

計算機プログラミング

第12回目

July 1, 2021

(10:30~12:00)

担当教員: Thi Thi Zin (ティティズイン)
<thithi@cc.miyazaki-u.ac.jp>



＜これまで学んだ内容について復習＞

過去問（2014年度より）

問題1

3行3列の行列 a と 3行1列の行列 x の掛け算を行うプログラムを作成せよ。

$$b = ax = \begin{bmatrix} 1.2 & 2.3 & 3.4 \\ 5.8 & 2.1 & -4.9 \\ 3.2 & 6.4 & 5.2 \end{bmatrix} \begin{bmatrix} 3.5 \\ 2.1 \\ 8.8 \end{bmatrix}$$

穴埋め
問題

穴埋めした後の
プログラム

```
#include <stdio.h>
int main(void){
    double    a[3][3]={ {1.2, 2.3, 3.4}, {5.8,2.1,-4.9},{3.2,6.4,5.2} };
    double    x[3]= (a)
    (b)
    (c)

    for ( (d) ) {
        b[i] = 0.0;
        for ( j = 0 ; j < 3 ; j++ ) {
            b[i]+=(e)
        }
        printf(" b [ %d ] = %f¥n", i, b[ i ] );
    }
    return 0;
}
```

```
#include <stdio.h>
int main(void){
    double    a[3][3]={ {1.2, 2.3, 3.4}, {5.8,2.1,-4.9},
                        {3.2,6.4,5.2} };
    double    x[3]={3.5, 2.1, 8.8};
    double    b[3];
    int       i, j ;

    for ( i = 0 ; i < 3 ; i ++ ) {
        b[i] = 0.0;
        for ( j = 0 ; j < 3 ; j++ ) {
            b[i]+=a[i][j]*x[j];
        }
        printf(" b [ %d ] = %f¥n", i, b[ i ] );
    }
    return 0;
}
```

問題1 解答例

(a) { 3.5, 2.1, 8.8 } ;

(b) double b[3];

(c) int i, j;

(d) i = 0 ; i < 3 ; i++

(e) a[i][j] * x[j] ;

過去問（2014年度より）

問題 2

東京の各月の平均温度を示す。

平均気温が 10℃以上 20℃未満の月数と 20℃以上の月数を表示するプログラムを作成せよ。

1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
7.7	4.9	8.4	14.3	18.2	22.3	24.8	27.7	24.2	16.6	9.6	5.9

解答例

```
#include "stdio.h"

int main( void ){
    double    temp[12] = {7.7, 4.9, 8.4, 14.3, 18.2,
                          22.3, 24.8, 27.7, 24.2,
                          16.6, 9.6, 5.9};

    int        mon10, mon20;
    double     t0;
    int        i;

    mon10 = 0;
    mon20 = 0;

    for (i = 0; i < 12; i++) {

        t0 = temp[i];

        if ((t0 >= 10) && (t0 < 20)) {
            mon10++;
        }
        if (t0 >= 20) {
            mon20++;
        }

    }

    printf("10℃以上20℃未満の月数 %d\n", mon10);
    printf("20℃以上の月数 %d\n", mon20);

    return 0;
}
```

過去問 (2014年度より)

問題3

右に示すように、読み込んだ整数の段数を持つ
直角三角形（右上が直角）を表示するプログラム
を作成せよ。

実行例

何段ですか: 5 ↵

```
*****
****
***
**
*
```

補足説明

forループjでは一行にあるスペースと☆の数を表示する

読み込んだ整数を5とした例で考えてみる

スペースの数 ☆の数

1段目:	0	5
2段目:	1	4
3段目:	2	3
4段目:	3	2
5段目:	4	1

スペースはi個

☆は(n-i)個

スペースの数 ☆の式

i=0:	0 (iと同じ)	5-0
i=1:	1 (iと同じ)	5-1
i=2:	2 (iと同じ)	5-2
i=3:	3 (iと同じ)	5-3
i=4:	4 (iと同じ)	5-4

考え方

一行の文字数=段数n

スペースの数=k 段目 - 1

星の数 = 段数n + 1 - k 段目

この部分は
どちらでもいける

```
for (i = n; i > 0; i--) {
    for (j = 0; j < n - i; j++) {
        printf(" ");
    }
    for (j = 1; j <= i; j++) {
        printf("*");
    }
    printf("\n");
}

return 0;
}
```

解答例

```
#include "stdio.h"

int main( void ){

    int        n, i, j;

    printf("何段ですか: ");
    scanf_s("%d", &n);

    for (i = 0; i < n ; i++) {
        for (j = 0; j < i; j++) {
            printf(" ");
        }
        for (j = 1 ; j <= n - i ; j++) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

過去問 (2014年度より)

問題 4

5人の学生の体重を読み込んで、最も重い人と最も軽い人の体重を表示するプログラムを作成せよ。

なお、実行結果が以下になるように、プログラムを作成すること。体重は **double** 型で表すものとする。

(#include から始めてください)

体重を入力してください。

1 番: 52.5

2 番: 64.0

3 番: 58.3

4 番: 67.4

5 番: 62.0

最も重い人の体重: 67.4

最も軽い人の体重: 52.5

解答例

```
wmax = -999.9 ;
wmin = 999.9 ;

for (i = 0; i <= 4; i++) {
    if (wmax < weight[i]) {
        wmax = weight[i];
    }

    if (wmin > weight[i]) {
        wmin = weight[i];
    }
}

printf("最も重い人の体重:%4.1f¥n", wmax);
printf("最も軽い人の体重:%4.1f¥n", wmin);

return 0;
}
```

この書き方
もできる

```
#include <stdio.h>

int main(void){
    double    weight[5];
    double    wmax, wmin;
    int        i;

    printf("体重を入力してください¥n");

    for (i = 0; i <= 4; i++) {
        printf("%d 番:", i + 1);
        scanf_s("%lf", &weight[i]);
    }

    wmax = weight[0];
    wmin = weight[0];

    for (i = 1; i <= 4; i++) {
        if (wmax < weight[i]) {
            wmax = weight[i];
        }

        if (wmin > weight[i]) {
            wmin = weight[i];
        }
    }

    printf("最も重い人の体重:%4.1f¥n", wmax);
    printf("最も軽い人の体重:%4.1f¥n", wmin);

    return 0;
}
```

過去問 (2014年度より)

問題 5

西暦 y 年 m 月 d 日の曜日を求めるプログラムを以下の Zeller の公式を用いて作成せよ。

Zeller の公式

$$h = \left\{ y + \left\lfloor \frac{y}{4} \right\rfloor - \left\lfloor \frac{y}{100} \right\rfloor + \left\lfloor \frac{y}{400} \right\rfloor + \left\lfloor \frac{13m + 8}{5} \right\rfloor + d \right\} \bmod 7$$

[] 内の割り算は整数で、mod は剰余である。

このとき $h = 0$ が日曜日で、 $h = 1$: 月曜日、 $h = 2$: 火曜日、... $h = 6$: 土曜日となる。

ただし、1月と2月については前年の13月、14月として計算しなければならない。

解答例

```
#include <stdio.h>

int main(void) {

    int        y, m, d;
    int        h0 , h ;
    char dd[7][3] = { "日", "月", "火", "水",
                      "木", "金", "土" };

    printf("(西暦)何年? ");
    scanf_s("%d", &y);
    printf("何月? ");
    scanf_s("%d", &m);
    printf("何日? ");
    scanf_s("%d", &d);

    if (m < 3) {
        y = y - 1;
        m = m + 12;
    }

    h0 = y + y / 4 - y / 100 + y / 400
        (13 * m + 8) / 5 + d;
    h = h0 % 7;

    printf("%n%s曜日です\n", dd[h]);

    return 0;
}
```

2016年度過去問より配列をもっと練習しましょう

問題1 次の配列 `d0[9]` がある。 `d0` のデータから `k` の前後3つのデータの平均

$$d1[k] = (d0[k-1] + d0[k] + d0[k+1]) / 3$$

を要素とする配列 `d1[9]` を計算するプログラムを作成せよ。ただし、`k = 0` のときは `d0[0]` と `d0[1]` の平均、`k = 8` のときは `d0[7]` と `d0[8]` の平均で代用する。(20点)

i	0	1	2	3	4	5	6	7	8
d0	8.4	14.3	18.2	22.3	24.8	27.7	24.2	16.6	9.6
d1	11.35								13.1

解答例

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

```
int main(void){
```

```
double    d0[9] = { 8.4, 14.3, 18.2, 22.3,
                    24.8, 27.7, 24.2, 16.6, 9.6 };
double    d1[9];
int       k;

d1[0] = (d0[0] + d0[1]) / 2;
d1[8] = (d0[7] + d0[8]) / 2;

for (k = 1; k < 8; k++) {
    d1[k] = (d0[k-1] + d0[k] + d0[k+1]) / 3;
    printf("d1[ %d ] = %f ¥n", k, d1[k]);
}
```

```
return 0;
```

```
}
```

この部分は
どちらでもいける

考え方

- $n=3$ の移動平均

$$d1[k] = (d0[k-1] + d0[k] + d0[k+1]) / 3$$

- 例外(1) $k=0$

$$d1[0] = (d0[0] + d0[1]) / 2$$

- 例外(2) $k=8$

$$d1[8] = (d0[7] + d0[8]) / 2$$

```
double    d0[9] = { 8.4, 14.3, 18.2, 22.3,
                    24.8, 27.7, 24.2, 16.6, 9.6 };
double    d1[9];

double    sum;
int       k,i;

d1[0] = (d0[0] + d0[1]) / 2;
d1[8] = (d0[7] + d0[8]) / 2;

for (k = 1; k < 8; k++) {
    sum = 0;
    for (i = -1; i <= 1; i++) {
        sum += d0[k + i];
    }
    d1[k] = sum / 3;
    printf("d1[ %d ] = %f ¥n", k, d1[k]);
}
```

2016年度過去問より

問題2 右に示すような図形を表示するプログラムを作成する。図形の左には星の数を表示し、最後にすべての星の数を表示する。下のプログラムの (a)~(h)の部分を埋めよ。

```
#include <stdio.h>
int main(void) {
    (a)
    int sum;
    (b)
    for (i = 1; i <= 9; i++) {
        if (i <= 5){
            (c)
        }
        else{
            (d)
        }
        printf("%d ", k);
        for (j = 1; j <= k; j++){
            printf("*");
        }
        (e)
        sum = (f)
    }
    printf(" * の数は %d です。", (g));
    (h)
}
```



```
> problem2.exe
1  *
2  **
3  ***
4  ****
5  *****
3  ***
2  **
1  *
0
* の数は 21 です。
>
```

- *****を5個表示するには
 - `printf("**");` で「*」が一つだけ表示される
 - 次に `printf("**");` が実行されると「* *」になる
 - `printf("**");` を 5回繰り返せば「* * * * *
 - ただし、次に `printf("**");` が実行されると * 6個になるので 5回繰り返した後に改行 `printf("\n");` を行う

解答

(a) `int i, j, k ;`
 (b) `sum = 0;`
 (c) `k = i;`
 (d) `k = 9 - i;`

(e) `printf("\n");`
 (f) `sum + k;`
 (g) `sum`
 (h) `return 0;`

2016年度過去問より配列をもっと練習しましょう

問題3 2進数 1001 1101 を10進数 (157) に変換するプログラムを作成せよ。なお2進数は以下のような配列 bit[8] を用いること。bit[0] が最上位の桁 (2 の7乗) であることに注意せよ。
(#include から始めてください) (20点)

```
int bit[8] = { 1, 0, 0, 1, 1, 1, 0, 1 };
```

考え方

- 2進数 **1001 1101** は以下のような数である。

1 の位(bit[7])	1	$1 \times 1 = 1$
2 の位(bit[6])	0	$2 \times 0 = 0$
4 の位(bit[5])	1	$4 \times 1 = 4$
:		
64 の位(bit[1])	0	$64 \times 0 = 0$
128 の位(bit[0])	1	$128 \times 1 = 128$

合計 157

	0	1	2	3	4	5	6	7
bit	1	0	0	1	1	1	0	1
桁	128	64	32	16	8	4	2	1

\uparrow \uparrow \uparrow \uparrow \uparrow
 2^7 2^6 2^2 2^1 2^0

解答例 1

```
#include <stdio.h>

int main(void){

    int bit[8] = { 1, 0, 0, 1, 1, 1, 0, 1 };
    int dec = 0;
    int d0 = 128;

    int i;

    for (i = 0; i <= 7; i++) {
        dec = dec + bit[i] * d0;
        d0 = d0 / 2;
    }
    printf("10進数は %d\n", dec);
    return 0;
}
```

2016年度過去問より配列をもっと練習しましょう

問題3 2進数 1001 1101 を10進数 (157) に変換するプログラムを作成せよ。なお2進数は以下のような配列 bit[8] を用いること。bit[0] が最上位の桁 (2 の7乗) であることに注意せよ。
(#include から始めてください) (20点)

```
int bit[8] = { 1, 0, 0, 1, 1, 1, 0, 1 };
```

解答例 2 抜粋

```
int main(void) {

    int bit[8] = { 1, 0, 0, 1, 1, 1, 0, 1 };
    int dec = 0;
    int d0 = 1;

    int i;

    for (i = 7; i >=0; i--) {
        dec = dec + bit[i] * d0;
        d0 = d0 * 2;
    }

    printf("10進数は %d¥n", dec);
    return 0;
}
```

2016年度過去問より配列をもっと練習しましょう

解答例

問題5

下に示すような3行3列の `double` の行列 `A` の転置行列 `AT` を作成し、

$$A = \begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \\ 7.0 & 8.0 & 9.0 \end{pmatrix}$$

$$B = A + AT$$

$$C = A - AT$$

の演算を行うプログラムを作成せよ。計算結果は表示させなくてもよい。

(`#include` から始めてください)

(20点)

```
#include <stdio.h>

int main(void) {

    double A[3][3] = { { 1.0, 2.0, 3.0 },
                        { 4.0, 5.0, 6.0 },
                        { 7.0, 8.0, 9.0 } };

    double AT[3][3], B[3][3], C[3][3];

    int i, j;

    /* 転置行列の作成 */
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            AT[i][j] = A[j][i];
        }
    }

    /* B = A + AT, C = A - AT */
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            B[i][j] = A[i][j] + AT[i][j];
            C[i][j] = A[i][j] - AT[i][j];
        }
    }

    return 0 ;
}
```

2017年度過去問より穴埋め問題を練習しましょう

問題2 右に示すような図形を表示するプログラムを作成する。図形の左には星の数を表示し、最後にすべての星の数を表示する。下のプログラムの (a)~(h) の部分を埋めよ。(20点)

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

int main(void) {
    int i, j, k, m;

    for ( (a) )
    {
        k = abs( i - 4 ) ;
        m = (b) ;
        for ( j = 1 ; j <= k ; j++ )
        {
            printf(" ");
        }
        for ( j = 1 ; j <= m ; j++ )
        {
            (c) ;
        }
        for ( (d) )
        {
            printf(" ");
        }
        printf("\n") ;
    }
    return 0;
}
```



補足説明

	空白の数	☆の数	空白の数
1段目:	3	1	3
2段目:	2	3	2
3段目:	1	5	1
4段目:	0	7	0
5段目:	1	5	1
6段目:	2	3	2
7段目:	3	1	3

forループ j では一行にある空白、☆、空白の数を表示する

空白 k 個 ☆ (7-2k) 個 空白 k 個

空白の数 ☆の式

i=1:	3 = i-4	1 =
i=2:	2 = i-4	3 =
i=3:	1 = i-4	5 =
i=4:	0 = i-4	7 =
i=5:	1 = i-4	5 =
i=6:	2 = i-4	3 =
i=7:	3 = i-4	1 =

7-2k

k

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

int main(void)
{
    int i, j, k, m;

    for ( i = 1 ; i <= 7 ; i++ )
    {
        k = abs( i - 4 ) ;
        m = 7 - 2 * k ;
        for ( j = 1 ; j <= k ; j++ )
        {
            printf( " " ) ;
        }
        for ( j = 1 ; j <= m ; j++ )
        {
            printf( "*" ) ;
        }
        for ( j = 1 ; j <= k ; j++ )
        {
            printf( " " ) ;
        }
        printf("\n") ;
    }
    return 0;
}
```

考え方

- ある1行を考えると
 - k 個の空白
 - m 個の "*"
 - ? 個の空白
- 空白の個数 $k = \text{abs}(i - 4)$
 - $k + m + k = 7$ なので $m = 7 - 2 * k$

解答例