

第3回 if文・switch文

これまでのプログラムは、main関数から始まり、上から順番に逐次実行していき、main関数の終了とともに終わるプログラムでした。今回は今までのプログラムに条件分岐を付け加えるための文法である、「if文」と「switch文」を学んでいきます。

○if文

プログラムを分岐させる上で、最も基本的な制御文が「if文」です。英語と同じように「もしも～なら」の意味です。if文の構造は次のようになります。

```
if( 条件式 ){
    条件を満たす場合に実行する処理 ;
}
```

「条件式」は、例えば「変数 a の値が 10 である時」、「変数 b の値と変数 c の値が同じである時」といったものです。これらの条件式は C 言語においては通常は「関係演算子」を用いて記述します。関係演算子は以下の通りです。

関係演算子	意味
<	左辺が右辺より小さい
>	左辺が右辺より大きい
<=	左辺が右辺以下
>=	左辺が右辺以上
==	左辺と右辺が同値関係
!=	左辺と右辺が異なる
&&	左辺かつ右辺
	左辺または右辺

上の表の内、==、!=は前回少しでてきましたね。関係演算子は演算子の一種で、左辺と右辺の関係についての演算子を特にこう呼びます。=が一つだと同値関係ではなく代入式になってしまうので注意してください。

if文は条件式が満たされたとき、{}の内側の処理を行います。この{}がない場合、if文の直後の文のみ(;が一度出てくるまで)が条件式が満たされたとき実行されます。条件式が満たされなかった場合、{}内または直後の文を無視して先に進みます。それでは例を見てみましょう。

```
#include<stdio.h>
int main( void ){
    int x ; // int 型の変数 x を宣言
    printf( "数を入力してください。¥n" ) ; // プログラム実行者に明示
    scanf( "%d" , &x ) ; // 数値入力処理
    if( x > 50 ){ // 条件文！！
        // 条件が満たされる時のみ実行
        printf( "%d は 50 より大きいです。¥n" , x ) ;
    }
    // 条件分岐終了。以下は条件に関わらず実行される
    return ( 0 ) ;
}
```

上記のようにすれば、50以上の数字が入力された際に、{}内の文が実行されます。条件式が満たされる時、満たされない時をそれぞれ「真」、「偽」と呼びます。&&、||については、例えば「a が 20 かつ b が 15 のとき」ならば、「a == 20 && b == 15」と記述します。||についても同様です。ただし、この二つは&&が||よりも先に計算されます。

○else

さきほどのプログラムでは、50 以下の数が入力された場合、そのまま終了してしまいます。ここで登場するのが else 文です。else 文を使えば、条件式を満たさなかった場合の処理について記述することができます。

```
if ( x > 50 ) { // 条件文！！
    // 条件が満たされる時のみ実行
    printf( "%dは50より大きいです。¥n" , x ) ;
} else {
    // 条件が満たされなかった時のみ実行
    printf( "%dは50以下です。¥n" , x ) ;
}
```

上記のように、if 文の後に else { 条件を満たさない場合に実行する処理 ; } という風にすればよいだけです。ちなみに、if 文同様 1 文のみの場合は {} は省略することができます。しかし、いずれの場合もプログラムが見づらくなるものですので、できるだけ {} でくった方が良いでしょう。

○else if 文

else if 文を使えば、最初の if 文の条件式を満たさなかった場合の中で、さらに条件分岐を加えることができます。実際には else if で一組のキーワードというワケではありませんが、あまり気にする必要はありません。

```
if ( x > 50 ) { // 条件文！！
    // 条件が満たされる時のみ実行
    printf( "%dは50より大きいです。¥n" , x ) ;
} else if ( x > 0 ) {
    printf( "%dは0より大きいです¥n" , x ) ;
} else {
    printf( "%dは0以下です¥n" , x ) ;
}
```

上記のように記述します。今回も条件の合う場合に直後の文のみ実行したい場合は {} は省略可能です。else if は複数連ねていくことが可能ですが、途中または最初で条件式を満たした時点で以降に連ねた else、else if は全て実行されないことに注意しましょう。

○switch 文

複数の分岐を作る場合、より効率的に記述することができるのが switch 文です。以下のように記述します。

```
switch( 整数変数名 ) {
    case 定数整数:
        定数に一致する場合に実行する処理 ;
        break;
    case 定数整数:
        定数に一致する場合に実行する処理 ;
        break;
    case 定数整数:
        (略)
    default:
        どの定数にも一致しない場合に実行する処理 ;
        break;
}
```

() 内に整数変数名を記述すると、その値が定数整数と一致した時に、そこから break; するまでの処理を実行します。Break; するまでの処理を実行し続けるので、break; を記述しないと、その下の条件を満たした場合に実行するはずの処理まで実行してしまいますが、これをうまく利用したプログラムを書くこともできます。また、定数整数は定数整数なので、変数は書くことができません。また、switch 文は整数しか扱えないことにも注意しましょう。

○練習問題

1. scanf関数で整数を入力させ、0～59 なら F、60～69 なら C、70～79 なら B、80～89 なら A、90～100 なら S と表示するプログラムを作れ。
2. 3つの数値を scanf 関数をつかって入力させ、最も小さい数値を表示するプログラムを作れ。
3. 1番での問題の条件分岐を switch 文を使って書け。
ヒント: 整数を整数で割った場合、小数点以下は切り捨てられる。