

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】 int 型の配列 test を以下のようにグローバル変数として宣言する。

```
#define N 5
int test[N];
```

この配列に対して、「各要素の値を、 2^i (2 の i 乗) へと変換する」関数 pow_array() を作成してください。ただし、 i は配列の添字を表しており、 $0 \leq i \leq 4$ となる整数とします。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 pow_array();
/* 課題 5-2 を参考に、配列 test の全要素に対する繰り返し処理を作る */
/* 繰り返し処理の中で課題 5-1 を参考に、関数 pow() 使って求める */
/* 処理の結果は配列 test に反映される */
```

main() で動作を確認してください。配列 test の出力には以下の関数 show() が必要です。(第 4 回 サンプルプログラムと同じです)

```
void show()
{
    int i;
    for(i=0; i<N; i++) {
        printf("test[%d]: %d\n", i, test[i]);
    }
}
```

[main() での処理]

```
pow_array();
show();
```

[実行例]

```
test[0]: 1
test[1]: 2
test[2]: 4
test[3]: 8
test[4]: 16
```

小テストの注意点

(20 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, 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 kiso5 「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 kiso5 test1.c
\$ ~kogai/report kiso5 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
(ただし、提出するファイルの先頭には、出席番号と氏名を記入する)

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>
#include <math.h>

/* 配列をグローバル変数として宣言する */
#define N 5
int test[N];

/* 関数のプロトタイプ宣言 */
void show();
void pow_array();

/* 配列 test の要素を出力する */
void show()
{
    int i;
    for(i=0; i<N; i++) {
        printf("test[%d]: %d\n", i, test[i]);
    }
}

/* 配列 test の全ての要素を 2 の i 乗にする */
void pow_array()
{
    int i;
    for(i=0; i<N; i++) {
        /* pow() を全ての要素に対して行う */
        test[i] = pow(2, i);
    }
}

int main(void)
{
    /* pow_array() の動作確認 */
    pow_array();
    show();

    return 0;
}
```