

プログラミングII

<http://bit.ly/Prog3i>

Cプログラミングの復習

前期 第1週

2019/4/10

今回は以下の内容を確認してみます

- ▶ **ポインタ**を利用した処理
- ▶ ポインタを利用した**配列**の処理
- ▶ ポインタを利用した**構造体**の処理

【課題1-1】

仮引数で与えられた3つのポインタ num1, num2, num3 が参照する整数に対して、「*num1の値が *num2へ、*num2の値が *num3へ、*num3の値が *num1へ、と1つ隣に値が移る」関数 shift_num() を作成して下さい。

[この関数のプロトタイプ宣言]

```
void shift_num(int *num1, int *num2, int *num3);
```

```
/* num1, num2, num3が参照している整数値を代入処理を用いて移す */  
/* 値を一時的に退避するための変数を用意する必要がある */
```

【課題1-1】

[mainでの処理]

```
int n1, n2, n3;  
n1 = 6; n2 = 2; n3 = -8;  
  
shift_num(&n1, &n2, &n3); /* 1回隣に移す */  
printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);  
  
shift_num(&n1, &n2, &n3); /* 更にもう1回隣に移す */  
printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);  
  
shift_num(&n1, &n2, &n3); /* 更にもう1回隣に移すと値が元に戻る */  
printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);
```

[実行例]

```
n1: -8, n2: 6, n3: 2  
n1: 2, n2: -8, n3: 6  
n1: 6, n2: 2, n3: -8
```

【課題1-2】

仮引数のポインタが参照する配列に対して、指定された範囲内の要素の合計を求める関数sum_range()を作成してください。

[この関数のプロトタイプ宣言]

```
int sum_range(int *ptr, int s, int e);
```

```
/* ptrが参照する配列のs番目からe番目までの要素の合計を求め、  
   結果をreturnで戻す */
```

```
/* 「s <= e」を満たした値が渡されることを前提にしてよい */
```

【課題1-2】

[mainでの処理]

```
/* 配列はグローバルに宣言するのではなく、main()内で宣言する */
```

```
int test[5] = {80, 60, 55, 22, 75};
```

```
int test2[5] = {76, 85, 47, 92, 68};
```

```
printf("sum_range: %d\n", sum_range(test, 1, 3));
```

```
printf("sum_range: %d\n", sum_range(test2, 2, 4));
```

[実行結果]

```
sum_range: 137
```

```
sum_range: 207
```

【課題1-3】

次のような時間（時と分）の情報を持つ構造体型を宣言します。

```
typedef struct Time {  
    int hour;  
    int minute;  
} Time;
```

この構造体の情報を画面に出力する関数show()を作成してください。

[この関数のプロトタイプ宣言]

```
void show(Time *t, char *s);
```

```
/* まず、sが参照している文字列を出力する */
```

```
/* tが参照しているメンバhourとminuteを出力する */
```

```
/* 出力の書式は実行結果の様子を参照 */
```

【課題1-3】

[mainでの処理]

```
Time t1 = {40, 50}; /* 40時間50分 */  
Time t2 = {30, 40}; /* 30時間40分 */  
show(&t1, "initial t1");
```

[実行結果]

```
(initial t1) 40:50
```


まだ余裕のある人は…

【課題1-4】

「引数が参照している2つのTimeの構造体が表す時間を加算する」関数add_time()を作成してください。

[この関数のプロトタイプ宣言]

```
void add_time(Time *t1, Time *t2);
```

```
/* メンバhour同士とminute同士を加算する */
```

```
/* 加算した結果は、t1が参照しているメンバへ代入する */
```

```
/* minuteで60分を超える場合は、
```

```
hourへ繰り上げる (minuteと60との商をhourへ加える) */
```

【課題1-4】

[mainでの処理(課題1-3の処理に続けて書く)]

```
add_time(&t1, &t2);    /* 40:50 と 30:40 を加算する */
```

```
show(&t1, "added t1");
```

[実行結果]

```
(added t1) 71:30
```