

## 小テスト

プログラムファイルの先頭行に、C のコメントとして自分の番号と名前を書いてください。

【問 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は以下のようになります。

```
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 */  
/* 交換処理と同様に、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 */
```

main() で動作を確認してください。

```
[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
```

(20 点)

### 小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。つまり、**通常の定期試験と同様**。
- 小テスト中は、**演習室外へのネットワークアクセスは遮断される**。

### 小テスト中に参照できるもの

- 教科書と配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- \* **上記以外の情報を参照することはカンニング行為とする**  
(例：USB で接続された機器に保存されているファイルの参照, ネットワークを介した情報の参照など)

### 答案の提出

- 保存したファイルは次のように「report」コマンドで提出する（ちゃんと提出できた場合は、「Succeed.」と画面に表示される）  
\$ ~kogai/report 「提出先」 「プログラムファイル」
- 複数のファイルを提出する場合は、report コマンドを分けて提出する。例えば、test1.c と test2.c のファイルを提出したい場合は、次のように 2 回に分けて提出する  
\$ ~kogai/report kiso3 test1.c  
\$ ~kogai/report kiso3 test2.c
- 同じ問題に対して、複数の提出ファイルが存在した場合は、更新日時が新しい方を提出ファイルとする
- 提出するファイルは、誰から提出されたのか区別されるため、ファイル名は各自で自由に決めて良い
- 「提出先」への提出は試験時のみ可能である

## 小テストの模範解答

```
/* 自分の番号と名前をここに書く */
#include <stdio.h>

/* 関数のプロトタイプ宣言 */
void shift_num(int *num1, int *num2, int *num3);

/* 変数の値を入れ替える */
void shift_num(int *num1, int *num2, int *num3)
{
    int tmp;
    /* *num3 を tmp へ */
    tmp = *num3;
    /* *num2 を*num3 へ */
    *num3 = *num2;
    /* *num1 を*num2 へ */
    *num2 = *num1;
    /* tmp を*num1 へ */
    *num1 = tmp;
}

int main(void)
{
    int n1, n2, n3;

    /* sort_num() の動作確認 */
    n1 = 6; n2 = 2; n3 = -8;
    shift_num(&n1, &n2, &n3);
    printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);

    shift_num(&n1, &n2, &n3);
    printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);

    shift_num(&n1, &n2, &n3);
    printf("n1: %d, n2: %d, n3: %d\n", n1, n2, n3);

    return 0;
}
```