2013 年 7 月 18 日（木）実施

構造体

レコードと構造体

前々回の教材で触れたように，複数の項目に渡るデータを一まとめにしたものを**レコード**という。例えば，次のように学籍番号，氏名，履修科目コード，点数，評価といった項目による 1 人分のデータを一まとめにしたものは 1 件分のレコードである。

学籍番号	氏名	履修科目コード	点数	評価
12x0123	上田奈緒子	z1021234	100	S

C 言語では，このようなレコードのデータ構造を**構造体**（structure）によって**カプセル化**して表現する。構造体を用いれば，個々の項目を別々の変数で表す場合に比べてデータ間の関係が把握しやすく，プログラムの可読性が増す。

また，C 言語の関数では通常の変数を return 文に書いた場合，1 つのデータしか戻せないが，構造体を return 文に記述することが可能であるので，複数のデータを一まとめにして戻すことができる。

なお，配列同士は要素数が等しくても代入は行えないが，構造体同士はデータ構造が等しければ代入を行うことができる。

構造体の利用

C 言語で構造体を用いる際の一連の流れは次のようになる。

1) レコード設計を行い，**構造体の型定義**を行う

2) **構造体変数の宣言**を行う　（その際に初期化を行うことも可能）

3) 構造体変数の**メンバ**への代入，メンバの参照　（レコードの各項目をメンバと呼ぶ）

上述の学生データを例にとって，この一連の流れを表すと，次のようになる。

例）
```
struct STUDENT
{
    char          id[8];
    char      name[21];
    char      class[9];
    unsigned int point;
    char          eval;
};

struct STUDENT x;

strcpy(x.id,     "12x0123");
```

```
    strcpy(x.name,  "上田奈緒子");
    strcpy(x.class, "z1021234");

    x.point = 100;
    x.eval  = 'S';
```

この例で，STUDENT は**構造体タグ**と呼ばれる。ここに書いた形式のほかに，構造体の型定義と構造体変数の宣言とを一度に行うことも可能である。(**struct** 構造体タグ名{･･･}構造体変数名;)

例題 1（構造体で表現されたレコードの csv ファイルへの書き出し）

次のソースプログラムをテキストエディタで入力して，prog14-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。なお，実行時に入力する履修科目コードは **x1043234** である。それ以外の文字列を与えた場合の動作も確認すること。

```
/* prog14-1.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 5

int main(void)
{
    struct STUDENT
    {
        char         id[8];
        char      name[21];
        char      class[9];
        unsigned int point;
        char          eval;
    };

    struct STUDENT prog[MAX];

    int i;
    char temp[9];
    FILE *fpr;

    if (NULL == (fpr = fopen("stprog.csv","w")))
    {
        printf("stprog データファイルが開けません。¥n");
        exit(1);
    }

    fprintf(fpr,"学籍番号,氏名,履修科目コード,点数,評価¥n");

    for (i=0; i<MAX; i++)
    {

        printf("%d 人目の履修科目コードを入力してください：", i+1);
        scanf(" %s", temp);
```

```
        if (strcmp(temp,"x1043234") == 0)
        {
            strcpy(prog[i].class, temp);

            printf("%d 人目の学籍番号を入力してください：", i+1);
            scanf(" %s", prog[i].id);
            printf("%d 人目の氏名を入力してください：", i+1);
            scanf(" %s", prog[i].name);
            printf("%d 人目の点数を入力してください：", i+1);
            scanf(" %u", &prog[i].point);
            printf("%d 人目の評価を入力してください：", i+1);
            scanf(" %c", &prog[i].eval);

            fprintf(fpr,"%s,%s,%s,%u,%c¥n",
              prog[i].id,prog[i].name,prog[i].class,prog[i].point,prog[i].eval);
        }
        else
        {
            printf("履修科目コードが正しくありません。"
            "もう一度入力してください。¥n");
            i--;
        }
    }

    fclose(fpr);

    return 0;
}
```

【解説】

1. unsigned int は負号なし整数のデータ型であり，対応する変換指定には%u を用いる。

2. prog は，構造体の配列として宣言されている。この場合，配列の各要素が構造体となる。

3. strcpy は，第 2 引数に書かれた文字列を第 1 引数に複写する文字列操作のライブラリ関数である。

4. printf では，複数の文字列リテラルを並べると文字列は繋がって表示される。else 節にある printf では，文字列が長いので行を分割するためにこのような書き方をしている。

5. 文字コードが一致せずに再入力を促す場合，既にループ変数 i の値は 1 だけ進んでいるので，i--で元に戻してから，再入力させる。

例題 2（**構造体で表現されたレコードの csv ファイルからの読み込み**）

次のソースプログラムをテキストエディタで入力して，prog14-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。入力ファイルには例題 1 で作成されたものを用いる。

```
/* prog14-2.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 5
#define MAXBUF 256

char headln[MAXBUF];

struct STUDENT
{
    char         id[8];
    char      name[21];
    char      class[9];
    char      point[4];
    char       eval[2];
};

void dispdata(struct STUDENT [], int);

int main(void)
{

    struct STUDENT prog[MAX];

    int i;
    char temp[MAXBUF];
    FILE *fpr;

    if (NULL == (fpr = fopen("stprog.csv","r")))
    {
        printf("stprog データファイルが開けません。¥n");
        exit(1);
    }

    fgets(headln, sizeof headln, fpr);

    for (i=0; i<MAX; i++)
    {
        if (fgets(temp, sizeof temp, fpr) == NULL) break;

        strcpy(prog[i].id, strtok(temp,",¥n"));
        strcpy(prog[i].name, strtok(NULL,",¥n"));
        strcpy(prog[i].class, strtok(NULL,",¥n"));
        strcpy(prog[i].point, strtok(NULL,",¥n"));
        strcpy(prog[i].eval, strtok(NULL,",¥n"));
    }

    dispdata(prog, MAX);

    fclose(fpr);

    return 0;
}
```

```
void dispdata(struct STUDENT data[], int n)
{
    int i;

    printf("   %s", headln);

    for (i=0; i<n; i++)
        printf("%d) %s,%s,%s,%s,%s¥n", i+1,
          data[i].id,data[i].name,data[i].class,data[i].point,data[i].eval);
}
```

【解説】

1. strtok は，第 1 引数で指定した文字列を第 2 引数で指定した文字列中のいずれかの文字で区切られる**字句**（token）の列に分割する。なお，2 回目以降の検索には NULL（空ポインタ）を指定する。

2. ここでは，strcpy を用いて各メンバに字句を複写していく目的で全てのメンバを char 型配列としている。なお，prog[i].point は atoi(prog[i].point) によって数値化することができる。

演習 1

例題 1 のプログラムでは履修科目コードが固定されている。これを第 1 プログラム引数として任意のコードが与えられるようにするプログラムを考える。この空欄 1)　　，2)　　を埋めてソースプログラムを完成させ，テキストエディタで入力して，ex14-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

```
/* ex14-1.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 5

int main(int argc, char *argv[])
{
    struct STUDENT
    {
        char         id[8];
        char      name[21];
        char      class[9];
        unsigned int point;
        char          eval;
    };

    struct STUDENT prog[MAX];
```

```
    int i;
    FILE *fpr;

    if (argc != 1)    )
    {
        printf("利用法：ex14-1 履修科目コード¥n");
        exit(1);
    }

    if (NULL == (fpr = fopen("stprog_ex.csv","w")))
    {
        printf("stprog_ex データファイルが開けません。¥n");
        exit(1);
    }

    fprintf(fpr,"学籍番号,氏名,履修科目コード,点数,評価¥n");

    for (i=0; i<MAX; i++)
    {

        strcpy(prog[i].class, 2)    );

        printf("%d 人目の学籍番号を入力してください：", i+1);
        scanf(" %s", prog[i].id);
        printf("%d 人目の氏名を入力してください：", i+1);
        scanf(" %s", prog[i].name);
        printf("%d 人目の点数を入力してください：", i+1);
        scanf(" %u", &prog[i].point);
        printf("%d 人目の評価を入力してください：", i+1);
        scanf(" %c", &prog[i].eval);

        fprintf(fpr,"%s,%s,%s,%u,%c¥n",
          prog[i].id,prog[i].name,prog[i].class,prog[i].point,prog[i].eval);
    }

    fclose(fpr);

    return 0;
}
```

演習 2（**余裕のある人向け**）

演習 1 のプログラムを改良して，履修科目コード.csv（例えば x1043234.csv）という名前のファイルを開き，データを書き出すプログラムを考える。このソースプログラムをテキストエディタで入力して，ex14-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。(提出するファイルは ex14-2.c の完成版とする)

【ヒント】

1) 文字列を格納する配列を 2 個宣言して（例えば s1，s2），一方には第 1 プログラム引数を複写して格納し，他方には文字列リテラル".csv"を宣言時に初期化して格納する。

```
    char s1[30];
    char s2[]=".csv";

    strcpy(s1, argv[1]);
```

2) 更に文字列連結を行うライブラリ関数 strcat を用いて，後者を前者に連結する。

```
    strcat(s1,s2);
```

提出物：

1) csv ファイル stprog.csv

2) 例題 2 の出力結果をコピーして貼り付けたテキストファイル res14.txt

3) 演習 1 のソースプログラムのファイル ex14-1.c の完成版

4) csv ファイル stprog_ex.csv