

# 第 8 回情報工学演習課題

AJG23055 牧野唯希

2024 年 6 月 6 日

## 1 課題 1

2つの double 型変数の四則演算を行う関数の定義を 1 つにまとめたファイル, プロトタイプ宣言をまとめたファイル, これらの関数を用いた main 関数があるファイルをそれぞれ作成し, 分割コンパイルをして, プログラムを実行し, 挙動を確認せよ.

加算, 減算, 乗算, 除算の 4 つの関数を作る

関数の double 型仮引数を 2 つとし, 戻り値を計算結果とする

```
~/section0606$ gcc -c kadai8-AJG23055-1.c
~/section0606$ gcc -c kadai8-AJG23055-1-main.c
kadai8-AJG23055-1-main.c: In function 'main':
kadai8-AJG23055-1-main.c:10:3: warning: ignoring return value of 'scanf'
declared with attribute 'warn_unused_result' [-Wunused-result]
   10 |     scanf("%lf", &a);
      |     ^~~~~~
kadai8-AJG23055-1-main.c:12:3: warning: ignoring return value of 'scanf'
declared with attribute 'warn_unused_result' [-Wunused-result]
   12 |     scanf("%lf", &b);
      |     ^~~~~~
~/section0606$ gcc -c kadai8-AJG23055-1.h
~/section0606$ gcc -o kadai8-AJG23055-1.exe kadai8-AJG23055-1-main.o kadai8-AJG23055-1.o
~/section0606$ ./kadai8-AJG23055-1.exe
a=5
b=4
a+b=9.000000
a-b=1.000000
a*b=20.000000
a/b=1.250000
```

図 1 課題 1 出力結果

## 2 課題 2

キーボードから配列の要素数を入力し、malloc 関数でメモリを確保することで、ポインタを用いて int 型配列を実現せよ。また、配列に 0 から 99 までの乱数を代入し、これらの最大値、最小値、平均値、分散、中央値を求め、コマンドラインに出力せよ。

```
~/section0606$ gcc -o kadai8-AJG23055-2.exe kadai8-AJG23055-2.c
kadai8-AJG23055-2.c: In function 'main':
kadai8-AJG23055-2.c:13:3: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
   13 |     scanf("%d", &n);
      |     ^~~~~~
~/section0606$ ./kadai8-AJG23055-2.exe
Enter the number of elements: 10
Maximum: 81
Minimum: 2
Average: 52.00
Variance: 727.60
Median: 58.50
~/section0606$ ./kadai8-AJG23055-2.exe
Enter the number of elements: 1000
Maximum: 99
Minimum: 0
Average: 48.09
Variance: 832.20
Median: 49.00
~/section0606$ ./kadai8-AJG23055-2.exe
Enter the number of elements: 5
Maximum: 86
Minimum: 37
Average: 59.40
Variance: 311.84
Median: 56.00
```

図 2 課題 2 出力結果

## 3 課題 3

授業支援システムにある kadai8-3.c は、関数 replace で 2 つの引数 a,b の値を入れ替えようとしている。しかし、このままでは値を入れ替えることは出来ない。プログラムの一部を修正して、値を入れ替え可能にせよ。また、レポートに値を入れ替えられるようになった理由を書け。

修正したプログラムを提出すること

ファイルの名前を提出用に変更しておいて下さい

```
~/section0606$ gcc -o kadai8_3.exe kadai8_3.c
~/section0606$ ./kadai8_3.exe
a=10, b=6
a=10, b=6
~/section0606$ gcc -o kadai8-AJG23055-3.exe kadai8-AJG23055-3.c
```

図 3 課題 3-1 kadai8<sub>3</sub>.c出力結果

第3図のように、数字の入れ替わりが行われていないことが分かる。そこで、プログラムを修正し、再度コンパイルしたものを第4図に示す。

```
~/section0606$ gcc -o kadai8-AJG23055-3.exe kadai8-AJG23055-3.c
~/section0606$ ./kadai8-AJG23055-3.exe
a=10, b=6
a=6, b=10
```

図4 課題3-2 出力結果

こちらのプログラムでは、正しく数字が入れ替わっていることが分かる。

## 4 課題4

授業支援システムにある kadai8-4.c というコードを実行すると、実行結果は以下の通りとなった。

```
~/section0606$ gcc -o kadai8_4.exe kadai8_4.c
~/section0606$ ./kadai8_4.exe
pt_c: 0x7ffdf23b6e33, 55
pt_i: 0x7ffdf23b6e34, 1234
pt_c2: 0x7ffdf23b6e34, -46
```

図5 課題4-1 出力結果（コンパイル時）

コードを見ると、pt\_c2 は pt\_i を代入したものである。しかし、実行結果から、アドレスの値は同じになったものの、数値が変わって出力された。これはどうしてか。レポート内で理由を述べよ。

pt\_c: 0x7ffdf23b6e33, 55

ここで、pt\_c は mychar のアドレスを指しており、その値は 55

pt\_i: 0x7ffdf23b6e34, 1234

ここで、pt\_i は mydt のアドレスを指しており、その値は 1234

pt\_c2: 0x7ffdf23b6e34, -46

ここで、pt\_c2 は pt\_i をキャストしたもので、mydt のアドレスを指している。しかし、値が 1234 ではなく -46 と表示されていた。

pt\_c2 は pt\_i のアドレスを指しているので、アドレスは同じとなる。

pt\_i は int \*型のポインタで、pt\_i を通じてその値を読み取るとき、1234 という int 型の値が読み取られる。しかし、pt\_c2 は char \*型のポインタにキャストされている。このため、int の最初の 1 バイトが char 型として読み取られている。1234 という整数のメモリ表現は検索した結果、0xD2 0x04 0x00 0x00 であった。

pt\_c2 が指している最初のバイトは 0xD2 で、これも調べた結果 -46 であった。

そのため、pt\_c2 の値が -46 になったのだと考えられる。

## 5 授業の感想

今回の授業課題は前回に比べると少し容易であった。しかし、C言語の難関分野の1つでもある、ポインタががっつりと関わってきたりしたので、手ごたえはあった。また、新しく複数のファイルからのコンパイルなどを学んだ。課題は今までとは異なり、なぜこのような結果が得られるのかを考察する形式だったが、自分なりに調べながら考察することが出来た。