

4M 情報工学Ⅱ

第4回

関数の設計

(配列データの受け渡し)

授業予定

- ~~第1回 コンピュータの基礎2~~
- ~~第2回 情報工学 I の復習~~
- ~~第3回 関数の設計~~
- 第4回 配列データの受け渡し
- 第5回 ファイル書き込み
- 第6回 ファイル読み込み
- 第7回 総合演習
- 第8回 前期中間試験
- 第9回 試験の解答と解説
- 第10回 ポインタの基礎
- 第11回 ポインタと関数
- 第12回 構造体の宣言と利用
- 第13回 構造体の配列的利用
- 第14回 総合演習
- 第15回 前期末試験

返す平均値は
整数ではない
可能性が
あるので

(先週の内容) 課題2解説

キャスト変換
(整数型を実数型へ)

```
1 #include <stdio.h>↓
2
3 double averageof(int x, int y, int z){↓
4     double v;↓
5     ↓
6     v = ((double)x + (double)y + (double)z) / 3;↓
7     ↓
8     return (v);↓
9 }↓
10 ↓
11 int main(void){↓
12     int na, nb, nc;↓
13     ↓
14     puts("Input 3 integer numbers.");↓
15     ↓
16     printf("number 1: "); scanf("%d", &na);↓
17     printf("number 2: "); scanf("%d", &nb);↓
18     printf("number 3: "); scanf("%d", &nc);↓
19     ↓
20     printf("The average of the 3 is... %.2f\n", averageof(na, nb, nc));↓
21     ↓
22     return (0);↓
23 }
```

実数型で
小数点以下第2桁まで
出力

List 6-12

配列の受け渡し

戻り値 (return *) を必要としないので void 型

Int_set 関数用
ローカル変数の宣言
main 関数の変数名
と重複してもよい

仮引数で記述する配列には
[] を付ける

仮引数 (配列, 配列の要素数)

渡された配列の全ての要素を
0 に. この配列は main 関数から
渡されたものなので, main 関数
からも見える.

配列の宣言

main 関数用
ローカル変数の宣言

関数の
呼び出し

実引数
(配列, 配列の要素数)

実引数で記述する配列には
[] を付けない

```
/*
#include <stdio.h>

/* --- 要素数 no の配列 vc の要素に 0
void int_set(int vc[], int no)
{
    int i;
    for (i = 0; i < no; i++)
        vc[i] = 0;
}

int main(void)
{
    int i;
    int ary1[] = {1, 2, 3, 4, 5};
    int ary2[] = {3, 2, 1};

    int_set(ary1, 5);
    int_set(ary2, 3);

    for (i = 0; i < 5; i++) printf("ary1[%d] = %d\n", i, ary1[i]);
    for (i = 0; i < 3; i++) printf("ary2[%d] = %d\n", i, ary2[i]);

    return (0);
}
```

実行結果

```
ary1[0] = 0
ary1[1] = 0
ary1[2] = 0
ary1[3] = 0
ary1[4] = 0
ary2[0] = 0
ary2[1] = 0
ary2[2] = 0
```

課題5

- 演習6-8 (教科書P.133)

要素がnoであるint型の配列vcの並びを逆順にする関数

```
void rev_intary(int vc[], int no)
```

```
{
```



```
}
```

を作成せよ。

本日の課題(その1) 演習6-8(p.133)

```
1 #include <stdio.h>↓  
2 ↓  
3 void rev_intary(int vc[], int no){↓  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13 }↓  
14 ↓  
15 int main(void){↓  
16     int i;↓  
17     int ary[] = {1, 2, 3, 4, 5, 6};↓  
18     ↓  
19     rev_intary(ary, sizeof(ary)/sizeof(ary[0]));↓  
20     ↓  
21     for(i=0; i<sizeof(ary)/sizeof(ary[0]); i++){↓  
22         printf("rev_ary[%d] : %d\n", i, ary[i]);↓  
23     }↓  
24     printf("\n");↓  
25     ↓  
26     return (0);↓  
27 }↓
```

ここを記述しよう

配列全体の大きさ

配列1個の大きさ

配列の要素数

課題6

- 演習6-11 (教科書P.139)

5人の学生の3教科(国語・数学・英語)の点数を二次元配列を用いて処理するプログラムを自作関数を2つ用いて作成せよ。

自作関数1): 科目別の最高点を求める。

自作関数2): 各学生の3教科の平均点を求める。

課題6

```
1 #include <stdio.h>↓
2 ↓
3 int saikou(const int val[3][5], int nb, int no){↓
4 ↓
5 ↓
6 ↓
7 ↓
8     return(max);↓
9 }↓
10 ↓
11 double heikin(const int val[3][5], int nb, int no){↓
12 ↓
13 ↓
14 ↓
15 ↓
16     return(average);↓
17 }↓
18 ↓
19 int main(void){↓
20     int    i, j, sta1;↓
21     double sta2;↓
22     int    score[3][5];↓
23 ↓
24     printf("5人の点数を入力して下さい。¥n");↓
25     for(i=0; i<5; i++){↓
26         printf("[%d] 国語: ", i+1);    scanf("%d", &score[0][i]);↓
27         printf("    数学: ");        scanf("%d", &score[1][i]);↓
28         printf("    英語: ");        scanf("%d", &score[2][i]);↓
29     }↓
30 ↓
31     printf("各科目別の最高点は¥n");↓
32     for(j=0; j<3; j++){↓
33 ↓
34         sta1=saikou(score, j, 5);↓
35 ↓
36         switch (j){↓
37             case 0:    printf("国語=%d¥n", sta1);    break;↓
38             case 1:    printf("数学=%d¥n", sta1);    break;↓
39             case 2:    printf("英語=%d¥n", sta1);    break;↓
40         }↓
41     }↓
42 ↓
43     printf("各学生の3教科の平均点は¥n");↓
44     for(i=0; i<5; i++){↓
45 ↓
46         sta2=heikin(score, 3, i);↓
47 ↓
48         printf("[%d]人目の平均点=%.2f¥n", i+1, sta2);↓
49     }↓
50     return(0);↓
51 }↓
52 [EOF]
```