

# Cプログラミング 入門

— 配列・マクロ (#define) —

早稲田大学

# 今回の目標

- **配列**が適切に使える.
  - 宣言
  - 添字演算子「`[]`」
  - 添字の有効範囲
  - 初期化
  - 多次元配列
  
- **#define** で固定値が使える
  - 定義と使用法

# 配列

## 配列 (int 型の変数を 100 個用意する場合)

```
int Data[100];          /*配列宣言*/
```

- 同じ扱いの変数が複数必要なとき、配列を使う
- 一つ一つの変数は、通し番号 (添え字) をつけて区別
- 添字演算子「`[]`」を使い、中に番号を指定する
- 番号は `0` から始まり、`100-1=99` で終わる。  
Data[0] から Data[99] までの 100 個の変数が用意される
- 代入に関しては、今までと同様

```
Data[0]=3;
```

```
Data[10]=2;
```

# 配列の初期化

- 配列は宣言と同時に初期化することが可能

```
int Data[5]={23,34,45,68,41};
```

- 宣言の後でまとめて値を代入することはできない

```
int Data[5];  
Data = {23,34,45,68,41};
```

- その場合は、次のようにする

```
int Data[5];  
Data[0] = 23 ;  
Data[1] = 34 ;  
...
```

## 配列を使用すると...

添字を指定すれば、通常の変数として使える (data3.c)

```
#include <stdio.h>

int main(void){
    int Data[3];           /* 配列宣言 */
    Data[0]=10;           /* 代入 */
    Data[1]=Data[0]*2;    /* 計算・代入 */
    Data[2]=Data[1]+3;    /* 計算・代入 */
    printf("Data=%d, %d.¥n",Data[0],Data[2]);
    return 0;
}
```

- 最初に int 型 3 つと宣言しているので、使える添字は 0 から 2 まで.

## 例題 1 : 配列使用プログラム例

### 試験の点数を 3 教科入力して合計を求める (array.c)

```
#include <stdio.h>
int main(void){
    int Scores[3],Sum=0, i;                /* 配列宣言 */
    for(i=0;i<3;i++){                      /* for ループ */
        printf("Input score %d:",i);      /* 表示 */
        scanf("%d",&Scores[i]);          /* 入力 */
    }
    for(i=0;i<3;i++){                      /* for ループ */
        Sum += Scores[i];                 /* 加算 */
    }
    printf("Total is %d.¥n",Sum);         /* 表示 */
    return 0;
}
```

- プログラムを写し, 実行せよ
- ファイル名は, array.c とする

## 配列を使用すると...

- 科目数が 100 に増えても、プログラムは「3」を「100」に変えるだけ
- 配列なしでは同じことを何回も書かなければならない
- 配列を使わないと次のようにしなければいけない

```
int Score0,Score1,Score2,Score3,...,Sum=0, i;  
printf("Input score 0:");  
scanf("%d", &Scores0);  
printf("Input score 1:");  
scanf("%d", &Scores1);  
printf("Input score 2:");  
scanf("%d", &Scores2);  
printf("Input score 3:");  
scanf("%d", &Scores3);  
...
```

## マクロ定義 (#define)

- 配列使用例 (array.c) では、同じ「3」が何回も出てきた
- 「3」を何度も書くのは面倒. 打ち間違えると...
- 「#define」を使用して、適当な単語を「3」と定義し、以後は単語を使えば修正が**簡単**. またタイプミスもコンパイル時点で発見できる

### マクロ定義 (#define 文字列1 文字列2)

文字列1を文字列2で置き換える

```
#define SIZE 3
```

- この行以降は、「SIZE」と書くと「3」と扱われる
- マクロ定義には注釈を書く (推奨)
- 定数に対して名前が与えられ、プログラムが読みやすい
- 慣習として、マクロ名は大文字で書く



## 例題 2 : #define 使用プログラム例

### 科目数の 3 を「SIZE」という名前に変える (array2.c)

```
#include <stdio.h>
#define SIZE 3 /*マクロ定義*/

int main(void){
    int Scores[SIZE], Sum=0, i; /* 配列宣言 */
    for(i=0; i<SIZE;i++){ /* for ループ */
        printf("Input score %d:", i); /* 表示 */
        scanf("%d", &Scores[i]); /* 入力 */
    }
    for(i=0; i<SIZE;i++) /* for ループ */
        Sum += Scores[i]; /* 加算 */

    printf("Total is %d,¥n", Sum); /* 表示 */
    return 0;
}
```

- array.c を array2.c とコピー
- マクロ定義を用いて書き換え実行せよ

## 例題 3

### 課題：(ファイル名：score.c)

- 次の計算式で表される 20 人の点数を配列に入れよ

$$i \text{ 番目の人の点数} = (i * 83 + 11) \% 101 \quad (i = 0, 1, \dots, 19)$$

- 次に全員の点数をカンマ区切りで表示せよ
- 最後に最高得点を表示せよ

```
12, 95, 77, ..., 0,  
Highest score is ???.
```

- ファイル名は「score.c」とする

## 例題3のヒント

- 変数 max に配列の先頭の値を代入
- 配列の残りの値を順番に調べ、max より大きければ max を更新する
- メインの部分は以下のようにする

```
max =Scores[0];
for(i=1;…;…){
    if(max <…) …
    printf("Highest score is %3d.¥n ",…);
}
```

- 20 を表すのに「#define」を忘れずに使うこと

# 多次元配列

- 2次元, 3次元などの多次元配列も可能

```
int Scores[50][5];
for (i=0; i<50; i++){
    printf("Student %d:¥n",i);
    for(j=0;j<5;j++){
        printf("Input subject %d:",j);
        scanf("%d",&Scores[i][j]);
    }
}
```

- 配列 Scores は, 50人分の点数を5科目記憶できる
- サイズの大きな配列は, 別の方法でメモリを確保する

# まとめ

- 配列

- 宣言
- 添字演算子「[]」
- 添字の有効範囲
- 初期化
- 多次元配列

- #define

- 定義と使用法