

## 2-8 : 制御文—while 文、do~while 文

### ○while 文

while 文とは指定した条件を満たしている限り、延々とその命令を繰り返し実行し続けさせる制御文です。逆に、最初から条件を満たしていない場合この制御文は動作しません。

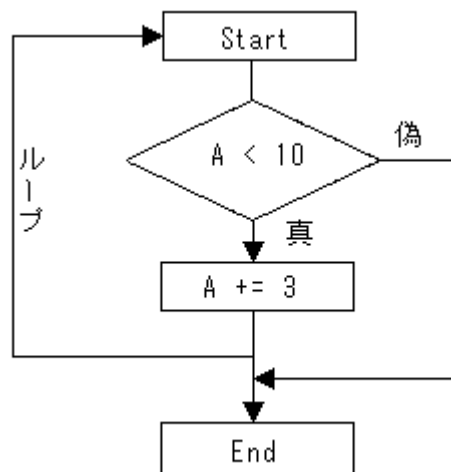
### <一般式>

```
while( 条件式 ) { 命令 ; }
```

### <例>

```
int A;           //変数 A の宣言  
A = 0;          //A の初期値を 0 とする  
while(A<10) {A+=3;} //A が 10 未満である限り 3 を足し続ける
```

これを実行すると、変数 A は while 内で 3 ずつ大きくなっていき、12 になった時このループから出てきます。



・ while(1) {命令文;}

こう書いた文は特殊で、コンピュータは命令を無限に繰り返します。命令を無限に繰り返させてはパソコンがフリーズしてしまいますので while 文の中の命令に工夫が必要です。

```
int B;
B = 0;
while(1) {
    if(B > 10) {break;}      //if(B > 10)break;でも可
    B += 3;
}
```

break は switch 文でも使われていますが、「その break を括っている制御文の一つ外に強制的に脱出する命令」です。break を置けば無限ループからも出ることができるわけです。

・ for 文を while 文に書き直す

for 文を同じくループを取り扱う物として while 文に書き直すことができます。

for 文

```
X = 0;
for(int i = 0; i < 10; i++) {
    X++;
}
```

while 文

```
X = 0;
int i = 0;
while(i < 10) {
    i++;
    X++;
}
```

どちらの命令も同じ結果を出します。

<問題>

「int 型で初期値 32 の変数 X を、2 で割り続けて 4 にする」というプログラムを作ってください。

○do~while 文

while 文では条件の確認を先に行いますが do~while 文というのは、まず一度命令を実行してから条件を確認し、その後も同じ命令を続けるかで分岐する制御文です。

<一般式>

```
do{ 命令文 ;}while( 条件式 );
```

<例 1 >

```
int Y;  
Y = 1;  
do{Y *= 2}while(Y < 3);
```

この場合 Y は 4 になり出てきます。

<例 2 >

```
int Z;  
Z = 10;  
do{Y *= 2}while(Y < 3);
```

この場合、初期値で既に Z は条件から外れていますが、それに関わらず命令は一度実行されて Z は 20 となります。

<問題>

int KS[50]; を宣言しておき、do~while 文を使って KS[0]~KS[49] 全てに 1 を代入してください。

