

制御文(if 文)

やっと制御文の話をしてします。

変数 `a` の値が何だったら～～という処理をする
そうでなかったら～～という処理をする

というマイコン・パソコンを使う上での一番のメリットですね。

とりあえずソースコードを書きます。

```
#include <stdio.h>
int main (){
    int a=0;

    if(a==1){
        printf("a は 1 です。¥n");
    }
    printf("おわり。");
}
```

実行結果



「`a は 1 です。`」は表示されなかったですね。
これは、`if` 文のおかげですね。

```
if(a==1){
    ...a が 1 の時だけ処理される内容...
}
```

このように、`if` 文のあとの `{}` の中の内容は `()` 内の条件が成立しているときだけ、今回は「`a が 1 の時`」だけ実行されます。

```
a ==1
```

「 == 」は同値の意味です。「(左辺)の内容が(右辺)と同じ」ですね。

これは同値がどうかをチェックするためのモノなので、単体で使うことはできません。

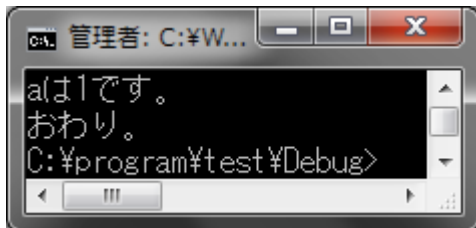
a=1;などと混同しないようにしましょう。

今度は a の内容を 1 の状態で実行してみましょう。

```
#include <stdio.h>
int main (){
    int a=1;

    if(a==1){
        printf("a は 1 です。 ¥n");
    }
    printf("おわり。 ");
}
```

実行結果

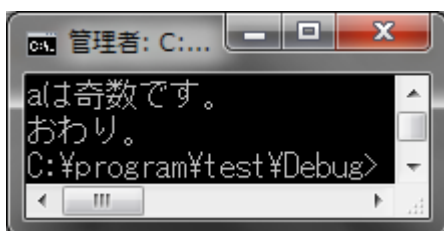


今度は a=1 なので、「a は 1 です」が表示されましたね。

if 文はこのようにも使えます。

```
#include <stdio.h>
int main (){
    int a=1;
    if(a%2==1) printf("a は奇数です。 ¥n");
    if(a%2==0) printf("a は偶数です。 ¥n");
    printf("おわり。 ");
}
```

実行結果



```
if(a%2==1) printf("a は奇数です。¥n");
```

今度は{}がありません。

この場合、次の 1 命令分が、条件に当てはまったときの処理になります。

ここでは「a は奇数です」と表示する処理ですね。

```
if(a%2==1) printf("a は奇数です。¥n");
```

```
if(a%2==0) printf("a は偶数です。¥n");
```

この描き方に無駄があると思いませんか？

2 で割ったあまりは 0 か 1 しかありませんよね？

そこで、こう書き換えましょう。

```
if(a%2==1) printf("a は奇数です。¥n");
```

```
else printf("a は偶数です。¥n");
```

「else」は if 文のあとに付けることができ、if 文で条件で当てはまらなかった場合にされる処理を記述します。

次の例ならわかりやすいはず。

```
#include <stdio.h>
int main (){
    int a=0;

    if(a==1){
        printf("a は 1 です。");
    }
    else{
        printf("a は 1 ではありません。");
    }
}
```

実行結果



通称「else-if」といわれるものもあります。

```
#include <stdio.h>
int main (){
    int a=1;

    if(a==1) printf("a は 1 です。");
    else if(a==2) printf("a は 2 です。");
    else printf("a は 1 でも 2 でもありません。");
}
```

実行結果

int a=1;のとき



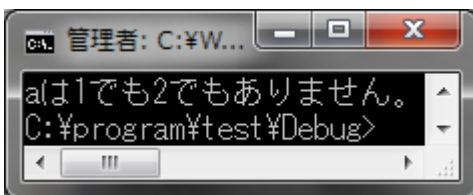
A screenshot of a Windows command prompt window. The title bar reads 'C:\> 管理者: C:\¥...'. The command prompt shows the output 'aは1です。' followed by the prompt 'C:\¥program¥test¥Debug>'. The window has standard Windows window controls (minimize, maximize, close).

int a=2;のとき



A screenshot of a Windows command prompt window. The title bar reads 'C:\> 管理者: C:\¥...'. The command prompt shows the output 'aは2です。' followed by the prompt 'C:\¥program¥test¥Debug>'. The window has standard Windows window controls (minimize, maximize, close).

int a=3; など 1 でも 2 でもないとき



A screenshot of a Windows command prompt window. The title bar reads 'C:\> 管理者: C:\¥W...'. The command prompt shows the output 'aは1でも2でもありません。' followed by the prompt 'C:\¥program¥test¥Debug>'. The window has standard Windows window controls (minimize, maximize, close).

```
if(a==1) /* a==1 の時の処理*/
else if(a==2) /* a==2 の時の処理*/
else /* a が 1 でも 2 でもないときの処理*/
```

となります。

if 文の中に if 文を入れることもできます。
条件式だけ書きます。

```
int a=0, b=0;

if(a==0){
    if(b==0){
        printf("a は 0,b も 0");
    }
    else    printf("a は 0 だが、b は 0 以外");
}
else printf("a も b も 0 以外");
```

処理される部分は printf 関数で表示される文章と同じ内容になります。

いきなりかなり難しいですが問題です。

問.次の条件式の場合の実行結果はどうなるでしょう？

```
#include <stdio.h>

int main (){
    int a=0, b=0,c=0;

    if(a==1) c=1000;
    else if(a==0){
        if(b==1){
            a++;
            c++;
        }
        else if(b==0) c=10;
        else if(a==0) c=100;    //②
    }
    if(c==100){
        a=1000;
    }
    if(c=10){
        c=100;
        c++;    //①
        a++;
    }
    printf("a は%d¥nb は%d¥nc は%d",a,b,c); //a,b,c の数値を表示
}
```

ヒント

- ①のように if 文の中で if 分の条件に当てはまらなくなっても処理は継続されます。
- ②の else-if では、もし全ての条件に当てはまらなかった場合はそのまま次に進みます

答.



```
管理者: C:¥...
aは1
bは0
cは101
C:¥program¥test¥Debug>
```

こうなります。

どのように処理されたのかを表します。

処理されていない部分は薄字、それ以外の部分は処理されます。

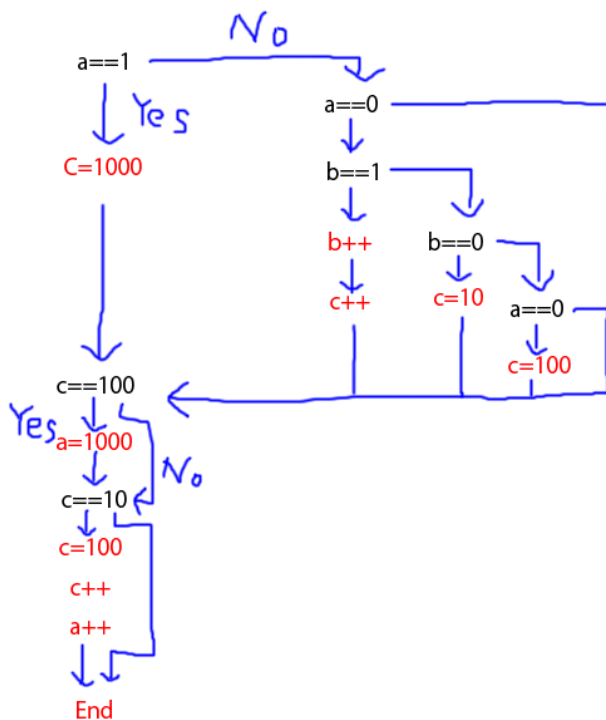
```
#include <stdio.h>

int main (){
    int a=0, b=0,c=0;

    if(a==1) c=1000;
    else if(a==0){
        if(b==1){
            a++;
            c++;
        }
        else if(b==0) c=10;
        else if(a==0) c=100;
    }
    if(c==100){
        a=1000;
    }
    if(c=10){
        c=100;
        c++;
        a++;
    }
    printf("a は%d¥nb は%d¥nc は%d",a,b,c); //a,b,c の数値を表示
}
```

図式化すると、

真下にいったら「条件に当てはまっている」、右にいったら「当てはまっていない」です。



わかりにくいですがこうなっています。

次回は switch-case ・ for 文 ・ while 文の解説をします。