

# 4M 情報工学Ⅱ

## 第10回

### ポインタの基礎

# 授業予定

- ~~• 第1回 情報工学 I の復習~~
- ~~• 第2回 数値データの受け渡し~~
- ~~• 第3回 配列データの受け渡し~~
- ~~• 第4回 関数の設計・総合演習~~
- ~~• 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

# 間違いプログラム

## List 10-1

```
/*
 二つの整数の和と差を求める (間違い)
*/

#include <stdio.h>

/*--- n1とn2の和・差をsumとdiffに格納 (間違い) ---*/
void sum_diff(int n1, int n2, int sum, int diff)
{
    sum = n1 + n2;
    diff = (n1 > n2) ? n1 - n2 : n2 - n1;
}

int main(void)
{
    int na, nb;
    int wa = 0, sa = 0;

    puts("二つの整数を入力してください。");
    printf("整数A : "); scanf("%d", &na);
    printf("整数B : "); scanf("%d", &nb);

    sum_diff(na, nb, wa, sa);

    printf("和は%dです。 \n差は%dです。 \n", wa, sa);

    return (0);
}
```

main関数から値をもらう  
(もらうのみ⇒一方通行)

計算結果がmain関数  
へ反映されない

### 実行例

```
二つの整数を入力してください。
整数A : 57
整数B : 21
和は0です。
差は0です。
```

関数へ値を渡す

# なぜ間違いか？

- main関数から関数sum\_diffを呼び出すときに、引数na, nb, wa, saの**値**がコピーされる。
  - ⇒ このコピーは**一方通行!!**  
(自作関数内で計算された値は返せない)
- この問題を解決するには
  - ⇒ 記憶領域の**アドレス(番地)**を引き渡す

**これが「ポインタ」だ!!**

# 正解プログラム

## List 10-7

```
/*
 二つの整数の和と差を求める
*/

#include <stdio.h>

/*--- n1とn2の和・差をsumとdiffに格納 ---*/
void sum_diff(int n1, int n2, int *sum, int *diff)
{
  *sum = n1 + n2;
  *diff = (n1 > n2) ? n1 - n2 : n2 - n1;
}

int main(void)
{
  int na, nb;
  int wa = 0, sa = 0;

  puts("二つの整数を入力してください。");
  printf("整数A : "); scanf("%d", &na);
  printf("整数B : "); scanf("%d", &nb);

  sum_diff(na, nb, &wa, &sa);

  printf("和は%dです。 \n差は%dです。 \n", wa, sa);

  return (0);
}
```

main関数から記憶領域  
のアドレスをもらう

計算結果を格納した  
アドレスをmain関数へ返す

### 実行例

```
二つの整数を入力してください。
整数A : 57
整数B : 21
和は78です。
差は36です。
```

関数へアドレスを渡す

# ポインタを利用するには

- 記憶領域のアドレスを格納する**変数**として宣言する

⇒ ポインタ型変数宣言

「変数名の前に\*をつける」

int \*sum (格納する値がint型の場合)

- 間接演算子\*

⇒ ポインタに\*をつけるとオブジェクト(値)を表す

\*sum+=n1+n2

- ポインタ型変数のアドレスを取り出す

⇒ アドレス演算子&

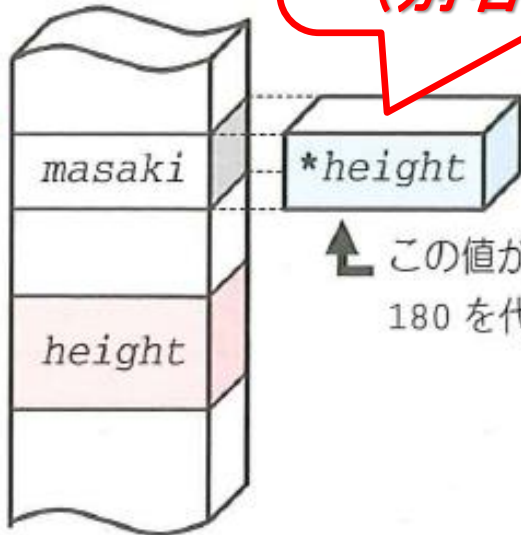
「変数名の前に&をつける」

&sum

# ポインタ渡し

\*height は masakiの  
**エイリアス**  
(別名/あだ名)

106番地



↑ この値が180未満であれば、  
180を代入

```
int main(void)
{
    int masaki = 179;
    :
    hiroko (&masaki)
}
```

アドレスが渡される

106番地

```
void hiroko(int *height)
{
    if (*height < 180)
        *height = 180;
}
```

int型ポインタとして宣言

**間接演算子\***  
そのポインタが指す  
オブジェクト(値)を示す

# scanf 関数とポインタ

- なぜ scanf (“%d”, &na) と  
変数名の前に&をつけるのか？

⇒scanf関数を用意するオブジェクト(変数)  
に対して値を格納しなければならない。

⇒変数の値そのものをもらっても仕方ない。

⇒ポインタが示す変数に対して、キーボード  
から入力した値を格納する



# 課題13

- 演習10-2 (教科書P.239)

三つのint 型整数を昇順に並べ替える関数

```
void sort3(int *n1, int *n2, int *n3){ }
```

を作成せよ。

# 課題13

```
1 #include <stdio.h>↓
2 ↓
3 void sort3(int *n1, int *n2, int *n3)↓
4 {↓
5 ↓
6 ↓
7 ↓
8 ↓
9 }↓
10 ↓
11 int main (void)↓
12 {↓
13     int na, nb, nc;↓
14 ↓
15     puts("3つの整数を入力して下さい");↓
16     printf("整数A:"); scanf("%d", &na);↓
17     printf("整数B:"); scanf("%d", &nb);↓
18     printf("整数C:"); scanf("%d", &nc);↓
19     ↓
20     sort3(     ここを考えよう     );↓
21 ↓
22     puts("昇順に並べ替えました");↓
23     printf("整数Aは%dです。¥n", na);↓
24     printf("整数Bは%dです。¥n", nb);↓
25     printf("整数Cは%dです。¥n", nc);↓
26 ↓
27     return(0);↓
28 }↓
29 [EOF]
```

# 課題14

- List10-9 の間違い修正 (教科書P.239)

`/*二つの実数値を交換する*/`

教科書のプログラムリストには間違いがあります。  
うまく実行できるように修正して下さい。

```
/*
 二つの実数値を交換する（間違い）
*/

#include <stdio.h>

/*---- nx・nyが指すオブジェクトの値を交換 ----*/
void swap(int *nx, int *ny)
{
    int temp = *nx;
    *nx = *ny;
    *ny = temp;
}

int main(void)
{
    double dx, dy;

    puts("二つの実数を入力してください。");
    printf("実数X：");    scanf("%lf", &dx);
    printf("実数Y：");    scanf("%lf", &dy);

    swap(&dx, &dy);

    puts("これらの値を交換しました。");
    printf("実数Xは%fです。\\n", dx);
    printf("実数Yは%fです。\\n", dy);

    return (0);
}
```

間違いを修正して下さい

#### 実行結果例

二つの整数を入力してください。  
実数X：53.5  
実数Y：21.68  
これらの値を交換しました。  
実数Xは9980.450456です。  
実数Yは50.568782です。