

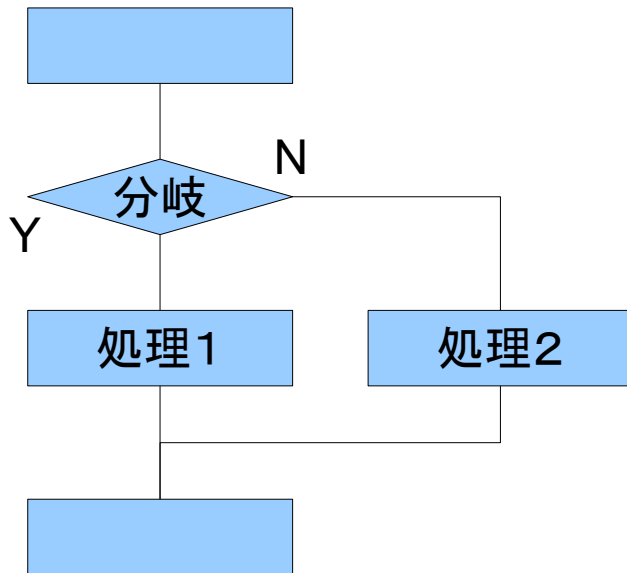
初級C言語講座Vol.2

復習と補足

- 基本的なプログラムの形
mainの中にプログラムを書く
- 変数の取り扱い
mainの先頭で宣言
- 整数の取り扱い
int型変数を使う
- 小数点以下の値の取り扱い
float型変数を使う
式中での変数の型によって計算結果が異なる
- 入出力の方法
printfとscanf

if文

- 分岐を行える制御文

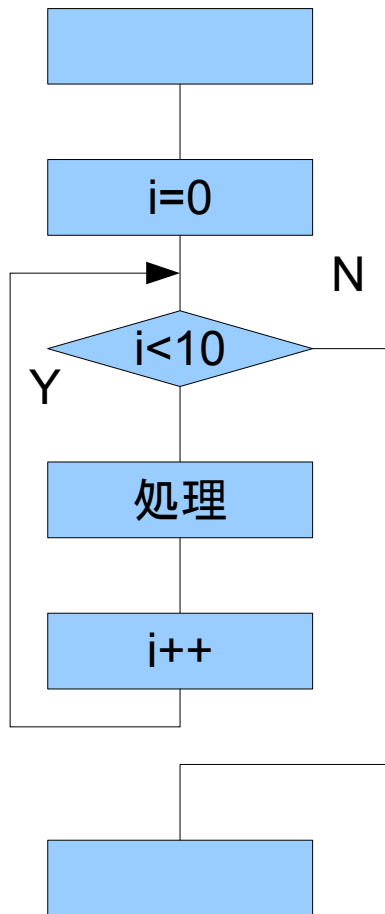


```
#include <stdio.h>

int main(void)
{
    int a;
    printf("値=>");
    scanf("%d", &a);
    if(a>10)
    {
        printf("10より大きい値です\n");
    }
    else
    {
        printf("10以下の値です\n");
    }
    return 0;
}
```

for文

- 繰り返しを行う文

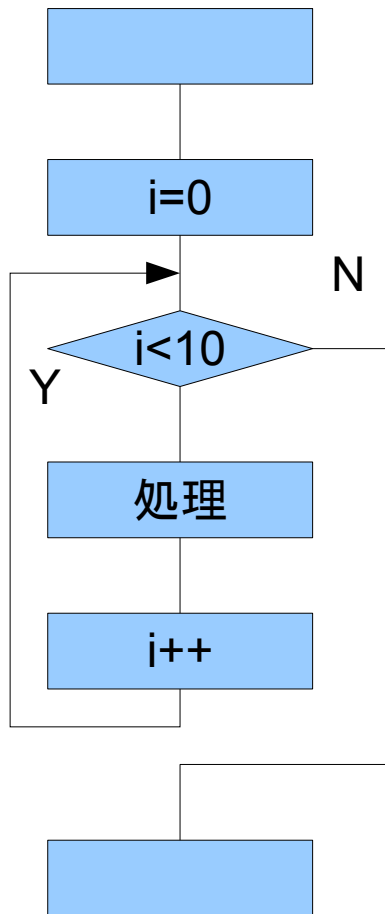


```
#include <stdio.h>

int main(void)
{
    int i;
    for (i=0; i<10; i++)
    {
        printf("%d\n", i);
    }
    return 0;
}
```

while文

- 繰り返しを行う文その2



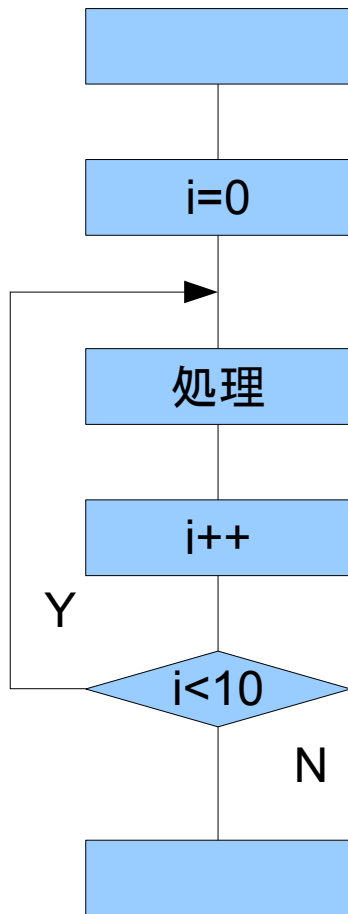
```
#include <stdio.h>

int main(void)
{
    int i;

    i=0;
    while (i<10)
    {
        printf("%d¥n", i);
        i++;
    }
    return 0;
}
```

do while文

- 繰り返しを行う文その3



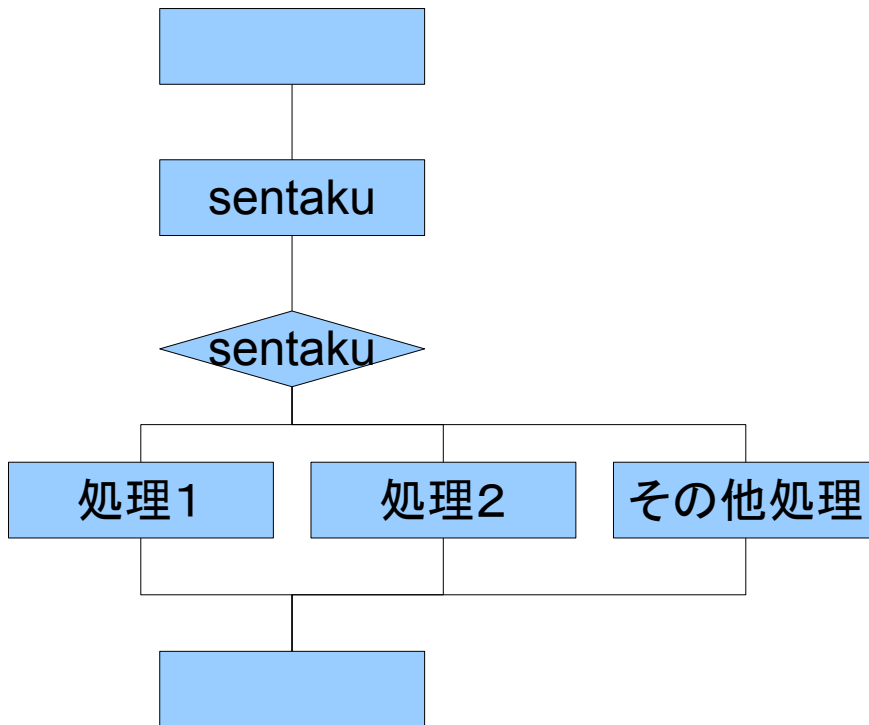
```
#include <stdio.h>

int main(void)
{
    int i;

    i=0;
    do {
        printf("%d\n", i);
        i++;
    } while (i<10);
    return 0;
}
```

switch文

- 式のとる値によって動作を選べる



```
#include <stdio.h>
```

```
int main(void)  
{
```

```
    int sentaku, a, b, c;
```

```
    printf("メニュー¥n");
```

```
    printf("1. 乗算2. 除算3. 加算4. 減算5. AND¥n");
```

```
    scanf("%d", &sentaku);
```

```
    printf("値をつ入力");
```

```
    scanf("%d %d", &a, &b);
```

```
    switch(sentaku)
```

```
    {
```

```
    case 1:
```

```
        c=a*b;
```

```
        break;
```

```
    case 2:
```

```
        c=a/b;
```

```
        break;
```

```
    case 3:
```

```
        c=a+b;
```

```
        break;
```

```
    case 4:
```

```
        c=a-b;
```

```
        break;
```

```
    case 5:
```

```
        c=a & b;
```

```
        break;
```

```
    default:
```

```
        printf("不正な選択肢です¥n");
```

```
    }
```

```
    printf("演算結果¥n%d¥n", c);
```

```
    return 0;
```

```
}
```

条件演算子

- ifなどの、条件を記述する際に利用する演算子達

数学での記号	C言語での記号	意味・補足説明
\leq	<code><=</code>	<code>=></code> ではだめ
\geq	<code>>=</code>	同上
$<$	<code><</code>	
$>$	<code>></code>	
$=$	<code>==</code>	
\neq	<code>!=</code>	
$($ 、 $\{$ 、 $[$	$($	中・大括弧は別な用途に利用するので、通常 <small>の</small> 括弧のみで記述する。
$)$ 、 $\}$ 、 $]$	$)$	

AND	<code>&&</code>	「AかつB」などを示す。「かつ」
OR	<code> </code>	「AまたはB」などを示す。「または」

課題1

- 値(n)を入力し、1～nの値について、次を同時に表示するプログラムを作れ。
 - すべての整数の合計
 - 奇数のみの合計
 - 偶数のみの合計

解答例

解答

```
#include<stdio.h>
int main (void)
{
    int sum, sumK, sumG, n, i;

    //入力
    printf("n=>");
    scanf("%d", &n);

    //計算用変数の初期化
    sum=0;
    sumK=0;
    sumG=0;
    // 1～nまでの繰り返し
    for (i=1; i<=n; i=i+1)
    {
        sum=sum+i;           //すべての整数の合計計算
        if (i%2==0)
        {
            sumG=sumG+i;    //偶数の合計計算
        }
        else
        {
            sumK=sumK+i;    //奇数の合計計算
        }
    }

    //表示
    printf("1～%dの¥n総和=%d¥n偶数の和%d¥n奇数の和%d¥n", n, sum, sumK, sumG);

    return 0;
}
```

課題2

- 2つの値(a,b)を入力して、次の値を出力するプログラムを作れ
 - 最大公約数
 - 最小公倍数
- アルゴリズムとして、「ユークリッドの互除法」を用いる。手順は次のとおり。
 - $a \geq b > 0$ とする。 $a < b$ なら入れ替え、 $a, b = 0$ なら終了。
 - b が a を割り切れる(余りが0)なら、 b が最大公約数。
 - a を b で割った余りを新たな b とし、 a は前の b (先の計算で代入する前の b)の値を代入して、ひとつ上に戻る。
 - 最小公倍数は、最初の a, b の積を最大公約数で割ったもの。

解答例

解答

```
#include<stdio.h>
int main (void)
{
    int a,b,r,ab;
    printf("a=>");          //2数の入力
    scanf("%d",&a);
    printf("b=>");
    scanf("%d",&b);

    //不正な入力値のチェック
    if(a<=0 || b<=0)
    {
        printf("入力値が不正です。¥n");
        return 0;          //処理を終了する。微妙だが今回はこれで。
    }
    //入れ替え関係処理
    if(a<b)
    {
        r=a;              //rにaを一度保存
        a=b;              //aにbを代入
        b=r;              //待避していたaの値(=rに保持)を代入
    }

    //計算
    ab=a*b;              //最小公倍数で利用する値を先に計算
    do {
        r=a%b;          //割った余りをrに保存し、判定と次の計算に利用
        a=b;           //入れ替えを行っておく
        b=r;           //同上
    } while(r!=0);      //余りをチェックし、割り切れているかどうかを判定条件とする。
    printf("最大公約数%d¥n", a);    //入れ替え済みであるので、bではなくaとする。
    printf("最小公倍数%d¥n", ab/a); //先に計算した値と定義による

    return 0;
}
```