

基礎プログラミング演習 II 教材 (#6)

■ ループ (繰り返し)

□ ループを作る

実験: 右上のプログラムを入力し、結果を予測し、実行して確認して下さい。

```
int count;
count=0;

printf("%d\n", count);
count=count+1;
printf("%d\n", count);
count=count+1;
printf("%d\n", count);
count=count+1;
```

こういった繰り返す処理 (ループ) は、while 文によって実現することができます。

実験: プログラムを右のように修正し、結果を予測し、実行して確認して下さい。このプログラムは無限に繰り返すので、Control - C (Control キーを押しながら C キーを押す) によって強制停止させてください。

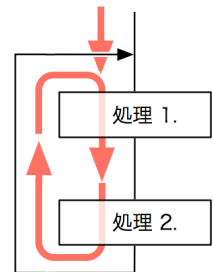
```
int count;
count=0;

while( 1 ) {
    printf("%d\n", count);
    count=count+1;
}
```

□ while 文 (無限に繰り返す使い方)

右に示すように while 文は後に続くブロック内 (中カッコに囲まれた部分) に書かれた処理を繰り返し実行します。

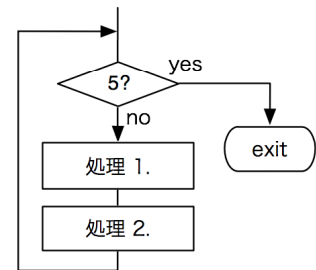
```
while( 1 ) {
    処理1... ;
    処理2... ;
}
```



上で実験したプログラムの動作が納得できますか? (while(1)の 1 の意味は後で説明します)

□ 途中で実行を停止する (if 文との組み合わせ)

しかし無限に繰り返すわけではなく、例えば 5 回繰り返したら止めたい場合があります。今までに学んだ範囲のテクニックで実現するには、どうすれば良いでしょうか?



課題 1.

count が 5 になったら exit() 関数を実行してプログラムを停止するようにプログラムを修正してください。(if 文で判定できますね?)

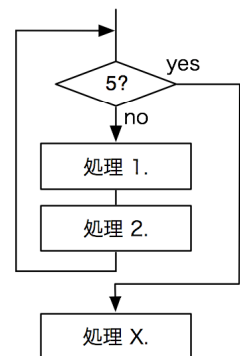
□ 途中でループから抜け出す (break)

しかし常に exit() で終了したいわけではなく、例えば 5 回繰り返したらループから抜けてそれ以降の処理を続行したい場合があります。

この目的のために break 文があります。

break はそれ以降の処理をせず、ループを抜け出します。つまり while () に続くブロック (中カッコ) の終わりまで飛ばすのです。

```
while( 1 ) {
    if( ... ) break;
    処理1... ;
    処理2... ;
}
処理x... ;
(続く...)
```



```
0
1
2
3
4
finished.
```

課題 2.

右の例のように最後に一行終了したことを意味するメッセージを表示するようにプログラムを修正してください。

□ 繰り返し処理の考え方

★教科書 p.75 以降の例 5.1、5.2、図 5.1 を参照。

二乗を求める処理には繰り返しの構造があり、変数を用いて処理を記述すれば、ループによって処理できることが納得できますか？

p.77 の「このように考えてみると…」の параグラフの説明が理解できるでしょうか。

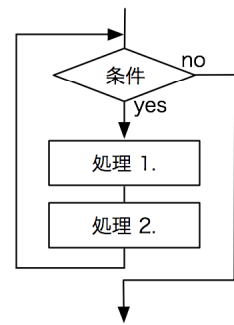
□ while (繰り返し判定条件付き)

p.77 のすぐ次のパラグラフにあるように、while には繰り返し条件の判定を直接記述できます。

★教科書 p.78 文法 5.1 を参照してください。

p.79 の例 5.3 のプログラムの内容が理解できますか？

注意すべきは while の条件式は「繰り返し処理を行うか否か」であって、先に試みた「繰り返しを中断するか否か」ではありません。
(論理が逆です。右の図の条件部の yes / no も逆転しています。)



課題 3.

break を使わず、while() の条件指定によってループを終了させてください。ただし 0~4 までを表示するのではなく、scanf()によって入力された数から 0 までカウントダウンさせるように。

■ 宿題 (今回の提出は課題 4. と 5. だけで結構です)

ループを学ぶときに、どのようなループの方法があるか、while 文の文法規則はどういったものか、といったことに集中しすぎてはいけません。重要なのはいかにループするか、ではなく、ループを用いて何をするか、です。

□ 課題 4.

1 から 50 までの数について、3 あるいは 5 の倍数だけを選んで表示させてください。
(つまり 3, 5, 6, 9, 10, 12, 15 ~ 45, 48, 50 です。)

□ 課題 5.

1 から 10 までの合計を求めるプログラムを作成してください。(55 になるはずですね)

考え方：

- ・はじめに合計をゼロとする
- ・1 回目のループの際には合計に 1 を加え、二回目のループの際には合計に 2 を加え…
- ・ループが終わった時の合計が結果として使えるのではないかと？
- ・まず 10 回ループするプログラムを作り、そこに上記の処理手順を組み込もう

押さえて欲しいポイント：目的の処理を実現する手順を考えることが、プログラミングそのものです。機能を確認することが目的の課題と、考え方を学ぶ課題の違いを意識してください。磨くべき能力を見失わないように。