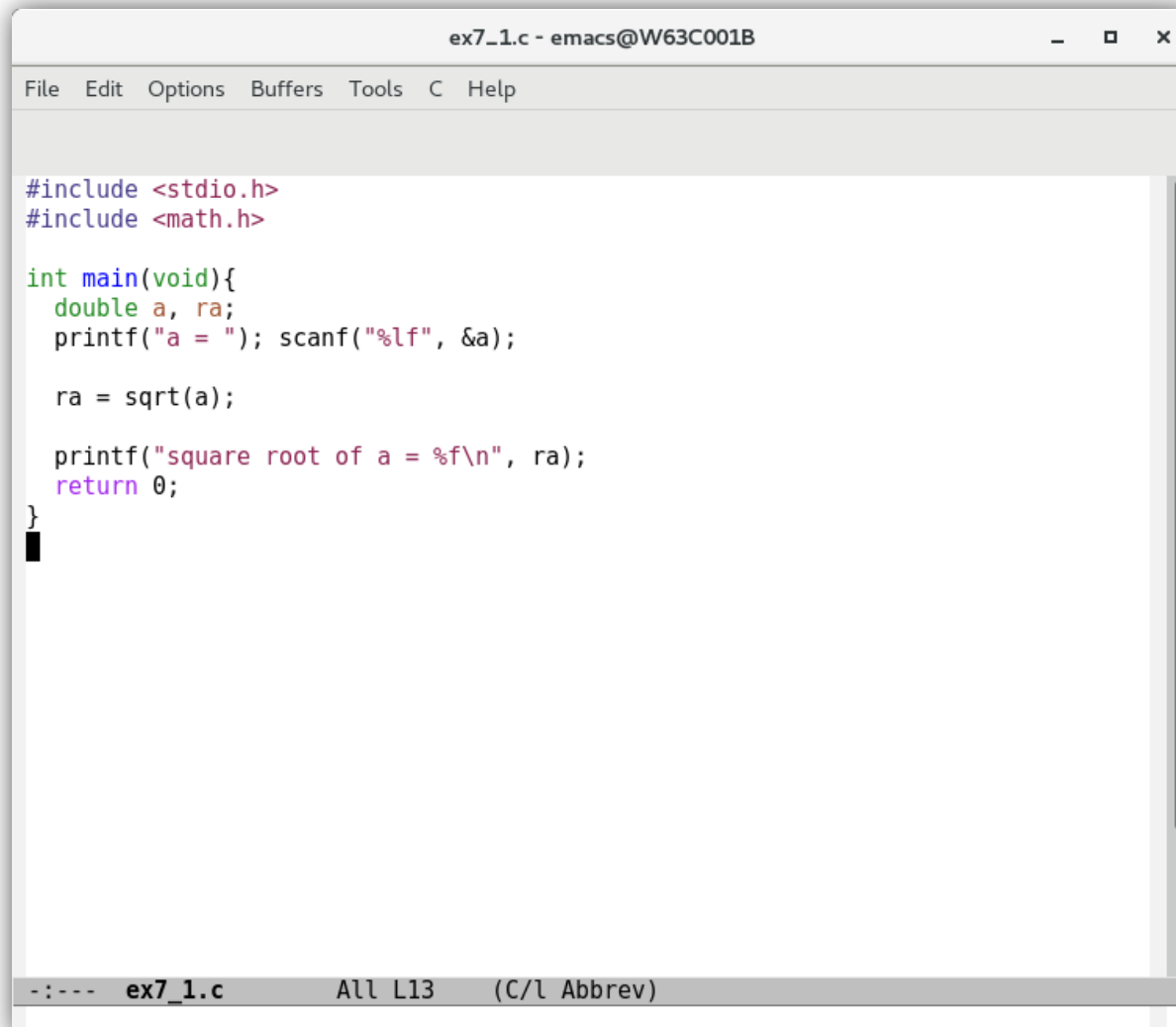
```
#include <stdio.h>
#include <math.h>

int main(void){
    double a, ra;
    printf("a = "); scanf("%lf", &a);

    printf("square root of a = %f\n", ra);
    return 0;
}
```

演習⑦ – 1 ~プログラム~



The image shows a screenshot of an Emacs editor window. The title bar at the top reads "ex7_1.c - emacs@W63C001B". Below the title bar is a menu bar with the following items: 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){
    double a, ra;
    printf("a = "); scanf("%lf", &a);

    ra = sqrt(a);

    printf("square root of a = %f\n", ra);
    return 0;
}
```

The code is color-coded: preprocessor directives are purple, the function signature is green, variable declarations are green, and the rest of the code is black. A vertical scrollbar is visible on the right side of the editing area. At the bottom of the window, a status bar shows the following information: -:--- ex7_1.c All L13 (C/l Abbrev).

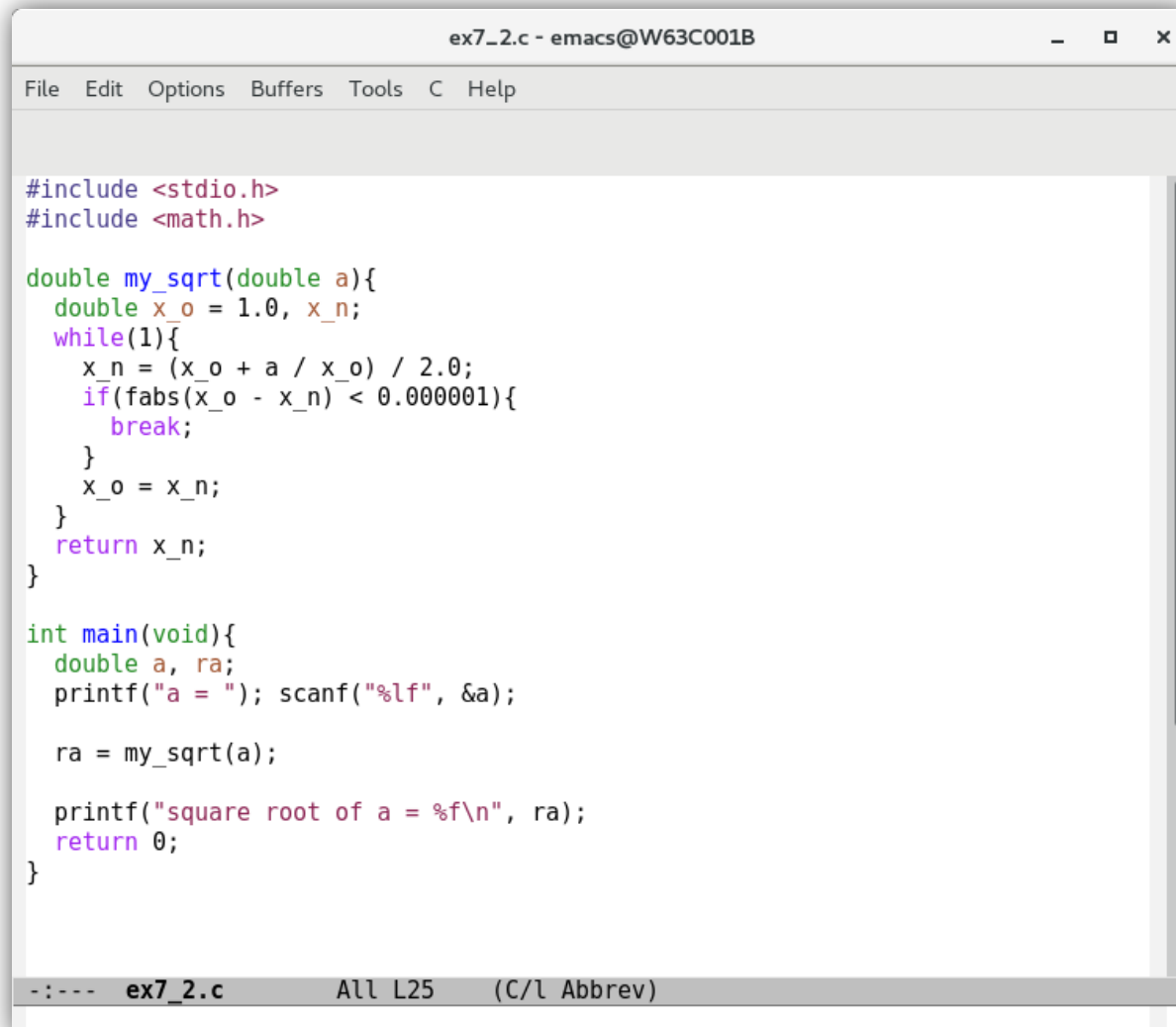
演習⑦－ 2 （余力がある人向け）

- ▶ 端末から入力したdouble型変数 $a \geq 0$ の平方根 \sqrt{a} を求める関数 **double my_sqrt(double a)** を自作せよ。
 - ▶ 作成したら演習⑦－ 1 と実行結果を比較してみる
- ▶ 以下の漸化式（**ニュートン法**）を用いて \sqrt{a} の近似値を求めることができる。

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{2}{x_n} \right), \quad x_1 = 1.0$$

- ▶ 補足
 - ▶ $x_1 = 1.0$ としているが, $x_1 > 0$ の任意の実数を指定可能
 - ▶ 収束条件は $|x_{n+1} - x_n| < 0.000001$ とする

演習⑦ – 2 ~プログラム例~



```
ex7_2.c - emacs@W63C001B
File Edit Options Buffers Tools C Help

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

double my_sqrt(double a){
    double x_o = 1.0, x_n;
    while(1){
        x_n = (x_o + a / x_o) / 2.0;
        if(fabs(x_o - x_n) < 0.000001){
            break;
        }
        x_o = x_n;
    }
    return x_n;
}

int main(void){
    double a, ra;
    printf("a = "); scanf("%lf", &a);

    ra = my_sqrt(a);

    printf("square root of a = %f\n", ra);
    return 0;
}

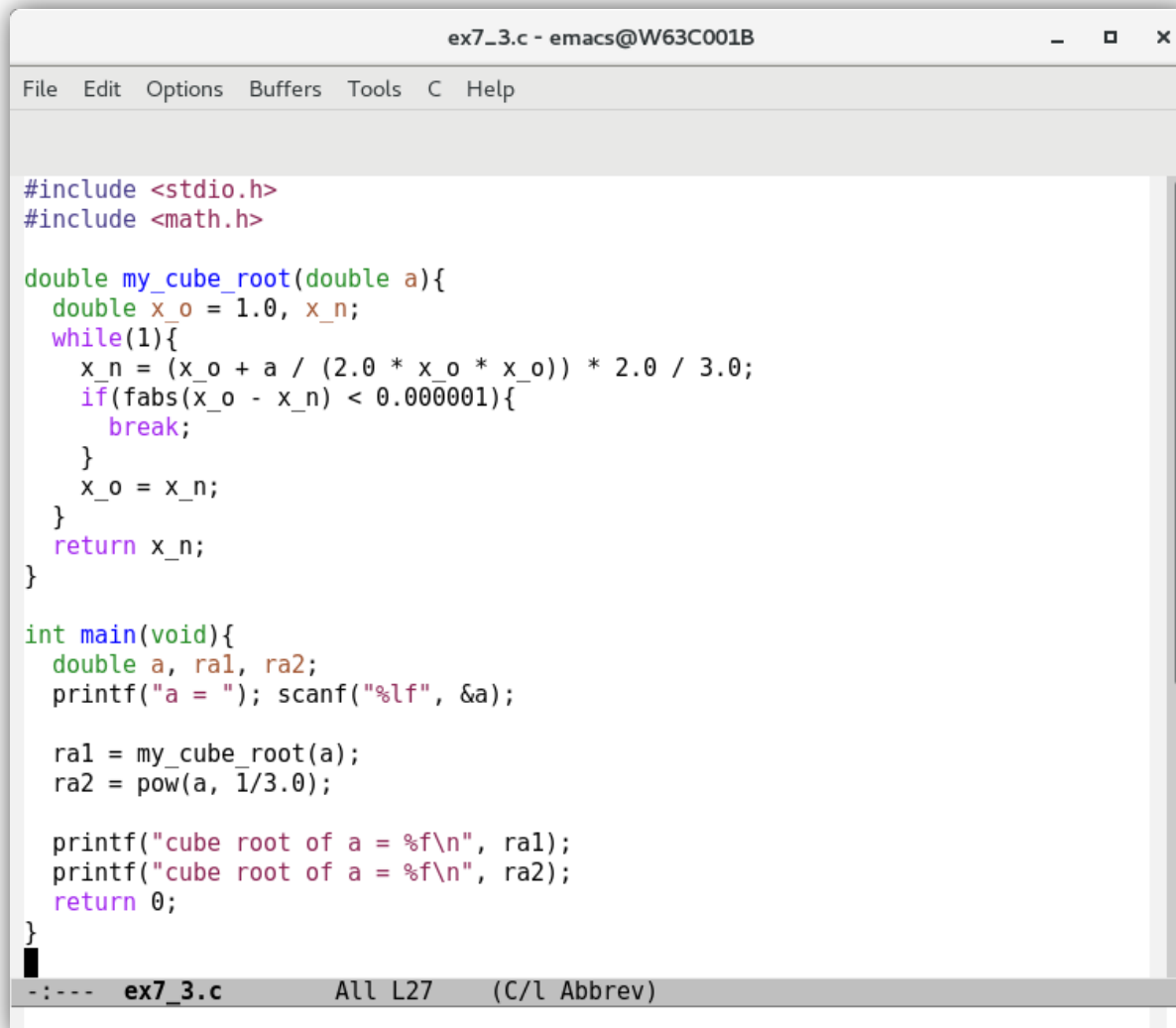
-:--- ex7_2.c      All L25      (C/l Abbrev)
```

演習⑦－ 3 （余力がある人向け）

- ▶ 端末から入力したdouble型変数 $a \geq 0$ の三乗根 $\sqrt[3]{a}$ を求める関数 `double my_cube_root(double a)` を自作し、`pow(a, 1/3.0)` の結果と比較せよ。
- ▶ ニュートン法について各自で調査し、演習⑦－ 2 と同じような漸化式を導出する。

??

演習⑦ – 3 ~プログラム例~



The screenshot shows an Emacs editor window titled "ex7_3.c - emacs@W63C001B". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The code is as follows:

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

double my_cube_root(double a){
    double x_o = 1.0, x_n;
    while(1){
        x_n = (x_o + a / (2.0 * x_o * x_o)) * 2.0 / 3.0;
        if(fabs(x_o - x_n) < 0.000001){
            break;
        }
        x_o = x_n;
    }
    return x_n;
}

int main(void){
    double a, ra1, ra2;
    printf("a = "); scanf("%lf", &a);

    ra1 = my_cube_root(a);
    ra2 = pow(a, 1/3.0);

    printf("cube root of a = %f\n", ra1);
    printf("cube root of a = %f\n", ra2);
    return 0;
}
```

The status bar at the bottom shows "ex7_3.c", "All L27", and "(C/l Abbrev)".