

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】 仮引数に与えられた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中で、**仮引数 n と値が等しくなる要素の個数を求める関数 `count_eq()`** を作成して下さい。ただし、仮引数 s, e は、 $s \leq e$ であり、配列の大きさを超えて指定されないことを前提に作ってよい。この関数のプロトタイプ宣言は以下のようになります。

```
int count_eq(int *ptr, int s, int e, int n);
/* 個数を数えるための変数を用意する */
/* ptr が参照する配列の s 番目から e 番目まで繰り返し処理を作る（課題 4-1 参照） */
/* 繰り返し処理の中で、ptr が参照している配列の i 番目が n と等しい場合、個数を 1 つ増やす */
/* 繰り返し処理終了後に、数えた個数を戻り値とする */
```

`main()` で動作を確認してください。

```
[main() での処理]
int a1[3] = {-9, 7, 7};
int a2[5] = {2, 2, -1, 5, 2};
printf("%d\n", count_eq(a1, 0, 2, 7));
printf("%d\n", count_eq(a2, 0, 4, 2));
printf("%d\n", count_eq(a2, 1, 3, 2));
[実行例]
2      (← 配列 a1 の 0~2 番目で 7 の個数)
3      (← 配列 a2 の 0~4 番目で 2 の個数)
1      (← 配列 a2 の 1~3 番目で 2 の個数)
```

(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不正行為とする
(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 kiso4 「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 kiso4 test1.c
\$ ~kogai/report kiso4 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

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 */
int count_eq(int *ptr, int s, int e, int n);

/* 指定した範囲内の要素の値が n と等しくなる個数を求める */
int count_eq(int *ptr, int s, int e, int n)
{
    int i, result;
    result = 0;
    /* s 番目から e 番目までを繰り返し n と比較する */
    for(i=s; i<=e; i++) {
        if( *(ptr+i) == n ) {
            result ++;
        }
    }
    return result;
}

int main(void)
{
    int a1[3] = {-9, 7, 7};
    int a2[5] = {2, 2, -1, 5, 2};

    /* count_eq() の動作確認 */
    printf("%d\n", count_eq(a1, 0, 2, 7));
    printf("%d\n", count_eq(a2, 0, 4, 2));
    printf("%d\n", count_eq(a2, 1, 3, 2));

    return 0;
}
```