

## 課題 13-1, 13-2 のプログラム例

```

#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_car(Car *p);
void add_car2(Car *pcar0, int n, double g);
void add_car3(Car *pcar0, 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 1) */
void add_car(Car *p)
{
    Car *new;

    new = (Car *)malloc(sizeof(Car));
    new->num = 1111; new->gas = 11.1;
    new->next = p->next;
    p->next = new;
}

/* リストに要素を追加する (その 2) */
void add_car2(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 */

```

```

    new->num = n;
    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;
}

/* リストに要素を追加する (その 3) */
void add_car3(Car *p, int n, double g)
{
    Car *new;
    /* ナンバーが奇数かどうか調べる */
    if(n % 2 != 0) {
        /* 構造体 Car の領域を確保する */
        new = (Car *)malloc(sizeof(Car));
        /* メンバ num に n を代入する */
        new->num = n;
        /* ガスの量が 0 かどうか調べる */
        if(g == 0) {
            new->gas = 10;
        } else {
            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 head2, head3;
    Car *new;
    /* add_car2() の動作確認 */
    head2.num = 0; head2.gas = 0;
    head2.next = NULL;
    show_carlist(&head2, "head2 (1)");
    add_car2(&head2, 1357, 40.3);
    add_car2(&head2, 2468, 33.8);
    add_car2(&head2, 3579, 26.1);
    show_carlist(&head2, "head2 (2)");
    /* add_car3() の動作確認 */
    head3.num = 0; head3.gas = 0;
    head3.next = NULL;
    show_carlist(&head3, "head3 (1)");
    add_car3(&head3, 1357, 40.3);
}

```

```

        add_car3(&head3, 2468, 33.8);
        add_car3(&head3, 3579, 0);
        show_carlist(&head3, "head3 (2)");

    return 0;
}

```

### 課題 13-3 のプログラム例

```

#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *in, *out;
    char ch;
    /* コマンドライン引数の個数をチェックする */
    if(argc != 3) {
        printf("パラメータの数が違います。\\n");
        return 1;
    }
    /* コピー元のファイルを読み込みモードで開く */
    in = fopen(argv[1], "r");
    /* コピー先のファイルを書き込みモードで開く */
    out = fopen(argv[2], "w");
    /* ファイルポインタが
       NULL の場合の処理は省略 */

    fprintf(out, "(character copy)\\n");
    /* ファイルの最後に到達するまで繰り返す */
    while((ch=fgetc(in)) != EOF) {
        printf("ch: %c \\n", ch);
        fputc(ch, out);
    }
    /* コピー元と先のファイルを閉じる */
    fclose(in);
    fclose(out);

    return 0;
}

```

### 課題 13-4 のプログラム例

```

#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])

```

```

{
    int i, result;

    result = 0;
    /* 1 番目以降の argv の文字列に対して繰り返す */
    for(i=1; i<argc; i++) {
        /* argv[i] の文字列の情報を出力する */
        printf("%s(%ld)\\n",
               argv[i], strlen(argv[i]));
        /* 合計に文字列の長さを加算する */
        result += strlen(argv[i]);
    }
    /* 加算した結果を出力する */
    printf("result: %d\\n", result);

    return 0;
}

```

### 課題 13-5 のプログラム例

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int i, result;

    result = 0;
    /* 1 番目以降の argv の文字列に対して繰り返す */
    for(i=1; i<argc; i++) {
        /* argv[i] の文字列の情報を出力する */
        printf("%s", argv[i]);
        /* argv[i] を整数に変換して加算する */
        result += atoi(argv[i]);
        /* 演算子の記号またはイコールを出力する */
        if(i != argc-1) {
            printf(" + ");
        } else {
            printf(" = ");
        }
    }
    /* 加算した結果を出力する */
    printf("%d\\n", result);

    return 0;
}

```