

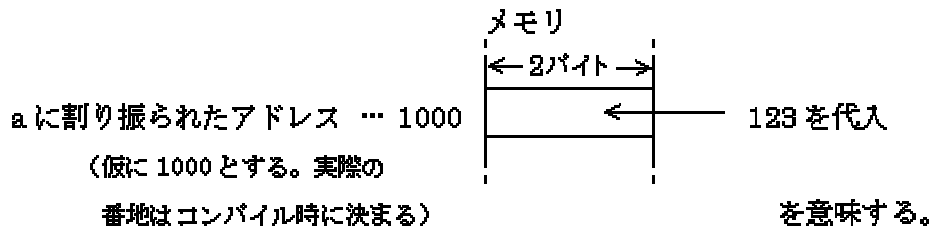
第7回 ポインタ

今回は、ポインタです。ポインタは、関数をまたいで変数の値をいじるときに使います。人によってはゲームで使う機会がないかもしれませんが、ソフトゼミAのラストなので気をいれて頑張りましょう。

・アドレス

アドレスとは……

`int a = 123;` は実際には下図のように、「メモリ上のあるアドレス、つまり番地（下図では1000番地）に変数 `a` としての領域を確保し、その領域に 123 を格納する」ということです。



この時、

`a` 変数 `a` の値 123 を返す
`&a` 変数 `a` のアドレス 1000 を返す

ポインタについて

C 言語では、変数のアドレスを数値として取り扱うことができます。これがポインタです。多分、言葉だけでは、わかりづらいと思うので宣言の仕方と例を載せます。

使い方：(格納したい変数の型) *(任意の変数名); (注意: *をつけ忘れると普通の変数)

(例) `int *p;` (`int` 型のアドレスをしまっておける `p` というポインタ)

アドレスをポインタに代入して利用する

```
int *p;
```

```
int a;
```

が宣言されている状態で

```
p=&a; (ポインタ=アドレス)
```

とやると `p` というポインタに `a` のアドレスが入ります。

ポインタにアドレスが入っているときに

*p(アドレスを格納したポインタ)とやると格納してるアドレスの変数と同じ扱いになります。

```
#include<stdio.h>
int main(void){
    int a=10;
    int *p;
    p=&a; //ポインタ p に変数 a のアドレスを入れる
    printf("a=%d\n 変数 a のアドレス=%d\n",a,&a);
    printf("p=%d\n",p); //この場合アドレスが返ってくる
    printf("p の中に入ってる値=%d\n",*p);
    return(0);
}
```

実行結果

a=10

変数 a のアドレス=1245008

p=1245008

p の中に入ってる値=10

(アドレスの値は、ほとんどの場合、実行結果とは同じになりませんが気にしなくて大丈夫です)

これだけでは、どのようなときに使うべきかわからないと思うので、意味のあるポインタの使い方を紹介します。

```
#include<stdio.h>
void swap1(int x,int y){
    int i;
    i=x;
    x=y;
    y=i;
}
void swap2(int *x,int *y){
    int i;
    i=*x;
    *x=*y;
    *y=i;
}
void swap3(int x,int y){
    int i;
    i=x;
    x=y;
    y=i;
    printf("%d %d\n",x,y);
}
```

```
int main(void){
    int x,y;
    x=10;y=20;
    printf("%d %d\n",x,y);
    swap1(x,y);
    printf("%d %d\n",x,y);
    swap2(&x,&y);
    printf("%d %d\n",x,y);
    x=10;y=20;
    swap3(x,y);
    return(0);
}
```

実行結果

10 20

10 20

20 10

20 10

これで、ソフトゼミAは、終了です。お疲れ様でした。