

Cプログラミング入門

(基幹5クラス)

第8回 配列・マクロ

本日の講義・演習項目

- ▶ 課題①の解説
- ▶ 配列・マクロ
- ▶ 授業内演習（第8回講義）

課題 1 – 1

▶ プログラムのポイント

▶ `sum : int型`, `avg : double型` → 平均値の計算方法

```
avg = (double)sum / 10;
```

```
avg = sum / 10.0;
```

▶ `sum : double型`, `avg : double型` → 合計値の表示

```
printf("Sum is %.0f¥n", sum);
```

▶ 実行結果

42, 36, 41, 30, 31

4.200000, 3.600000, 4.100000, 3.000000, 3.100000

課題 1 – 1

▶ 乱数について

▶ 設定

```
srand((unsigned)time(NULL));
```

- ▶ 生成 シード値 1970年1月1日0時0分0秒からの経過秒数

```
num = rand() % 6 + 1;
```

- ▶ プログラム中でrand()を複数回実行すると、シード値に応じた乱数列が生成される（実行環境によって異なる）
 - ▶ シード値1 : 1481765933, 1085377743, 1270216262, ...
 - ▶ シード値100 : 221617, 497652597, 241876805, ...
- ▶ 短時間に繰り返し実行した場合、実行結果にばらつきが生じない可能性がある

課題 1 – 2

▶ 実行結果

34952, 35111, 35103, 34794, 35300

3.495200, 3.511100, 3.510300, 3.479400, 3.530000

▶ 実行結果の比較・考察

- ▶ 課題 1 – 1 は平均値のばらつきが大きい
- ▶ 課題 1 – 2 は平均値のばらつきが小さい
- ▶ 出る目の確率がすべて同様に確からしいとすると、平均値の期待値は3.5
- ▶ 回数を10回から10,000回に増やすことで平均値が期待値に近づく結果となる（大数の法則）

課題 1 - 3

▶ プログラムのポイント

- ▶ if文の個数が多くなりすぎないように → 比較が多いほどわかりにくいプログラムに...

```
max = a; min = a;  
if(b > max){ max = b; }  
if(c > max){ max = c; }  
if(b < min){ min = b; }  
if(c < max){ min = c; }
```

4回の比較で十分！

▶ 実行結果

```
a = 13  
b = 8  
c = 24  
max = 24, min = 8
```

} 端末からの入力部分も
実行結果に含める

課題 1 - 4

▶ プログラムのポイント

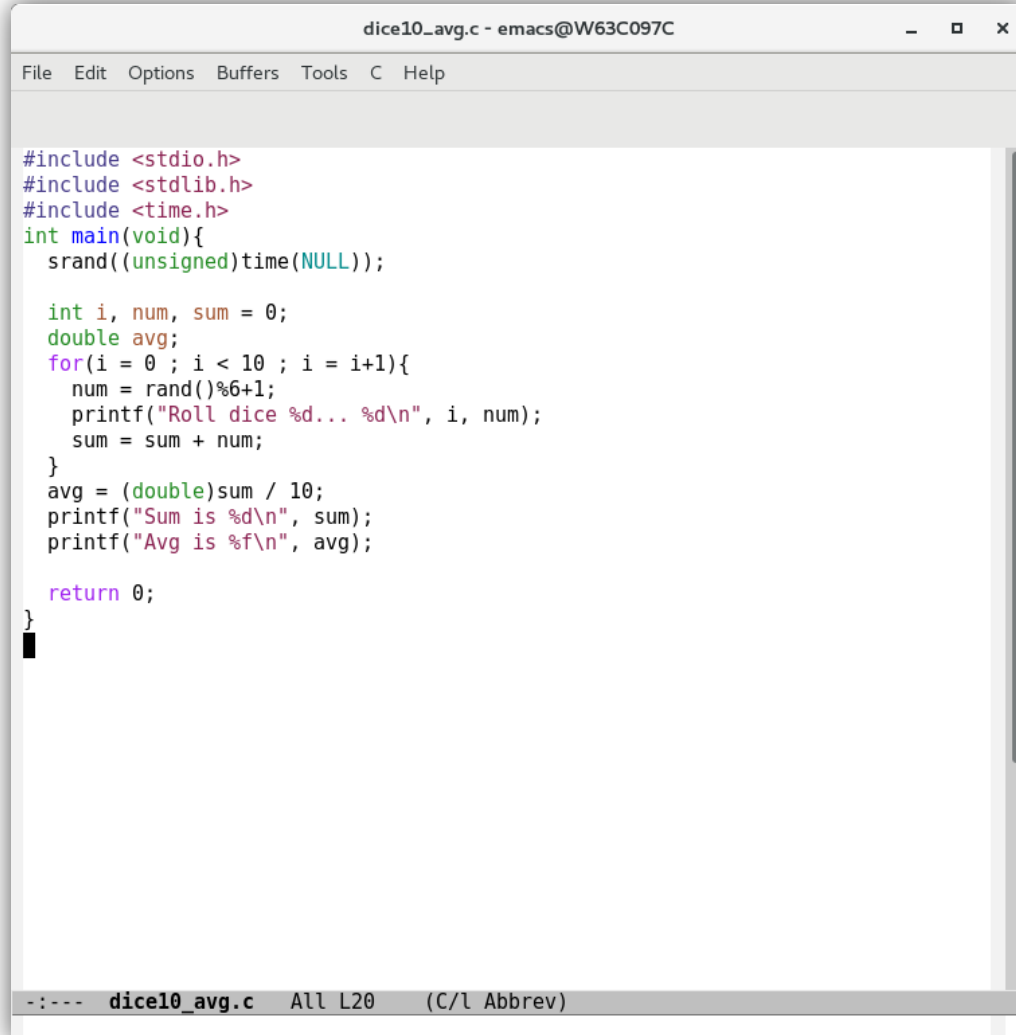
- ▶ for文中にif文を挿入
- ▶ カウントのために使う変数の初期化を忘れずに

```
int i, num, count = 0;
for(i = 0 ; i < 10000 ; i = i+1){
    num = rand()%6+1;
    if(num >= 5){ count = count + 1; }
}
```

▶ 実行結果

3336, 3249, 3376, 3330, 3352 ... 期待値3333に近い値

課題 1 – 1 のプログラム例



```
dice10_avg.c - emacs@W63C097C
File Edit Options Buffers Tools C Help

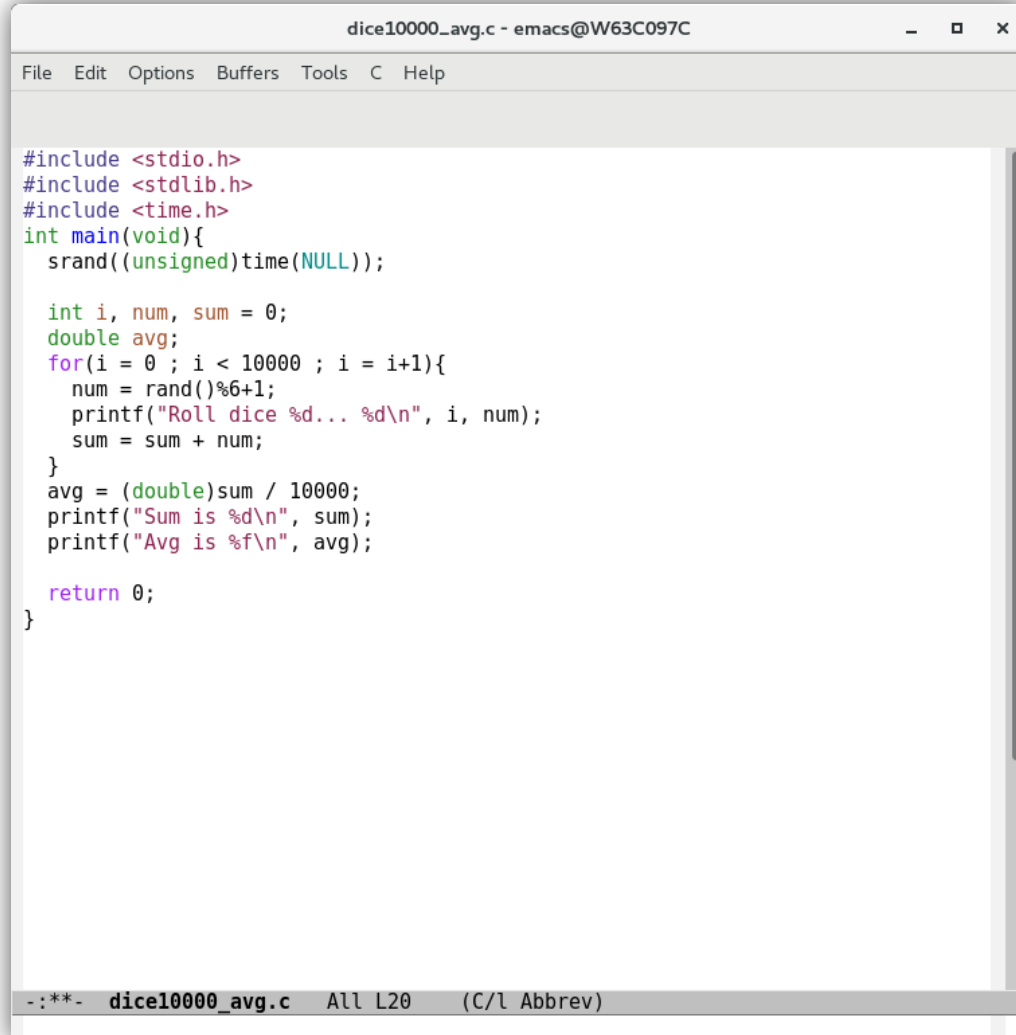
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(void){
    srand((unsigned)time(NULL));

    int i, num, sum = 0;
    double avg;
    for(i = 0 ; i < 10 ; i = i+1){
        num = rand()%6+1;
        printf("Roll dice %d... %d\n", i, num);
        sum = sum + num;
    }
    avg = (double)sum / 10;
    printf("Sum is %d\n", sum);
    printf("Avg is %f\n", avg);

    return 0;
}

-:--- dice10_avg.c  All L20  (C/l Abbrev)
```


課題 1 – 2 のプログラム例



```
dice10000_avg.c - emacs@W63C097C
File Edit Options Buffers Tools C Help

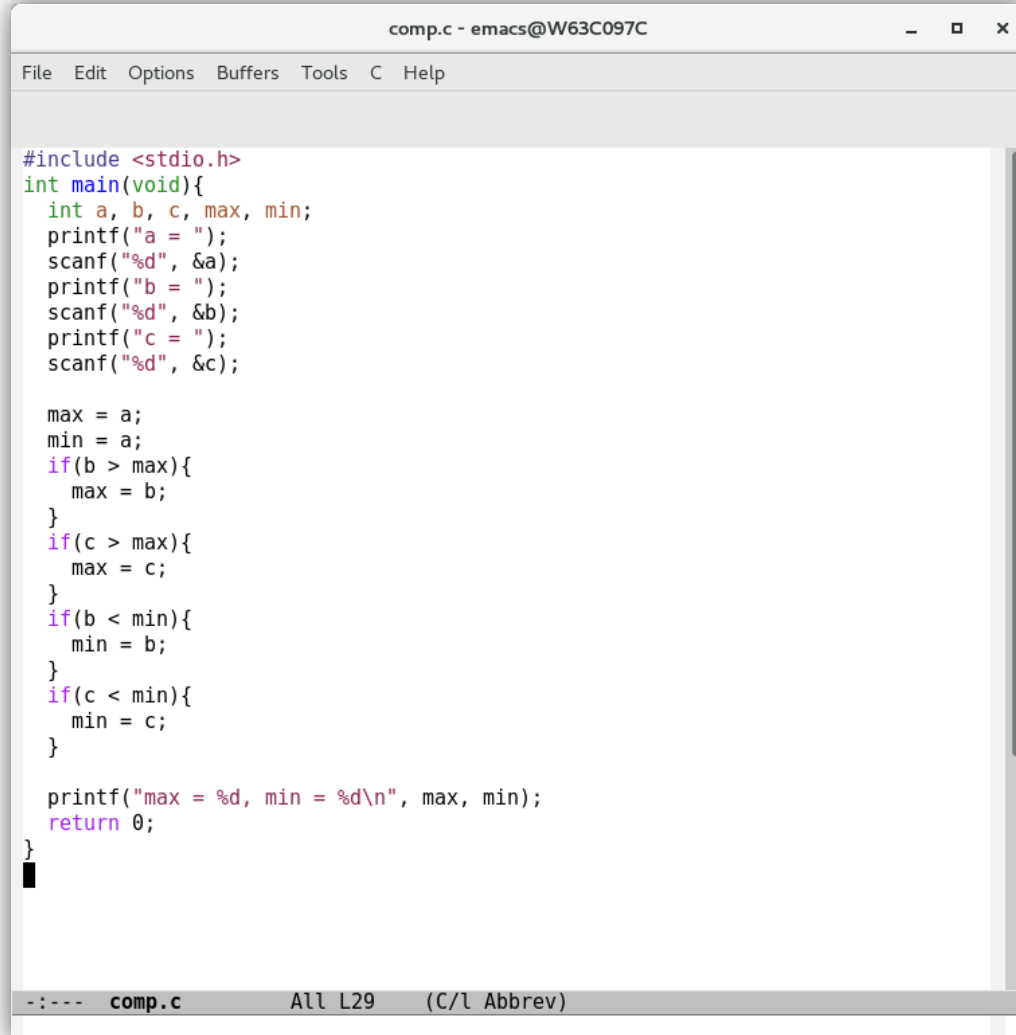
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(void){
    srand((unsigned)time(NULL));

    int i, num, sum = 0;
    double avg;
    for(i = 0 ; i < 10000 ; i = i+1){
        num = rand()%6+1;
        printf("Roll dice %d... %d\n", i, num);
        sum = sum + num;
    }
    avg = (double)sum / 10000;
    printf("Sum is %d\n", sum);
    printf("Avg is %f\n", avg);

    return 0;
}

-:**- dice10000_avg.c All L20 (C/l Abbrev)
```

課題 1 – 3 のプログラム例



```
comp.c - emacs@W63C097C
File Edit Options Buffers Tools C Help

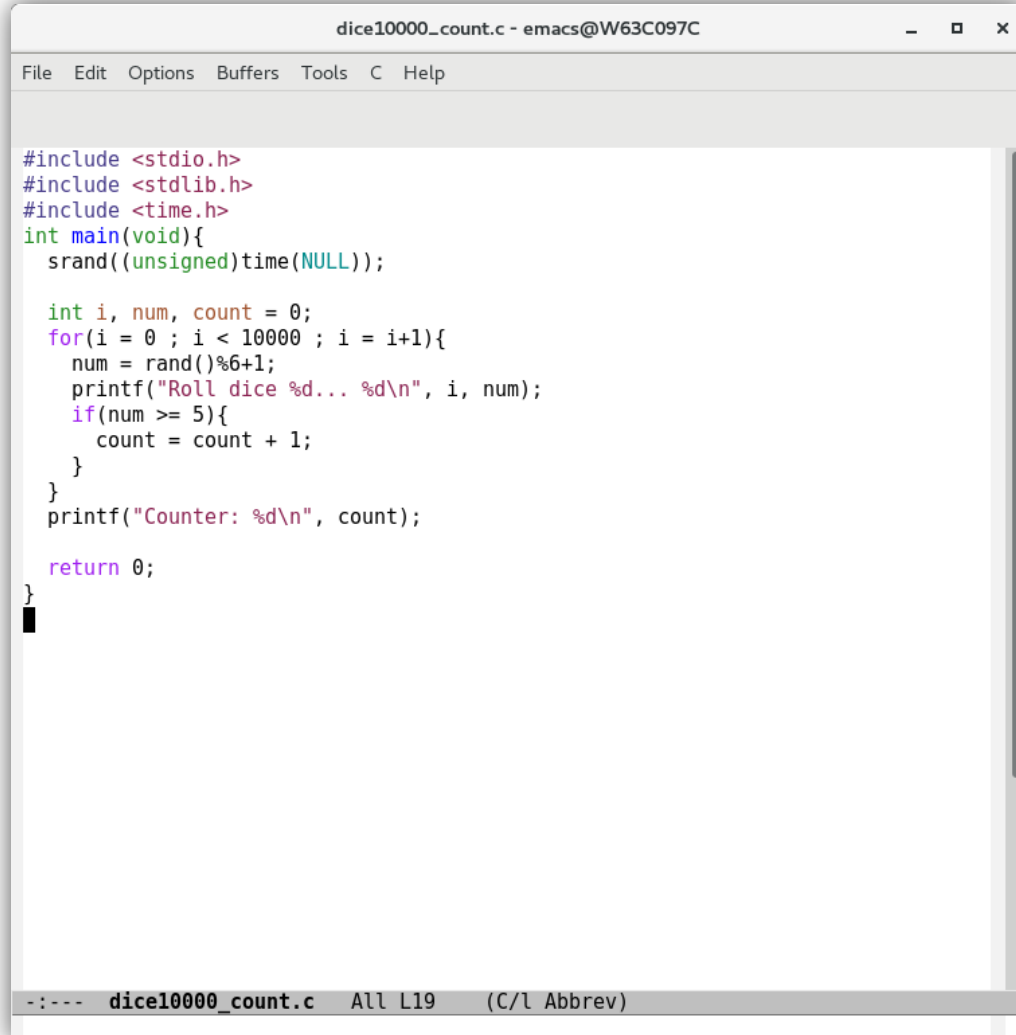
#include <stdio.h>
int main(void){
    int a, b, c, max, min;
    printf("a = ");
    scanf("%d", &a);
    printf("b = ");
    scanf("%d", &b);
    printf("c = ");
    scanf("%d", &c);

    max = a;
    min = a;
    if(b > max){
        max = b;
    }
    if(c > max){
        max = c;
    }
    if(b < min){
        min = b;
    }
    if(c < min){
        min = c;
    }

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

--- comp.c All L29 (C/l Abbrev)

課題 1 – 4 のプログラム例



```
dice10000_count.c - emacs@W63C097C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(void){
    srand((unsigned)time(NULL));

    int i, num, count = 0;
    for(i = 0 ; i < 10000 ; i = i+1){
        num = rand()%6+1;
        printf("Roll dice %d... %d\n", i, num);
        if(num >= 5){
            count = count + 1;
        }
    }
    printf("Counter: %d\n", count);

    return 0;
}

-:--- dice10000_count.c All L19 (C/l Abbrev)
```

本日の講義・演習項目

- ▶ 課題①の解説
- ▶ 配列・マクロ
- ▶ 授業内演習（第8回講義）

配列

▶ 配列の宣言

```
int i;           ... 変数の宣言  
int a[5];        ... 配列の宣言
```

変数

i



配列

a[0]

a[1]

a[2]

a[3]

a[4]



添え字は0~4
(1~5ではない)

▶ 配列の要素への代入

```
a[0] = 23;  
a[1] = 34;  
...
```

配列の何番目に代入するのかを
記述する必要がある

演習①

- ▶ 以下のプログラム **array1.c** を作成，実行

```
#include <stdio.h>
int main(void){
    int i, data[5], sum = 0;
    double avg;

    data[0] = 23; data[1] = 34; data[2] = 45;
    data[3] = 68; data[4] = 41;

    for(i = 0 ; i < 5 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 5;

    printf("sum = %d¥n", sum);
    printf("avg = %f¥n", avg);
    return 0;
}
```

} 代入

} 合計値と平均値
の計算

} 表示

演習① ~実行結果~



A terminal window titled "w536074@W63C060C:~/Cpro/08" with standard window controls. The menu bar includes "ファイル(F)", "編集(E)", "表示(V)", "検索(S)", "端末(T)", and "ヘルプ(H)". The terminal content shows the compilation and execution of a C program named "array1.c". The output of the program is "sum = 211" and "avg = 42.200000". The prompt "[w536074@W63C060C 08]\$" is visible at the end of each line.

```
w536074@W63C060C:~/Cpro/08
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[w536074@W63C060C 08]$ gcc array1.c -o array1
[w536074@W63C060C 08]$ ./array1
sum = 211
avg = 42.200000
[w536074@W63C060C 08]$
```

配列の初期化

▶ 配列の要素への代入

```
int a[5];  
a[0] = 23; a[1] = 34; a[2] = 45;  
a[3] = 68; a[4] = 41;
```

▶ 配列の初期化

```
int a[5] = {23, 34, 45, 68, 41};
```

同じ意味

▶ 配列の初期化は宣言時のみ可

▶ 以下の記述は許可されない

```
int a[5];  
a = {23, 34, 45, 68, 41};
```



演習②

- ▶ 以下のプログラム **array2.c** を作成，実行
 - ▶ 演習①と同様の結果が得られる

```
#include <stdio.h>
int main(void){
    int i, data[5] = {23,34,45,68,41}, sum = 0;
    double avg;

    for(i = 0 ; i < 5 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 5;

    printf("sum = %d¥n", sum);
    printf("avg = %f¥n", avg);
    return 0;
}
```



初期化

演習③

- ▶ 以下のプログラム **array3.c** を作成，実行

```
#include <stdio.h>
int main(void){
    int i, data[5], sum = 0;
    double avg;

    for(i = 0 ; i < 5 ; i++){
        printf("data[%d] = ", i);
        scanf( ? );
    }
    for(i = 0 ; i < 5 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 5;

    printf("sum = %d\n", sum);
    printf("avg = %f\n", avg);
    return 0;
}
```

... 標準入力から配列の要素に代入

実行結果（一部）

w536074@W63C060C:~/Cpro

ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

[w536074@W63C060C 08]\$ gcc array3.c -o array3

[w536074@W63C060C 08]\$./array3

data[0] = 23

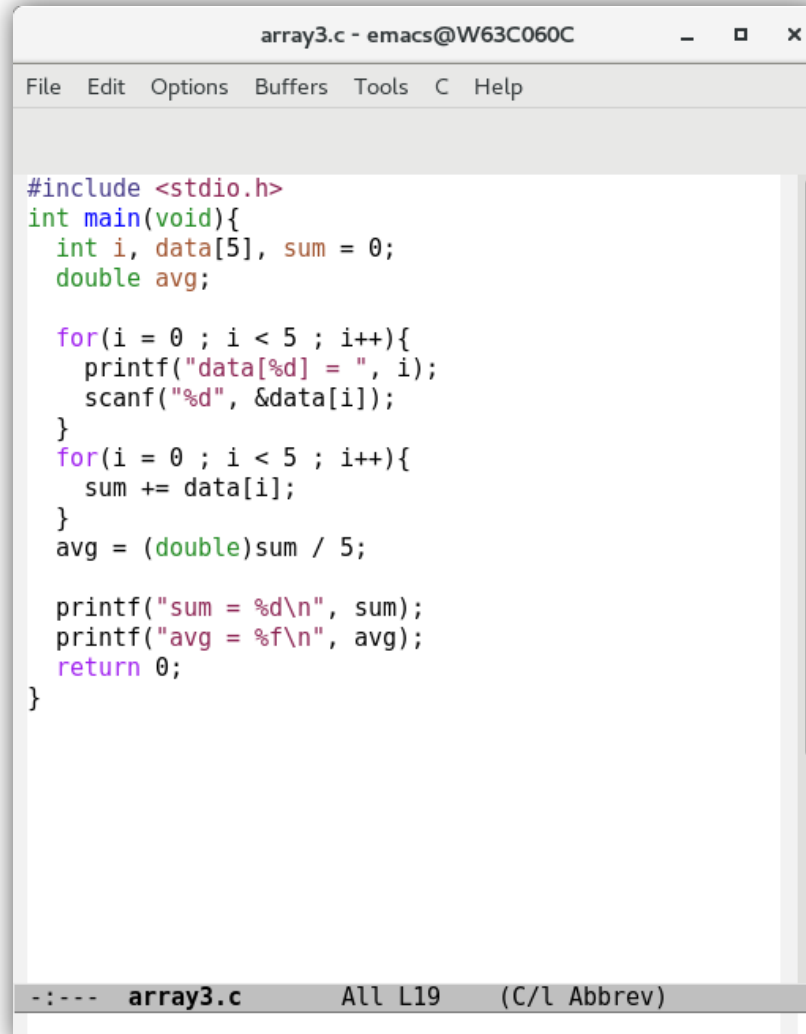
data[1] = 34

data[2] = 45

data[3] = █

} 端末から入力
できるように

演習③ ~解説~



```
array3.c - emacs@W63C060C
File Edit Options Buffers Tools C Help

#include <stdio.h>
int main(void){
    int i, data[5], sum = 0;
    double avg;

    for(i = 0 ; i < 5 ; i++){
        printf("data[%d] = ", i);
        scanf("%d", &data[i]);
    }
    for(i = 0 ; i < 5 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 5;

    printf("sum = %d\n", sum);
    printf("avg = %f\n", avg);
    return 0;
}

-:--- array3.c      All L19      (C/l Abbrev)
```

演習③でdataの要素数を5個から
10個に増やしたい

```
#include <stdio.h>
int main(void){
    int i, data[5], sum = 0;
    double avg;

    for(i = 0 ; i < 5 ; i++){
        printf("data[%d] = ", i);
        scanf();
    }
    for(i = 0 ; i < 5 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 5;

    printf("sum = %d¥n", sum);
    printf("avg = %f¥n", avg);
    return 0;
}
```

変更箇所は？

... 標準入力から配列の要素に代入

演習③のプログラム
array3.c

```
#include <stdio.h>
int main(void){
    int i, data[10], sum = 0;
    double avg;

    for(i = 0 ; i < 10 ; i++){
        printf("data[%d] = ", i);
        scanf();
    }
    for(i = 0 ; i < 10 ; i++){
        sum += data[i];
    }
    avg = (double)sum / 10;

    printf("sum = %d¥n", sum);
    printf("avg = %f¥n", avg);
    return 0;
}
```

変更箇所が多い

... 標準入力から配列の要素に代入

演習③のプログラム
array3.c

マクロ定義

#define

- ▶ 使用例

```
#define SIZE 5
```

- ▶ **SIZE**と書いた箇所をすべて**5**として扱う
- ▶ 通常, マクロ定義はmainよりも前

演習④

- ▶ 以下のプログラム **array4.c** を作成，実行

```
#include <stdio.h>
#define SIZE 5
int main(void){
    int i, data[SIZE], sum = 0;
    double avg;

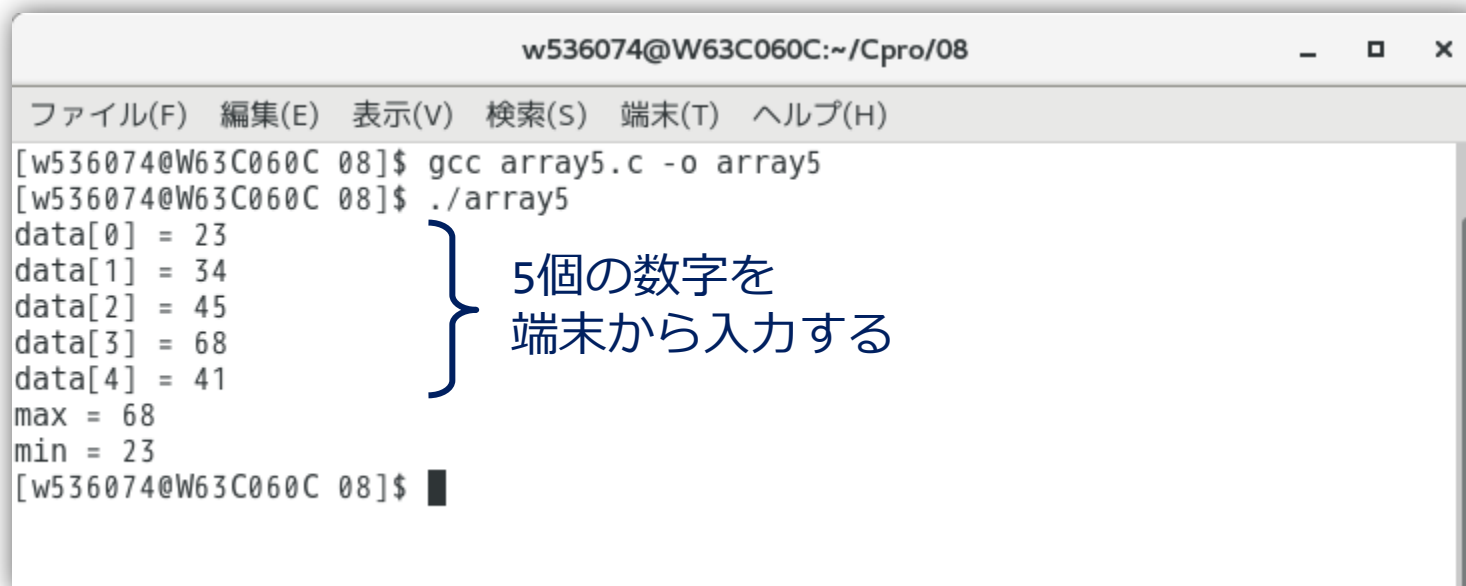
    for(i = 0 ; i < SIZE ; i++){
        printf("data[%d] = ", i);
        scanf("%d", &data[i]);
    }
    for(i = 0 ; i < SIZE ; i++){
        sum += data[i];
    }
    avg = (double)sum / SIZE;

    printf("sum = %d\n", sum);
    printf("avg = %f\n", avg);
    return 0;
}
```

← SIZEの値を変更した場合も
実行してみる (5 → 10)

演習⑤

- ▶ 演習④のプログラム (**array4.c**) を改変し, 最大値と最小値を出力するプログラム **array5.c** を作成せよ。
 - ▶ 入力データ数 (SIZE) は5個とする
- ▶ 実行例 :



```
w536074@W63C060C:~/Cpro/08
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[w536074@W63C060C 08]$ gcc array5.c -o array5
[w536074@W63C060C 08]$ ./array5
data[0] = 23
data[1] = 34
data[2] = 45
data[3] = 68
data[4] = 41
max = 68
min = 23
[w536074@W63C060C 08]$
```

5個の数字を
端末から入力する

演習⑤ ~解説~



```
array5.c - emacs@W63C060C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#define SIZE 5
int main(void){
    int i, data[SIZE], max, min;

    for(i = 0 ; i < SIZE ; i++){
        printf("data[%d] = ", i);
        scanf("%d", &data[i]);
    }

    max = data[0];
    min = data[0];

    for(i = 1 ; i < SIZE ; i++){
        if(data[i] > max){ max = data[i]; }
        if(data[i] < min){ min = data[i]; }
    }

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

--- array5.c All L23 (C/l Abbrev)

演習⑥（漸化式）

$$a_{n+1} = 2a_n + 3, \quad a_1 = 1$$



$$a_{10} = ??$$

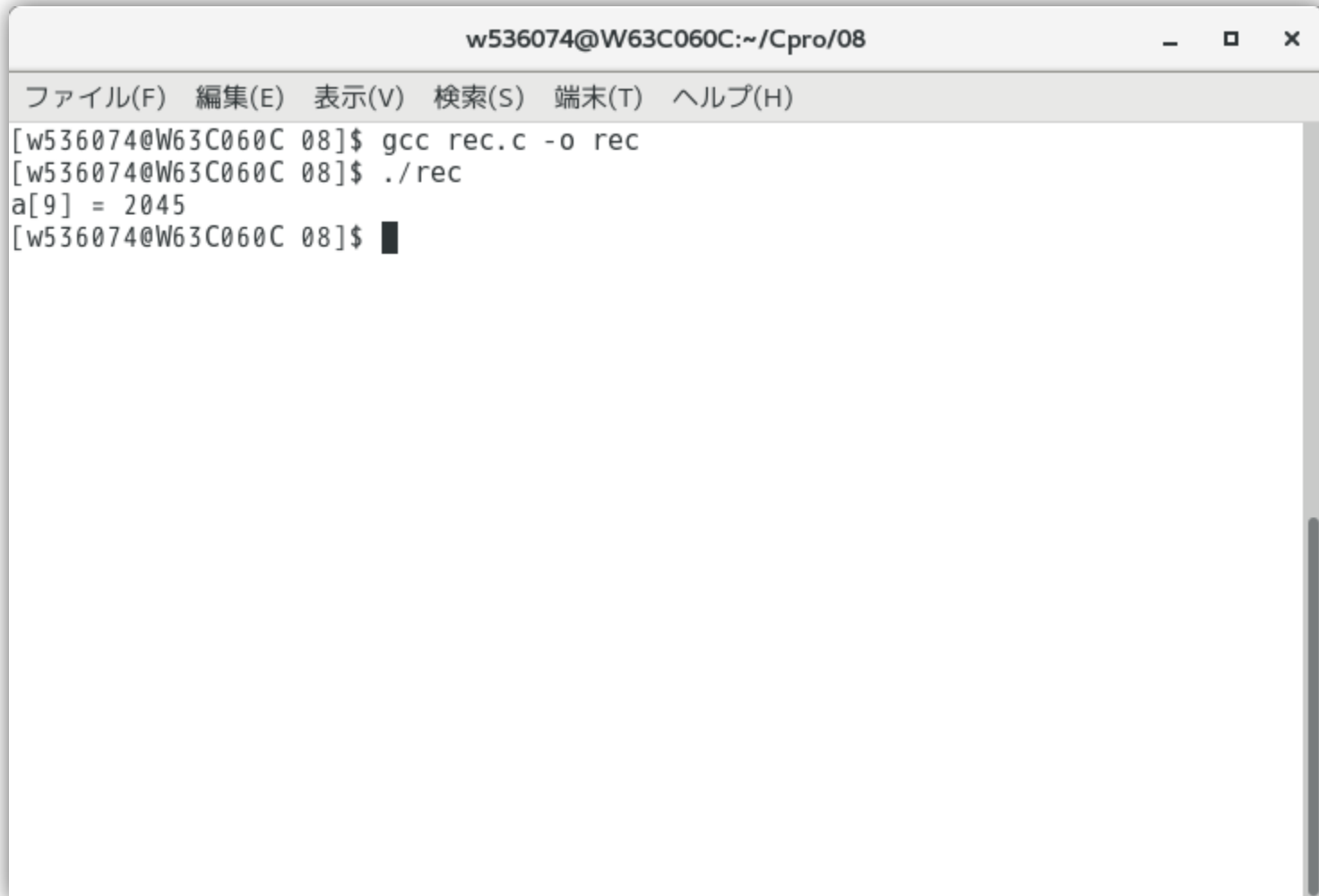
- ▶ 以下のプログラム **rec.c** を作成，実行

```
#include <stdio.h>
#define SIZE 10
int main(void){
    int i, a[SIZE];

    a[0] = 1;
    for(i = 0 ; i < SIZE-1 ; i++){
        a[i+1] = 2 * a[i] + 3;
    }

    printf("a[%d] = %d\n", SIZE-1, a[SIZE-1]);
    return 0;
}
```

演習⑥ ~実行結果~



```
w536074@W63C060C:~/Cpro/08
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[w536074@W63C060C 08]$ gcc rec.c -o rec
[w536074@W63C060C 08]$ ./rec
a[9] = 2045
[w536074@W63C060C 08]$
```

The image shows a terminal window with a title bar containing the username and host 'w536074@W63C060C:~/Cpro/08' and standard window controls. The terminal has a menu bar with options: ファイル(F), 編集(E), 表示(V), 検索(S), 端末(T), ヘルプ(H). The command history shows the compilation of 'rec.c' into 'rec' using 'gcc', followed by the execution of './rec'. The output of the program is 'a[9] = 2045'. The prompt '[w536074@W63C060C 08]\$' is followed by a black cursor block.

演習⑦（漸化式）

$$a_{n+2} = a_{n+1} + a_n, \quad a_1 = 1, \quad a_2 = 1$$



$$a_{20} = ??$$

- ▶ 演習⑥を参考に，フィボナッチ数列の第20項を求めるプログラム **fib.c** を作成しなさい。
 - ▶ フィボナッチ数列の漸化式は上の通り
 - ▶ 第20項は**a[19]**に記憶されるはず（**a[20]**ではない）
 - ▶ プログラムには必ずマクロ定義（**SIZE**）を含める

演習⑦ ~解説~

```
fib.c - emacs@W63C060C
File Edit Options Buffers Tools C Help

#include <stdio.h>
#define SIZE 20
int main(void){
    int i, a[SIZE];

    a[0] = 1;
    a[1] = 1;
    for(i = 0 ; i < SIZE-2 ; i++){
        a[i+2] = a[i+1] + a[i];
    }

    printf("a[%d] = %d\n", SIZE-1, a[SIZE-1]);
    return 0;
}
```

ループ回数に
注意！

プログラム例

fib.c All L15 (C/l Abbrev)

```
w536074@W63C060C:~/Cpro/08
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

[w536074@W63C060C 08]$ gcc fib.c -o fib
[w536074@W63C060C 08]$ ./fib
a[19] = 6765
[w536074@W63C060C 08]$ gcc fib.c -o fib
[w536074@W63C060C 08]$ ./fib
a[29] = 832040
[w536074@W63C060C 08]$
```

実行結果
(SIZE = 20 or 30)

演習⑦ ~解説~

▶ 配列を使う場合の注意点

`int a[20];`と宣言した場合、使用できる配列の要素は`a[0]~a[19]`

頻出の間違い①

```
for(i = 0 ; i < SIZE-1 ; i++){  
    a[i+2] = a[i+1] + a[i];  
}
```

- `SIZE=20`であるとする、`i`は0から18の範囲を動く
- `i=18`のとき、`a[20]`への代入が行われる

頻出の間違い②

```
for(i = 0 ; i < SIZE-1 ; i++){  
    a[i+1] = a[i] + a[i-1];  
}
```

- `i=0`のとき、`a[-1]`を使った計算が行われる

本日の講義・演習項目

- ▶ 課題①の解説
- ▶ 配列・マクロ
- ▶ 授業内演習（第8回講義）

授業内演習

- ▶ 講義開始時にCourse N@viにアップロードします

キーワード，次回の講義

- ▶ 本日のキーワード：
- ▶ 次回は6/14
- ▶ 次回講義までに予習ビデオ「第9回 乱数・数学ライブラリ」「第10回 関数・グローバル変数」を視聴し，各自プログラミング実習