2.ポインタ

ポインタとは、アドレスを格納する変数のことを言う。

アドレスとは、メモリ上に与えられた番号（番地）のこと。基本16進数。

変数を宣言すると、メモリ上のどこかのアドレスに変数が生成される。

int x = 10;

アドレス演算子の&を使うことで、アドレスを取得できる。

アドレスにアクセスすれば、変数の値も取得できる。

int\* ptr = &x;　　// アドレスを取得0x00ffddaa

ポインタにアクセスして値を取得することを参照外しという。

ポインタから値を取得するときは間接演算子の＊を使う。  
int val = \*ptr;　　 // アドレスの中身（今回はXのアドレスの10）

実際ポインタをどうやって使うのか？

int setPara (int\* ptr){ 　// ポインタを受け取る（アドレスをやり取りする）

\* ptr = 10; // アドレスの示す中身を変更  
}

int main()  
{

　int a = 0;

　setPara(&a); // aのアドレスを渡す

}

配列名はその配列の先頭要素へのポインタとして解釈される。

int a[10] = {0,1,2,3,4,5,6,7,8,9};

int\* ptr = a;

ptr[0] // a[0] = 0 というようにアクセスできる。

ポインタに整数iを加算すると、そのポインタが指す要素のi個前後の要素を指すポインタとなる。

ptr++; // a[1]の場所を示すようになる

動的に生成、破棄するにはnew, delete演算子を使う。

配列を破棄するのはdelete[]演算子。

動的に生成されたオブジェクトには名前がないので、ポインタを用いてアクセスする。

int\* ptr = int new;

delete ptr;

newでの生成に失敗したらbad\_alloc例外が投げられるのでそれを補足して例外処理を行うと良い。

int \*a = new int[9999999999999999999999999999999];

・参照渡しとポインタの違いは？？？

基本的には参照渡しのほうを使うようにすると良いが、ポインタを使わなければいけない場合もあったりする。

参照渡し

int setPara(int& ref){  
 ref = 10;

}

int main(){

int x;

setPara(x);   
}

ポインタ

int setPara(int\* ptr){

\*ptr = 10;  
}

int main(){

int x;

setPara(&x);  
}

参照は初期化していないといけない、nullを設定できないなど制限もある。