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】 構造体型 struct Car（または Car 型）の配列に対して、「指定した要素のメンバ gas を、引数で与えられた金額で入る分だけ給油する」関数 charge_gas() を作成して下さい。ただし、**ガソリンは、1 リットル当たり 120 円であるものとし、与えられる金額の単位は円とし、メンバ gas の単位はリットルとします。**この計算は int 型で計算して構いません。（つまり、小数点以下は切り捨て）

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 charge_gas(Car *pC, int i, int yen);
  /* 引数 yen は、与えられた金額を表し単位は円とする */
  /* ポインタ pC が参照している配列の i 番目の要素に対して、
       メンバ gas を引数 yen で給油できる分だけ加算する */
```

main() で動作を確認してください。**実行には、関数 show_cars() を使うため、この関数の定義も書く必要があります。**

```
[main() での処理]
  Car cars3[3] = {
      {1234, 15.5},
      {4567, 12.2},
      {7890, 10.5}
  };
  charge_gas(cars3, 1, 1800);
  charge_gas(cars3, 2, 960);
  show_cars(cars3, "charge_gas: cars3");
[実行例]
  --- charge_gas: cars3 ---
  [0] num: 1234, gas: 15.500000
  [1] num: 4567, gas: 27.200000      (← 1800 円で入る分、つまり 15 リットル給油された)
  [2] num: 7890, gas: 18.500000      (← 960 円で入る分、つまり 8 リットル給油された)
```

【問 2】 main() に、**load_cars() と save_cars() を付け加えて**、「**実行するたびに、メンバ gas のが給油され続ける**」ようにプログラムを改良してして下さい。

ファイルの中身の初期値とその実行結果は以下のようになります。

「cars.csv」ファイルの中身が次のような場合

```
1234,5.5
4567,8.3
7890,2.5
```

```
[実行結果]
  $ ./a.out
  (loading)
  --- charge_gas: cars3 ---
  [0] num: 1234, gas: 5.500000
  [1] num: 4567, gas: 23.300000   (← 8.3 に 1800 円分給油された)
  [2] num: 7890, gas: 10.500000   (← 2.5 に 960 円分給油された)
  (saving)

  $ ./a.out
  (loading)
  --- charge_gas: cars3 ---
  [0] num: 1234, gas: 5.500000
  [1] num: 4567, gas: 38.300000   (← 23.3 に 1800 円分給油された)
  [2] num: 7890, gas: 18.500000   (← 10.5 に 960 円分給油された)
  (saving)
```

（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   kiso11   「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  kiso11  test1.c
  $  ~kogai/report  kiso11  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>

#define NUM 3

/* 車を表す構造体 */
typedef struct Car {
    int num;
    double gas;
} Car;

/* 関数のプロトタイプ宣言 */
void show_cars(Car *pC, char *str);
void save_cars(Car *pC);
void load_cars(Car *pC);
void charge_gas(Car *pC, int i, int yen);

/* 車の配列を出力する */
void show_cars(Car *pC, char *str)
{
    int i = 0;
    printf("--- %s ---\n", str);
    for(i=0; i<NUM; i++) {
        printf("[%d] num: %d, gas: %lf\n",
               i, (pC+i)->num, (pC+i)->gas);
    }
}

/* 車の配列をファイルに書き込む */
void save_cars(Car *pC)
{
    FILE *fp;
    int i;
    fp = fopen("cars.csv", "w");
    if(fp == NULL) {
        printf("ファイルが開けませんでした。\n");
        return;
    }
    printf("(saving)\n");
    for(i=0; i<NUM; i++) {
        fprintf(fp, "%d,%lf\n",
                (pC+i)->num, (pC+i)->gas);
    }
    fclose(fp);
}

/* 車の配列をファイルから読み込む */
void load_cars(Car *pC)
{
    FILE *fp;
    int i;
    fp = fopen("cars.csv", "r");
    if(fp == NULL) {
        printf("ファイルが開けませんでした。\n");
        return;
    }
    printf("(loading)\n");
    for(i=0; i<NUM; i++) {
        fscanf(fp, "%d,%lf\n",
               &(pC+i)->num, &(pC+i)->gas);
    }
    fclose(fp);
}

/* 指定した要素の gas を走行距離分だけ消費する */
void charge_gas(Car *pC, int i, int yen)
{
    /* i 番目のメンバ gas に g を加算する */
    /* 1 リットル 120 円の計算とする */
    (pC+i)->gas += yen / 120;
}

/* 以降、問ごとに異なる main() を使う */
/* 【問 1】ファイルの読み書きなし */
int main(void)
{
    Car cars3[3] = {
        {1234, 15.5},
        {4567, 12.2},
        {7890, 10.5}
    };
    charge_gas(cars3, 1, 1800);
    charge_gas(cars3, 2, 960);
    show_cars(cars3, "charge_gas: cars3");
    return 0;
}

/* 【問 2】ファイルの読み書きあり */
int main(void)
{
    Car cars3[3];

    load_cars(cars3);  /* 追加 */
    charge_gas(cars3, 1, 1800);
    charge_gas(cars3, 2, 960);
    show_cars(cars3, "charge_gas: cars3");
    save_cars(cars3);  /* 追加 */

    return 0;
}
```