C 言語講習会　第七回

・ポインタ

・アドレスとポインタ

　アドレスとは、プログラムや変数などのデータの存在する場所を指すもののことです。いわば、メモリの住所です。＆をつけることにより、その変数のアドレスを確認することができます。

例 1

```
#include <stdio.h>

int main()
{
   char c;
   int i;
   double d, e;

   printf(“変数 c のアドレスは%p です¥n”, &c);
   printf(“変数 i のアドレスは%p です¥n”, &i);
   printf(“変数 d のアドレスは%p です¥n”, &d);
   printf(“変数 e のアドレスは%p です¥n”, &e);

   return 0;
}
```

環境によって実行結果は違いますが、「0012FE03」などのように１６進数で表示されているのが変数の住所であるアドレスです。
また、アドレス自体も変数として扱うことができます。アドレスを扱うためには、int 型ならば int *pi のように、*をつけることによって実行することができます。このアドレス専用の変数のことをポインタと呼びます。この場合、pi は「int 型へのポインタ」となります。

ポインタを使うには、

```
int a;
```

int *pa;
pa = &a;
とします。こうすることにより、a のアドレスが pa に代入されました。a の住所を pa に記録したと考えると分かりやすいと思います。

例 2

```
#include <stdio.h>

int main()
{
   int i;
   int *pa; //ポインタの宣言

   pa =   &i //ポインタに a のアドレスを入力

   printf(“適当な整数値を入力してください------”);
   scanf(“%d”, &a);
   printf(“変数 a に%d が代入されました。¥n”, a);
   printf(“変数 a のアドレスは%p です。¥n”, &a);
   printf(“変数 a を指しているポインタは pa です。¥n”);
   printf(“pa の値は%d です。¥n”, *pa);

   return 0;
}
```

実行結果は次のようになります。

```
適当な整数値を入力してください------10
変数 a に 10 が代入されました。
変数 a のアドレスは 0012FEE4 です。
変数 a を指しているポインタは pa です。
pa の値は 10 です。
```

これを確認して、アドレスやポインタのおおよその使い方をつかんでおいてください。

ちなみに、*pa = 5;のように、ポインタに直接値を入れることによっても、a = 5;とするのと同じ結果を得ることができます。ややこしいですが、*pa と a は同じ(値)、pa と&a は同じ(アドレス)と覚えてください。

・ポインタと引数

前回の講習会で行った関数作成ですが、変数 a と b の値を入れ替える関数を作ろうと考えると最初は大抵このように考えると思います。

例 3

```
#include <stdio.h>

void swap(int, int);

int main()
{
   int a, b;

   a = 10;
   b = 20;

   swap(a, b);     //a と b の値を入れ替える

   printf("a = %d, b = %d¥n", a, b);

   return 0;
}

void swap(int a, int b)
{
   int c;

   c = b;
   b = a;
```

```
    a = c;

    return;
}
```

---

swap 関数により a と b の値を入れ替えようと思ったのですが、これを実行してみると
a = 10, b = 20
という結果になり入れ替えが失敗していることが分かります。これは何が原因なのでしょうか。
実は、関数を呼び出した際に swap(a, b);としましたが、これは a と b の「値」を swap 関数に渡しているだけであって、swap 関数と main 関数の a,b は別物です。従って、swap 関数のなかで変数の値を変更しても main 関数にはなんの変化も起こしません。このような関数呼び出しを「値呼び出し」と言います。これは、関数呼び出しによって不用意に変数の値が変えられることを防ぐ安全装置の役割を果たしています。逆に、関数内で仮引数の値を変更すれば呼び出し元の実引数も変更される、という方法を「参照呼び出し」と言います。これは、関数の引数に変数のアドレスを渡すことにより実行することができます。
それでは、先ほどのプログラムを正しく書き直してみましょう。

例 3(正しく動く)

```
#include <stdio.h>

void swap(int *, int *);

int main()
{
    int a, b;

    a = 10;
    b = 20;

    swap(&a, &b);     //a と b の値を入れ替える

    printf("a = %d, b = %d¥n", a, b);
```

```
    return 0;
}

void swap(int *x, int *y)
{
    int z;

    z = *y;
    *y = *x;
    *x = z;

    return;
}
```

これで a と b の値が入れ替わっているのが確認できると思います。

・ポインタのポインタ

ポインタ自体も変数です。従って、ポインタの住所、つまりポインタのポインタも存在します

例 4

```
#include <stdio.h>

int main()
{
    int a;
    int *p;
    int **pp;

    p = &a;
    pp = &p;

    **pp = 10;
```

```
    printf(“a = %d,    *p = %d,    **pp = %d¥n”,    a, *p, **pp);

    return 0;
}
```

この場合、**pp は　*(*pp)　と考えることができ、*pp は p のアドレスを指しているため、最終的には*(p)と同義になります。