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】 「引数で与えられた車のナンバーが 10000 未満の場合はそのまま追加し、10000 以上の場合は下 4 桁の値をナンバーとして追加する」関数 add_car4() を作成してください。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 add_car4(Car *p, int n, double g);
  /* 課題 13-1 で作成した関数において、以下のような条件を追加する */
  /* まず、領域を確保して… */
  /* ・n が 10000 未満の場合は、そのまま num に代入する  */
  /* ・n がそれ以外の場合は、n の下 4 桁の値（つまり 10000 で割った余り）を num に代入する  */
  /* g はそのままメンバ gas に代入する */
```

main() で動作を確認してください。動作の確認には、関数 show_carlist() も必要になります。

```
[main() での処理]
  Car head4;
  head4.num = 0; head4.gas = 0; head4.next = NULL;
  show_carlist(&head4, "head4 (1)");
  add_car4(&head4, 135, 40.3);
  add_car4(&head4, 9999, 33.8);
  add_car4(&head4, 12345, 26.1);
  show_carlist(&head4, "head4 (2)");
[実行例]
  --- head4 (1) ---
  num: 0, gas: 0.000000
  --- head4 (2) ---
  num: 0, gas: 0.000000
  num: 2345, gas: 26.100000      (← 10000 以上の場合は下 4 桁が追加される)
  num: 9999, gas: 33.800000      (← 10000 未満の場合はそのまま追加される)
  num: 135, gas: 40.300000       (← 10000 未満の場合はそのまま追加される)
```

(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 kiso13 「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 kiso13 test1.c
$ ~kogai/report kiso13 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>
#include <stdlib.h>

/* 車の情報を持つ構造体の宣言 */
typedef struct Car {
    int num;
    double gas;
    struct Car *next;
} Car;

/* 関数のプロトタイプ宣言 */
void show_carlist(Car *start, char *str);
void add_car4(Car *p, int n, double g);

/* リストの要素を出力する */
void show_carlist(Car *start, char *str)
{
    Car *pcar;
    printf("--- %s ---\n", str);
    for(pcar = start; pcar!=NULL;
        pcar = pcar->next) {
        printf("num: %d, gas: %lf\n",
               pcar->num, pcar->gas);
    }
}

/* リストに要素を追加する (その 5) */
void add_car4(Car *p, int n, double g)
{
    Car *new;
    /* 構造体 Car の領域を確保する */
    new = (Car *)malloc(sizeof(Car));
    /* 確保した領域のメンバに引数の値を代入する */
    if(n <= 9999) {
        new->num = n;
    } else {
        new->num = n % 10000;
    }
    new->gas = g;
    /* 確保した領域の next を更新する */
    new->next = p->next;
    /* p の next は確保した領域を参照する */
    p->next = new;
}

int main(void)
{
    Car head4;
    /* add_car4() の動作確認 */
    head4.num = 0; head4.gas = 0; head4.next = NULL;
    show_carlist(&head4, "head4 (1)");
    add_car4(&head4, 135, 40.3);
    add_car4(&head4, 9999, 33.8);
    add_car4(&head4, 12345, 26.1);
    show_carlist(&head4, "head4 (2)");

    return 0;
}
```