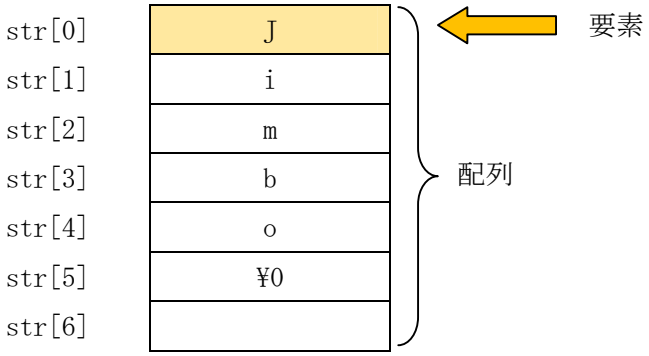
2013 年 4 月 25 日（木）実施

配列

配列

メインメモリ上に，同種のデータ型を有する複数のデータを格納する領域を連続的に確保し，それらの領域を番号付けして，ひとまとまりの対象として扱うものを**配列**と呼ぶ。また，配列としてまとめられた個々のデータを**要素**と呼ぶ。



配列の取り扱いに関して，次のような特徴がある。

1. プログラム中で用いる配列は必ず**宣言**しておく。

2. 配列の宣言時には，**データ型**及び**要素数**を指定する。

   例）char str[7]；　/* 文字（character）のデータ型の配列 str を宣言 ⇒ 要素数は 7 であるが，文字列の**終端**(NULL 文字，'¥0'と表記)に 1 文字分を使うので，扱える文字数は 1 バイトコードの文字の場合 1 要素で 1 文字分となり，6 文字が最大の長さとなる。*/

3. 配列の要素は 0 番から[要素数]－1 番までの添え字を用いて表され，各要素は変数と同様に扱える。 ⇒ 代入，参照は要素毎に行う。

   例）str[0]='a'；　/* 配列 str の 0 番の要素に文字'a'を代入(C 言語では 1 文字は引用符『' '』で囲んで表し，文字列とは別の扱いとなる。)　なお，宣言時に初期化することも可能である。⇒ char str[5]= "abcd"; */

4. 配列名は，0 番の要素がメインメモリ上に占める領域の先頭アドレスを表す。

   例）str と書くと，&str[0]を表す。これは配列 str の 0 番の要素の先頭アドレス(char 型の場合は個々の要素のサイズが 1 バイトなので，0 番の要素のアドレスそのもの)

例題 1（**整数型の配列**）

次のソースプログラムをテキストエディタで入力して，prog3-1.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。

```
/* prog3-1.c */
#include <stdio.h>

int main(void)
{
    int a[3];
    double heikin;

    printf("a[0]の値を整数で入力してください： ");
    scanf("%d", &a[0]);

    printf("a[1]の値を整数で入力してください： ");
    scanf("%d", &a[1]);

    printf("a[2]の値を整数で入力してください： ");
    scanf("%d", &a[2]);

    heikin = (double)(a[0]+a[1]+a[2])/3;

    printf("3 個のデータの平均値は%.1f です。¥n", heikin);

    return 0;
}
```

【解説】

1. double は倍精度の浮動小数点数(実数)のデータ型である。C 言語には float という単精度の浮動小数点数(実数)のデータ型も用意されているが，実用的な数値計算では double 型が必要である。

2. &f の f は第 2 回の教材にあるように，式の値を[-]dddd.dddddd の形の 10 進表現に変換する変換指定子である。なお，%.1f の .1 は精度が小数点以下 1 桁であることを表す。

3. 変数 heikin に対する代入分の等号の右辺で，(double)は**型キャスト**と呼ばれ，それに続くひとまとまりの式(a[0]+a[1]+a[2])の計算結果(ここでは int 型)を double 型に変換する。

＊ 前回の演習のプログラムを実行して確かめられるように，C 言語では int 型のデータを int 型のデータで割ると結果は整数部分しか得られない。double 型のデータを int 型のデータで割れば，結果は double 型となる。

＊＊ 型キャストはその場で一時的にデータ型変換を行うのみで，元の変数のデータ型を変更するものではない。

例題 2（**文字型の配列による文字列の扱い**）

次のソースプログラムをテキストエディタで入力して，prog3-2.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。なお，プログラムを実行する際

に，「神保 雅人」のように途中に１バイトコードの空白文字を入れた場合の動作を確認せよ。また，「神保　雅人」のように途中に２バイトコードの空白文字を入れた場合の動作を確認せよ。

```
/* prog3-2.c */
#include <stdio.h>

int main(void)
{
    char namae[21];

    printf("お名前を入力してください： ");
    scanf("%s", namae);

    printf("今日は、%s さん！¥n", namae);

    return 0;
}
```

【解説】

1. 配列 namae には，１バイトコードの文字であれば 20 文字分，２バイトコードの文字であれば 10 文字分の文字を格納できる。配列の要素数を 21 としているのは，文字列の終端として最後の要素に NULL 文字を置かなければならないからである。

2. 配列名 namae は，配列の 0 番の要素 namae[0]が格納されているメインメモリ上の領域の先頭アドレスを表す。scanf で %s を指定した場合，0 番の要素から文字列の終端までを格納する。

演習１

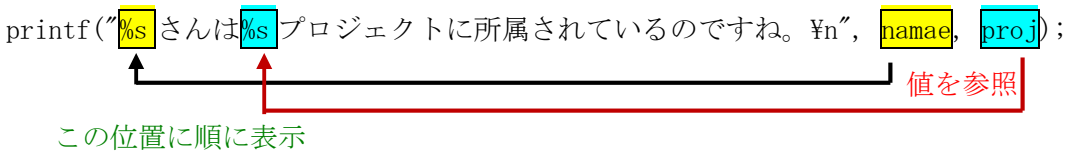
例題１のプログラムを５個のデータに対応して，合計も表示するように変更したものをテキストエディタで入力して，ex3-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

なお，合計を求める変数は，int 型で変数名を goukei とする。

演習２

例題２のプログラムを，名前の入力後に「今日は、○○さん。あなたはどちらのプロジェクトに所属していらっしゃいますか？」と問いかけ，プロジェクト名を入力してもらい，「○○さんは××プロジェクトに所属されているのですね。」と表示する仕様に変更したものをテキストエディタで入力して，ex3-2.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。

なお，プロジェクト名を格納する配列の配列名を proj とする。また，「○○さんは××プロジェクトに所属されているのですね。」と表示するための文は次のようになる。



【参考】（配列によらない文字の並びの扱い）

前回までに学んだ範囲で，１ 文字ずつ ２ 文字分をキーボードから入力してディスプレイに表示するプログラムは次のようになる。

```
#include <stdio.h>

int main(void)
{
    char moji1, moji2;

    printf("最初の１文字を入力：");
    scanf("%c", &moji1);

    rewind(stdin);

    printf("２番目の１文字を入力：");
    scanf("%c", &moji2);

    printf("%c", moji1);
    printf("%c¥n", moji2);

    return 0;
}
```

提出物：

1）例題 1，２ の出力結果をコピーして貼り付けたテキストファイル res3.txt

2）演習 １ 及び演習 ２ のソースプログラムのファイル ex3-1.c 及び ex3-2.c