

# ポインタと配列その1 (数学購読)

## 実習で利用するサンプルプログラム

### prog5-1.c (ポインタとアドレス演算)

```
#include <stdio.h>
int main( char argc, char *argv[] )
{
    int x=1, y=2;
    int *ip, *iq;

    printf( "1: x=%d y=%d\n", x, y);
    ip = &x;
    iq = ip;

    y = *ip;
    printf( "2: x=%d y=%d\n", x, y);

    *iq = 7;
    x = *iq+*ip+1;
    printf( "3: x=%d y=%d\n", x, y);

    exit(0);
}
```

### prog5-2.c ( Call by values and Call by address )

```
#include <stdio.h>

void swap_value( double a, double b)
```

```

{
    double temp;
    temp = a;
    a = b;
    b = temp;
    return;
}

void swap_address( double *a, double *b )
{
    double temp;
    temp = *a;
    *a = *b;
    *b = temp;
    return;
}

int main( int argc, char *argv[] )
{
    double x = 1, y=2;

    printf( "1: x = %lf, y = %lf\n", x, y );

    swap_value( x, y );
    printf( "2: x = %lf, y = %lf\n", x, y );

    swap_address( &x, &y );
    printf( "3: x = %lf, y = %lf\n", x, y );

    exit(0);
}

```

### prog5-3.c ( 配列とポインタ )

```

#include <stdio.h>

int main( int argc, char *argv[] )
{

```

```

int i;
double a[10] = {10,9,8,7,6,5,4,3,2,1}, b;
double *pa;

/* まず pa に配列 a[0] のアドレスを代入 */
pa = &a[0];
/* したがって、pa がポイントする先の値は a[0] */
printf( "*pa = %lf\n", *pa );

/*
   次に pa に配列の名前 a を代入これは配列の
   格納されているメモリの先頭アドレスに等しい.
*/
pa = a;
/* だから、pa のポイントする先にあるのは、a[0] */
printf( "*pa = %lf\n", *pa );

/*
   ポインタに値を i 加えると、アドレスは一つずつ
   配列を進む。よって *(pa+i) = a[i] となる.
*/
for(i=0;i<10;i++){
    printf( "(pa+%d) = %lf\n", i, *(pa+i) );
}

/* 最後に以下の計算の意味を考えてみましょう */
pa = &a[2]+3;
b = *(a+2)+3;
printf( "*pa = %lf, b = %lf\n", *pa, b );

exit(0);
}

```