

第9回 計算機基礎実習II

座席指定はありません。

計算機基礎実習II 2018 のウェブページから、以下の課題に自力で取り組んで下さい。

第7回の復習課題(rev07)

第9回の基本課題(base09)

第8回試験の結果

中間試験に関するコメント

コンパイルできない不完全なプログラムなど

→ プログラミングに慣れていない、あるいは複雑な問題は、要件をバラして段階的にプログラムを作成する。

exam08-2.c

3つの実数 x, y, z の値を順に入力すると、 $(x+y)z$ と $\sqrt{(x^2+\dots)}$ の値を出力することを繰り返すプログラム…。ただし、 x, y, z の値にすべて 0 が入力されたときはプログラムを終了するものとする。

- x, y, z を入力して x, y, z の値を出力するプログラム
- 1 の入力と出力を繰り返すプログラム
- x, y, z を入力して $(x+y)z$ の値を出力するプログラム
- 3 に加えて、 $x^2+y^2+z^2$ の値を出力するプログラム
- 3 に加えて、 $\sqrt{(x^2+y^2+z^2)}$ の値を出力するプログラム
- 4 に加えて、 x に 0 が入力されると終了するプログラム
- 4 に加えて、 x, y, z 全てが 0 のとき終了するプログラム

中間試験に関するコメント

複数条件の組み合わせ

$\text{if}(a < x < b) \rightarrow \text{if}(a < x \ \&\& \ x < b)$

→ C言語の比較演算子は「二項演算子」です。

$a < b$ が成立すれば 1、成立しなければ 0 を返すような演算子 $<$

したがって、 $\text{if}(a < x < b)$ のような式は、 $a=1, x=-1, b=3$ とすると

- $(a < x)$ は $1 < -1$ を満たさないので $(a < x) \rightarrow 0$
- $(a < x < b)$ は 1 より $(0 < b)$ となり、 $0 < 3$ なので
- $(a < x < b) \rightarrow (0 < b) \rightarrow (0 < 3) \rightarrow 1$
- $(a < x < b)$ は $1 < -1 < 3$ が満たされていないので偽、
- しかし、プログラムでは $(a < x < b) \rightarrow 1$ となり真

中間試験に関するコメント

プログラミングが書けない人ほどインデント(字下げ)しない
→ プログラムコードは人間のためのものです

```
#include <stdio.h>

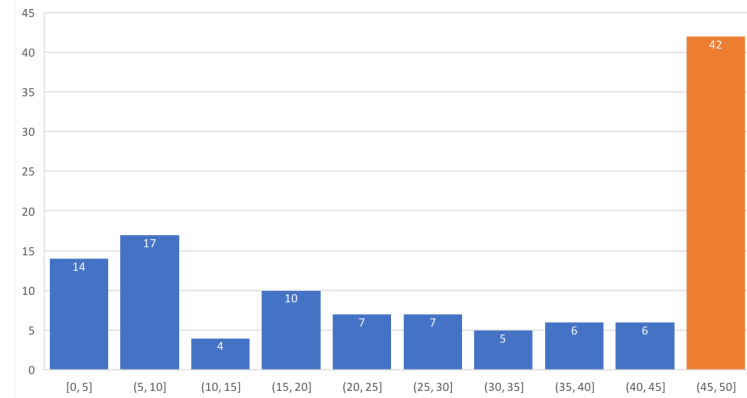
int main() {
    int a;
    scanf("%d", &a);

    return a/2;
}
```

```
#include<stdio.h> int main(){int a;scanf("%d",&a);return a/2;}
```

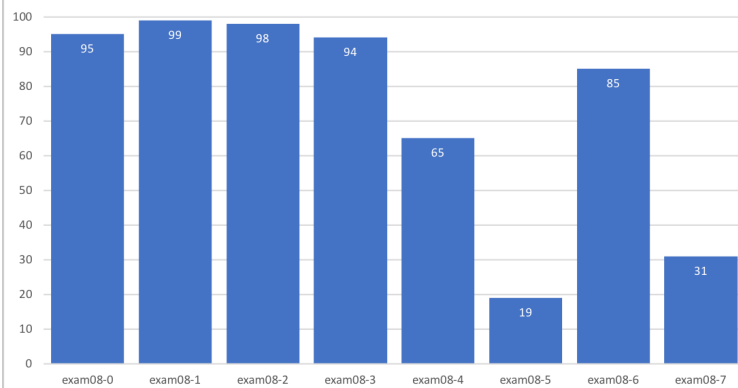
```
0016140 0000 0000 0000 0000 0099 0000 0001 0012
0016160 0e10 00e0 0000 0000 0000 0000 0000 0000
0016200 00b8 0000 0004 fff1 0000 0000 0000 0000
0016220 0000 0000 0000 0000 0001 0000 0004 fff1
0016240 0000 0000 0000 0000 0000 0000 0000 0000
0016260 00c3 0000 0001 0011 0740 0040 0000 0000
0016300 0000 0000 0000 0000 00d1 0000 0001 0014
0016320 0e20 00e0 0000 0000 0000 0000 0000 0000
0016340 0000 0000 0004 fff1 0000 0000 0000 0000
0016360 0000 0000 0000 0000 00dd 0000 0000 0012
:
```

kiso2-20181115 Total inc. abs=10



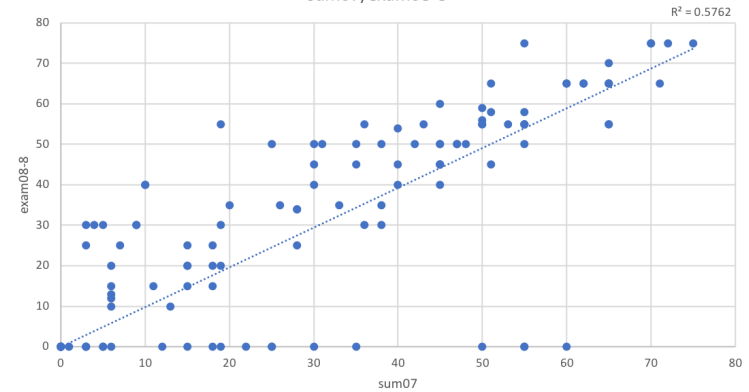
評価の分布

kiso2-20181115 Submit



解答数の分布

sum07/exam08-8



自己採点と素点の分布

第8回試験の解答例

exam08-1.c

```
#include <stdio.h>

int main() {
    int a;
    scanf("%d", &a);

    return a/2;
}
```

exam08-2.c

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

int main() {
    double x, y, z;

    while(1) {
        printf("x y z? ");
        scanf("%lf%lf%lf", &x, &y, &z);

        if (x==0 && y==0 && z==0) break;

        printf("%f %f\n", (x+y)*z, sqrt(x*x+y*y+z*z));
    }
    return 0;
}
```

exam08-3.c

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

int main() {
    int a, b;
    double x;

    //printf("範囲 a b? ");
    scanf("%d%d", &a, &b);

    if (a > b) exit(1);

    while(1) {
        //printf("x? ");
        scanf("%lf", &x);

        if (x < a || b < x)
            printf("範囲外です。\\n");
        else
            printf("範囲に含まれています。\\n");

        if (x == a || x == b) {
            printf("端点です。終了します。\\n");
            break;
        }
    }

    return 0;
}
```

exam08-4.c

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

int main() {
    char c;
    //printf("c? ");
    scanf("%c", &c);

    if (c<65 || 90<c) exit(1);

    int i;
    for (i=c; i<=90; i++)
        printf("%c ", i);

    printf("\n");
    return 0;
}
```

exam08-5.c

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

int main() {
    double x;
    double tr = M_PI/(2*sqrt(3));
    double sum = 0;

    scanf("%lf", &x);

    int k=0;
    while (1) {
        sum += pow(-1, k) / ((2*k+1) * pow(3, k));
        printf("n=%2d: %.10f\n", k, sum);
        if (fabs(sum-tr) < x) break;
        k++;
    }
    return 0;
}
```

exam08-6.c

```
#include <stdio.h>

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

    int i;
    for (i=a; i<=b; i++)
        if (i%2 == 1 || i%2 == -1)
            printf("%d ", i);
    printf("\n");
    return 0;
}
```

exam08-7.c

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

int main() {
    double x;
    double ep = DBL_MAX;
    double sum = 0;
    int n = 0;

    while (1) {
        //printf("x? ");
        scanf("%lf", &x);

        n++;
        sum += x;
        if (fabs(x) < fabs(ep)) ep = x;

        printf("n = %d, emin = %f\n", n, ep);
        if (ep == 0.0) break;
    }

    return 0;
}
```

第9回の内容

配列変数の宣言

配列変数の参照・代入

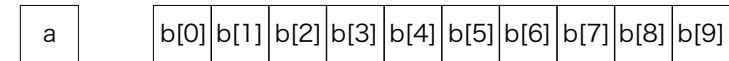
配列変数の初期化

配列変数とは

```
int main() {
```

```
    int a;           a という名前の整数型の変数を1つ用意
```

```
    int b[10];      b[0], b[1], ..., b[9] という名前の整数型の  
                    : 変数を合計10こ用意
```



b という整数型の変数は存在しない。b[0] や b[9] が一つの変数。

b[n] と宣言すると、b[0] から b[n-1] までが使える。b[n] は使えない。

配列変数の [] の中の何番目かを表す0以上の整数を「添字」と呼ぶ。

配列変数の何がうれしいのか

例えば、**1000個の変数**が必要なときに、

```
int a0, a1, a2, a3, ..., a998, a999;
```

としなくても、

```
int a[1000];
```

こうして宣言できる。

また、for 文などを使って、こんなふうに見える。

```
for (i=0; i<1000; i++) {  
    a[i] = i;  
}
```

配列変数が無ければ、a0=0; a1=1; a2=2; a3=3; ... a999=999;

配列変数の参照・代入

添字が付く以外は、通常の変数と同じ

```
int a[10];
```

```
a[0] = 0;  
a[1] = 1;  
a[2] = a[0] + a[1];
```

ただし、添字を整数で表現できるので

```
double x[10];  
int i = 0;
```

```
x[i] = 0.1;           // x[0] = 0.1  
i++;                 // i = 1  
x[i] = 0.2;          // x[1] = 0.2  
x[i*3] = x[i] * x[0]; // x[3] = x[1] * x[0]
```

ex09-1.c

整数型の配列変数 `a` を大きさ(要素の数)10で宣言し、さらに、この配列変数 `a` の `i` 番目の要素の値を `i*2` で初期化しなさい。このように宣言・初期化された配列変数 `a` の全ての要素(添字)番号 `i` と、`a` の `i` 番目の要素の値 `a[i]` を順に出力する

```
#include <stdio.h>

int main() {
    int i;
    int a[10];

    for (i=0; i<10; i++)
        a[i] = i*2;

    for (i=0; i<10; i++)
        printf("a[%d]=%d\n", i, a[i]);

    return 0;
}
```

配列変数の初期化

基本的には全ての要素に逐一代入する

```
for (i=0; i<10; i++)
    a[i] = i*2;
```

宣言時に定数で初期化することもできる

```
int a[10] = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18};
```

配列変数の初期化

宣言時に初期化することもできる...が、

```
int a[10] = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18};
```

もし、要素数より多くの値で初期化すると

```
int a[10] = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
```

コンパイル時に警告(Warning)される(エラーではない)

もし、要素数より少ない値で初期化すると

```
int a[10] = {0, 2, 4, 6};
```

以降の要素は全て 0 で初期化される。

宣言時の要素数を省略すると、初期化値の数でサイズが決まる

```
int a[] = {0, 2, 4, 6, 8};
// int a[5] = {0, 2, 4, 6, 8};
```

ex09-2.c

大きさ(要素の数)が10の実数型の配列変数 `x` に対して、各要素 `x[0]` から `x[9]` までの値を順にキーボードから入力すると、入力後に全ての `x` の要素の値とその総和を出力する

```
#include <stdio.h>

int main() {
    int i;
    double x[10];
    double sum = 0.0;

    for (i=0; i<10; i++) {
        printf("%d? ", i);
        scanf("%lf", &x[i]);
        sum += x[i];
    }

    for (i=0; i<10; i++)
        printf("x[%d] = %f\n", i, x[i]);
    printf("sum = %lf\n", sum);

    return 0;
}
```