

C言語7

ポインタ

アドレスを出力

```
#include<stdio.h>
int main(void){

    int a;
    a=5;

    printf("変数aの値は%dです。¥n", a);
    printf("変数aのアドレスは%pです。¥n", &a);

    a=15;

    printf("変数aの値は%dです。¥n", a);
    printf("変数aのアドレスは%pです。¥n", &a); //アドレスは変わらない!

    return 0;
}
```

【実行結果】

変数aの値は5です。
変数aのアドレスは0038FAC4です。
変数aの値は15です。
変数aのアドレスは0038FAC4です。

【注】

アドレスの値は実行環境やプログラムの実行状況により異なります。

ポインタ

ポインタにアドレスを格納

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

```
int main(void){
```

```
    int a;
```

```
    int *pA;
```

```
    a = 5;
```

```
    pA = &a;    //aのアドレスをpAに格納
```

```
    printf("変数aの値は%dです。¥n", a);
```

```
    printf("変数aのアドレスは%pです¥n", &a);
```

```
    printf("ポインタpAの値は、%pです。¥n", pA);
```

```
    return 0;
```

```
}
```

【実行結果】

変数aの値は5です。

変数aのアドレスは0022FE84です
ポインタpAの値は、0022FE84です。

間接参照

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

```
int main(void){
```

```
    int a;
```

```
    int *pA;
```

```
    a = 5;
```

```
    pA = &a;
```

```
    printf("変数aの値は%dです。¥n", a);
```

```
    printf("変数aのアドレスは%pです。¥n", &a);
```

```
    printf("*pAの値は%dです。¥n", *pA);
```

```
    return 0;
```

```
}
```

【実行結果】

変数aの値は5です。

変数aのアドレスは0032FB9Cです。

*pAの値は5です。

ポインタに別のアドレスを代入

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

```
int main(void){
```

```
    int a, b;
```

```
    int *pA;
```

```
    a = 5;
```

```
    b = 10;
```

```
    pA = &a; //変数aのアドレスを代入
```

```
    printf("変数aの値は%dです。¥n", a);
```

```
    printf("ポインタpAの値は%pです。¥n", pA);
```

```
    printf("pAの値は%dです。¥n", *pA);
```

```
    pA = &b; //変数bのアドレスを代入
```

```
    printf("変数bの値は%dです。¥n", b);
```

```
    printf("ポインタpAの値は%pに変更されました。¥n", pA);
```

```
    printf("pAの値は%dです。¥n", *pA);
```

```
    return 0;
```

```
}
```

【実行結果】

変数aの値は5です。

ポインタpAの値は001FF784です。

pAの値は5です。

変数bの値は10です。

ポインタpAの値は001FF77Cに変更されました。

pAの値は10です。

ポインタを使って変数の値を変更する

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

```
int main(void){
```

```
    int a;
```

```
    int *pA;
```

```
    a = 5;
```

```
    pA = &a;
```

```
    printf("変数aの値は%dです。¥n", a);
```

```
    *pA = 50;
```

```
    printf("*pAに50を代入しました。¥n");
```

```
    printf("変数aの値は%dです。¥n", a);
```

```
    return 0;
```

```
}
```

【実行結果】

変数aの値は5です。

*pAに50を代入しました。

変数aの値は50です。

引数とポインタ1 2つの値の交換(swap)

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

```
/* swap関数の宣言*/
```

```
void swap(int *px, int *py);
```

```
int main(void){
```

```
    int num1 = 5;
```

```
    int num2 = 10;
```

```
    printf("変数num1の値は%dです。¥n", num1);
```

```
    printf("変数num2の値は%dです。¥n", num2);
```

```
    printf("変数num1とnum2の値を交換します。¥n", num2);
```

```
    swap(&num1, &num2);
```

```
    printf("変数num1の値は%dです。¥n", num1);
```

```
    printf("変数num2の値は%dです。¥n", num2);
```

```
    return 0;
```

```
}
```

```
/* swap1関数の定義*/
```

```
void swap(int *px, int *py){
```

```
    int tmp;
```

```
    tmp = *px;
```

```
    *px = *py;
```

```
    *py = tmp;
```

```
}
```

【実行結果】

変数num1の値は5です。

変数num2の値は10です。

変数num1とnum2の値を交換します。

変数num1の値は10です。

変数num2の値は5です。

引数とポインタ2 並べ替え(sort)

```
#include<stdio.h>

void sort(int *na, int *nb){
    int tmp;
    if(*na>*nb){
        tmp=*na;
        *na = *nb;
        *nb = tmp;
    }
}

int main(void){

    int n1, n2;

    printf("2つの整数を入力して下さい。¥n");
    printf("整数1 :"); scanf("%d", &n1);
    printf("整数2 :"); scanf("%d", &n2);

    sort(&n1, &n2);

    printf("2つの数を小さい順に並べます。¥n");
    printf("%d, %d¥n", n1, n2);

    return 0;
}
```

【実行例1】

2つの整数を入力して下さい。

整数1:12

整数2:9

2つの数を小さい順に並べます。

9, 12

【実行例2】

2つの整数を入力して下さい。

整数1:15

整数2:23

2つの数を小さい順に並べます。

15, 23

引数とポインタ3 割り算の商と余り

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

```
void division(int a, int b,int *c, int *d){  
    *c =a/b;  
    *d =a%b;  
}
```

```
int main(void){
```

```
    int na, nb;
```

```
    int q=0; //商(quotient)
```

```
    int r=0; //余り(remainder)
```

```
    printf("2つの整数を入力して下さい。¥n");
```

```
    printf("整数1 :"); scanf("%d", &na);
```

```
    printf("整数2 :"); scanf("%d", &nb);
```

```
    division(na, nb, &q, &r);
```

```
    printf("%d/%d = %d 余り %d¥n", na, nb, q, r);
```

```
    return 0;
```

```
}
```

【実行例】

2つの整数を入力して下さい。

整数1 : 355

整数2 : 113

355/113 = 3 余り 16

引数とポインタ4 分数の掛け算

```
#include<stdio.h>
int gcd(int n, int r){
    int tmp;
    while(r>0){
        tmp=r;
        r=n%r;
        n=tmp;
    }
    return tmp;
}

void reduceFrac(int a, int b, int **k, int **l){
    int c = gcd(a, b);
    **k = a/c;
    **l = b/c;
}

void addFrac(int s, int t, int u, int v, int *na, int *nb){
    int p, q;
    p = s*u;
    q = t*v;
    reduceFrac(p, q, &na, &nb);
}

int main(void){
    int n1, n2, n3, n4;
    int num=0, den=0;
    while(1){
        printf("分数1を入力して下さい。¥n");
        printf("分子:"); scanf("%d", &n1);
        printf("分母:"); scanf("%d", &n2);
        printf("分数2を入力して下さい。¥n");
        printf("分子:"); scanf("%d", &n3);
        printf("分母:"); scanf("%d", &n4);

        addFrac(n1, n2, n3, n4, &num, &den);

        if(den!=1)
            printf("%d/%d × %d/%d=%d/%d¥n¥n", n1, n2, n3, n4, num, den);
        else
            printf("%d/%d × %d/%d=%d¥n¥n", n1, n2, n3, n4, num);
    }

    return 0;
}
```

【実行例】

分数1を入力して下さい。

分子:3

分母:7

分数2を入力して下さい。

分子:14

分母:5

$3/7 \times 14/5 = 6/5$

分数1を入力して下さい。

分子:3

分母:13

分数2を入力して下さい。

分子:26

分母:3

$3/13 \times 26/3 = 2$

分数1を入力して下さい。

分子:2

分母:9

分数2を入力して下さい。

分子:27

分母:6

$2/9 \times 27/6 = 1$