

演習問題 17

問1. 構造体 `struct Potter` を宣言し、実行例のように、キーボードからハリーポッターの題名 (原題)、日本語の題名、発行年を入力し、ディスプレイに表示するプログラムを作成せよ。

実行例

英語のタイトルを入力してください。

Harry Potter and the Philosopher's Stone

日本語のタイトルを入力して下さい。

ハリーポッターと賢者の石

発行年を入力して下さい。

1997

English Title : Harry Potter and the Philosopher's Stone

邦訳 : ハリーポッターと賢者の石

発行年 : 1997

プログラム

```
#include<stdio.h>

typedef struct Potter{
    /* ここに文を書きます。*/
}Potter;

int main(void){

    Potter potter1;

    printf("英語のタイトルを入力してください。 \n");
    /* ここに文を書きます。*/

    printf("日本語のタイトルを入力して下さい。 \n");
    /* ここに文を書きます。*/

    printf("発行年を入力して下さい。 \n");
    /* ここに文を書きます。*/

    printf("English Title : %s\n", potter1.title);
    printf("邦訳 : %s\n", potter1.jpstitle);
    printf("発行年 : %d\n", potter1.year);

    return 0;
}
```

問2. 1. の構造体の配列を作成し、実行例のように、キーボードからハリー・ポッターの作品を3つ入力し、ディスプレイに表示するプログラムを作成せよ。

実行例

英語のタイトルを入力してください。

Harry Potter and the Philosopher's Stone

日本語のタイトルを入力して下さい。

ハリーポッターと賢者の石

発行年を入力して下さい。

1997

英語のタイトルを入力してください。

Harry Potter and the Chamber of Secrets

日本語のタイトルを入力して下さい。

ハリーポッターと秘密の部屋

発行年を入力して下さい。

1998

英語のタイトルを入力してください。

Harry Potter and the Prisoner of Azkaban

日本語のタイトルを入力して下さい。

ハリーポッターとアズカバンの囚人

発行年を入力して下さい。

1999

English Title : Harry Potter and the Philosopher's Stone

邦訳 : ハリーポッターと賢者の石

発行年 : 1997

English Title : Harry Potter and the Chamber of Secrets

邦訳 : ハリーポッターと秘密の部屋

発行年 : 1998

English Title : Harry Potter and the Prisoner of Azkaban

邦訳 : ハリーポッターとアズカバンの囚人

発行年 : 1999

プログラム

```
#include<stdio.h>

typedef struct Potter{

    /* ここに文を書きます。*/

}Potter;

int main(void){

    /* ここに文を書きます。*/

    for(i=0; i<3; i++){
        printf("英語のタイトルを入力してください。 \n");
        /* ここに文を書きます。*/

        printf("日本語のタイトルを入力して下さい。 \n");
        /* ここに文を書きます。*/

        printf("発行年を入力して下さい。 \n");
        /* ここに文を書きます。*/

        getchar(); //おまじないだと思って付ける！
    }

    for(i=0 ; i<3 ; i++){

        /* ここに文を書きます。*/

    }

    return 0;
}
```

問3. 複素数 (complex number) は、 $z = x + iy$ のように、二つの実数 x と y の組によって表される。ここで、 i は虚数単位で、 $i^2 = -1$ が成り立つ。複素数 $z_1 = x_1 + iy_1$ と $z_2 = x_2 + iy_2$ の足し算 (addition) の結果 $z_1 + z_2 = z_3$ は、 $z_3 = (x_1 + x_2) + i(y_1 + y_2)$ で与えられる。

複素数を表す構造体 `struct Cnum` と複素数の足し算を行う関数 `struct addCnums` を作成し、実行例のように2つの複素数の和を計算するプログラムを作成せよ。なお、引数の受け渡しは、値渡しでおこなうものとする。

実行例1

複素数 cn1 を入力してください。

実部 : 2.7

虚部 : 5.1

複素数 cn2 を入力してください。

実部 : 3.6

虚部 : 8.9

cn1 = 2.7+5.1i

cn2 = 3.6+8.9i

cn1 + cn2 = 6.3+14.00i

実行例2

複素数 cn1 を入力してください。

実部 : 8.2

虚部 : -4.3

複素数 cn2 を入力してください。

実部 : -5.4

虚部 : 1.9

cn1 = 8.2-4.3i

cn2 = -5.4+1.9i

cn1+cn2 = 2.8-2.4i

問4. 問3.のプログラムにおいて関数の引数をポインタで渡すようにあらためよ。

実行例

複素数 cn1 を入力してください。

実部 : 3.5

虚部 : -9.4

複素数 cn2 を入力してください。

実部 : -5.6

虚部 : -7.1

cn1 = 3.5-9.4i

cn2 = -5.6-7.1i

cn1+cn2 = -2.1-16.5i

問 5. 2つの複素数 (complex number) $z_1 = x_1 + iy_1$ と $z_2 = x_2 + iy_2$ の掛け算 (multiplication) $z_1 \times z_2 = z_3$ は、 $z_3 = (x_1 \times x_2 - y_1 \times y_2) + i(x_1 \times y_2 + x_2 \times y_1)$ で与えられる。

複素数を表す構造体 `struct Cnum` と複素数の掛け算を行う関数 `struct multCnums` を作成し、実行例のように2つの複素数の積を計算するプログラムを作成せよ。なお、引数の受け渡しは、値渡しでおこなうものとする。

実行例 1

```
複素数 cn1 を入力してください。  
実部：-0.5  
虚部：0.86603  
複素数 cn2 を入力してください。  
実部：-0.5  
虚部：0.86602  
cn1 = -0.50000+0.86603i  
cn2 = -0.50000+0.86602i  
cn1*cn2 = -0.50000-0.86603i
```

実行例 2

```
複素数 cn1 を入力してください。  
実部：1  
虚部：1  
複素数 cn2 を入力してください。  
実部：3  
虚部：-3  
cn1 = 1.00000+1.00000i  
cn2 = 3.00000-3.00000i  
cn1*cn2 = 6.00000+0.00000i
```

問 6. 問 5. のプログラムにおいて関数の引数をポインタで渡すようにあらためよ。

実行例

```
複素数 cn1 を入力してください。  
実部：3.3  
虚部：7.5  
複素数 cn2 を入力してください。  
実部：-4.1  
虚部：6.8  
cn1 = 3.30000+7.50000i  
cn2 = -4.10000+6.80000i  
cn1*cn2 = -64.53000-8.31000i
```

問7. 3次元ベクトルは、3つの数字の組から構成されている。2つのベクトル $v1 = (v1.x, v1.y, v1.z)$ と $v2 = (v2.x, v2.y, v2.z)$ の内積(ドット積 dot product) は、

$$v1 \cdot v2 = v1.x * v2.x + v1.y * v2.y + v1.z * v2.z$$

で与えられる。3つの整数の組からなる3次元ベクトルの構造体 `struct Vector` と内積を計算する関数 `dotProd(Vector v1, Vector v2)` を作成し、実行例のように、キーボードから2つのベクトルの成分を代入し、それらの内積を表示するプログラムを作成せよ。ただし、関数への引数渡しは、値渡しでおこなうものとする。

実行例

```
ベクトル1を入力
2 7 4
ベクトル2を入力
5 1 3
vc1=(2, 7, 4)
vc2=(5, 1, 3)
vc1 · vc2 =29
```

問8. 問7.のプログラムにおいて関数の引数をポインタで渡すようにあらためよ。

実行例

```
ベクトル1を入力
7 2 4
ベクトル2を入力
3 9 6
vc1=(7, 2, 4)
vc2=(3, 9, 6)
vc1 · vc2 = 63
```

問9. 2つの3次元ベクトル $v1 = (v1.x, v1.y, v1.z)$ と $v2 = (v2.x, v2.y, v2.z)$ の外積 (クロス積 cross product) は、

$$v1 \times v2 = (v1.y*v2.z - v1.z*v2.y, v1.z*v2.x - v1.x*v2.z, v1.x*v2.y - v1.y*v2.x)$$

で与えられる3次元ベクトルである。3つの整数の組からなる3次元ベクトルの構造体 `struct Vector` と外積を計算する関数 `crossProd(Vector v1, Vector v1)` を作成し、実行例のように、キーボードから2つのベクトルの成分を代入し、それらの内積を表示するプログラムを作成せよ。ただし、関数への引数渡しは、値渡しでおこなうものとする。

実行例

```
ベクトル1を入力
3 4 5
ベクトル2を入力
9 6 1
vc1=(3, 4, 5)
vc2=(9, 6, 1)
vc1 x vc2 = (-26, 42, -18)
```

問10. 問9.のプログラムにおいて関数の引数をポインタで渡すようにあらためよ。

実行例

```
ベクトル1を入力
1 2 3
ベクトル2を入力
3 2 1
vc1=(1, 2, 3)
vc2=(3, 2, 1)
vc1 x vc2 = (-4, 8, -4)
```

問 11. 3 つの実数の組からなる 3 次元ベクトルの構造体 `struct Vector` とその絶対値を計算する関数 `absValue(Vector vc1)` を作成し、実行例のように、キーボードからベクトルの 3 成分を代入し、その絶対値を表示するプログラムを作成せよ。ただし、関数への引数渡しは、値渡しでおこなうものとする。

【注】

- (1) ベクトル $v = (v_x, v_y, v_z)$ の絶対値 $|v|$ は、 $|v| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ で与えられる。
- (2) 平方根の計算には、`math.h` 関数を用いよ。

実行例 1

ベクトルを入力

3 4 12

`vc1=(3.000000, 4.000000, 12.000000)`

`|vc1| = 13.000000`

実行例 2

ベクトルを入力

0.5 1.2 8.4

`vc1=(0.500000, 1.200000, 8.400000)`

`|vc1| = 8.500000`

問 12. 問 11. のプログラムにおいて関数の引数をポインタで渡すようにあらためよ。

実行例 1

ベクトルを入力

3.4 5.6 7.9

`vc1=(3.400000, 5.600000, 7.900000)`

`|vc1| = 10.263040`

実行例 2

ベクトルを入力

0.8 1.5 14.4

`vc1=(0.800000, 1.500000, 14.400000)`

`|vc1| = 14.500000`

問 13. 実行例のように、キーボードから二つの分数を入力して、それらの足し算をおこなうプログラムを、分数を表す構造体 struct Frac を定義し、分数の足し算をおこなう関数 addFrac(Frac f1, Frac f2) を作成して完成させなさい。ただし、引数は、値渡しで行う。

実行例

```
分数 1 を入力 : 1/6
分数 2 を入力 : 10/3
1/6+10/3=7/2

分数 1 を入力 : 2/3
分数 2 を入力 : 7/3
2/3+7/3=3
```

プログラム

```
#include<stdio.h>
typedef struct Frac{
    int n;    // n は、numerator(分子) の n
    int d;    // d は、denominator(分母) の d
}Frac;

int gcd(int n, int r){
    /*   ここに文を書きます。   */
}

struct Frac addFrac(Frac f1, Frac f2){
    /*   ここに文を書きます。   */
}

int main(void){

    Frac f1, f2, f3;
    char ch;
    printf("分数 1 を入力 : ");
    scanf("%d%c%d", &f1.n, &ch, &f1.d);
    printf("分数 2 を入力 : ");
    scanf("%d%c%d", &f2.n, &ch, &f2.d);

    /*   ここに文を書きます。   */

    return 0;
}
```

問 14. 問 13. のプログラムにおいて関数の引数をポインタで渡すように改めよ。

実行例

分数 1 を入力 : 2/5

分数 2 を入力 : 8/5

2/5+8/5=2

分数 1 を入力 : 1/4

分数 2 を入力 : 1/6

1/4+1/6=5/12

プログラム

```
#include<stdio.h>
typedef struct Frac{
    int n;    // n は、numerator(分子) の n
    int d;    // d は、denominator(分母) の d
}Frac;

int gcd(int n, int r){
    /*   ここに文を書きます。   */
}

void addFrac(Frac *f1, Frac *f2, Frac *f3){
    /*   ここに文を書きます。   */
}

int main(void){

    Frac f1, f2, f3;
    char ch;
    printf("分数 1 を入力 : ");
    scanf("%d%c%d", &f1.n, &ch, &f1.d);
    printf("分数 2 を入力 : ");
    scanf("%d%c%d", &f2.n, & ch, &f2.d);

    addFrac(&f1, &f2, &f3);

    /*   ここに文を書きます。   */

    return 0;
}
```

問 15. 山名とその標高をメンバに持つ山の構造体 `struct Mount` を作成し、実行例のようにキーボードから2つの山を入力し、ディスプレイに表示するプログラムを作成せよ。

実行例

青森の山を2つ入力してください。

山名：岩木山

標高：1625

山名：八甲田山

標高：1585

青森の山々

岩木山 1625m

八甲田山 1585m

問 16. 2つの `struct` 型のオブジェクトを交換する関数 `swap` を作成し、実行例のように2つの山名とその標高をキーボードから入力した順序でディスプレイに表示した後、両者を入れ替えて表示するプログラムを作成せよ。

実行例

青森の山を2つ入力してください。

山名：岩木山

標高：1625

山名：八甲田山

標高：1585

青森の山々

岩木山 1625m

八甲田山 1585m

2つの山を入れ替えました。

青森の山々

八甲田山 1585m

岩木山 1625m