C言語講座Vol.2

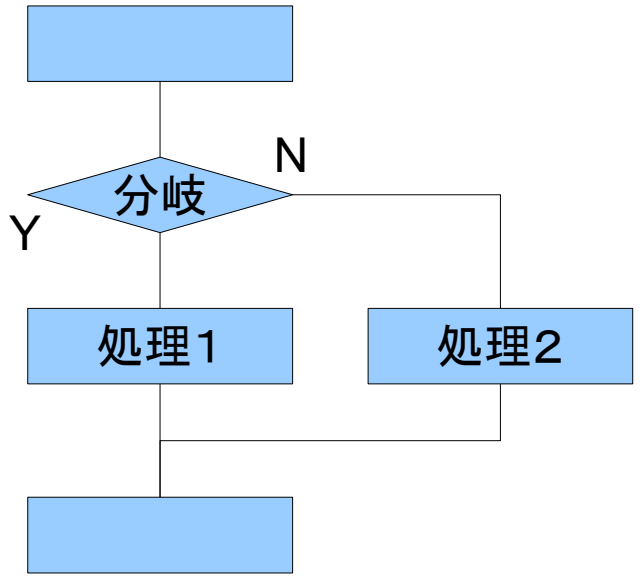
2009年5月22日　CISC

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

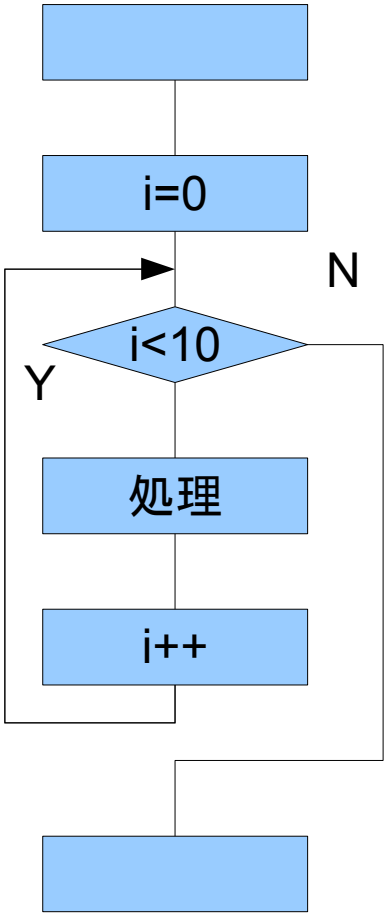
- 分岐を行える制御文




```c
#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

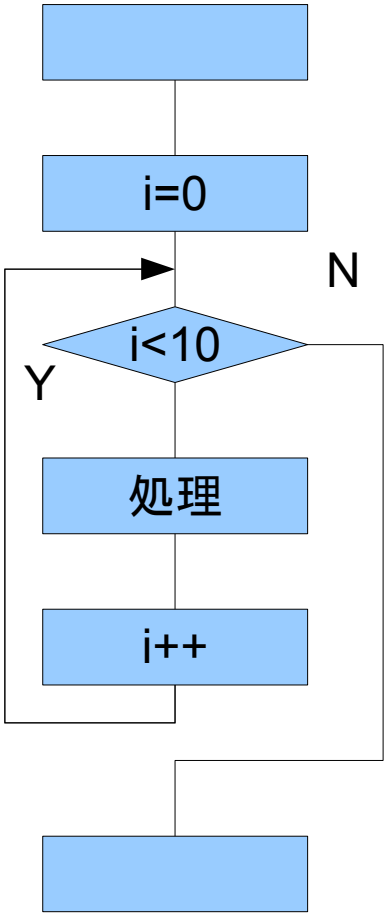



```c
#include <stdio.h>

int main(void)
{
    int i;
    for(i=0;i<10;i=i+1)
    {
        printf("%d¥n",i);
    }
    return 0;
}
```

変数iの値をまず0に設定し、
iの値を調べて、条件式にしたがって、
10より小さいか調べる。
次の行を実行した後で、iを1増加させ、
条件判断の直前へ戻る。

while文

- 繰り返しを行う文その２

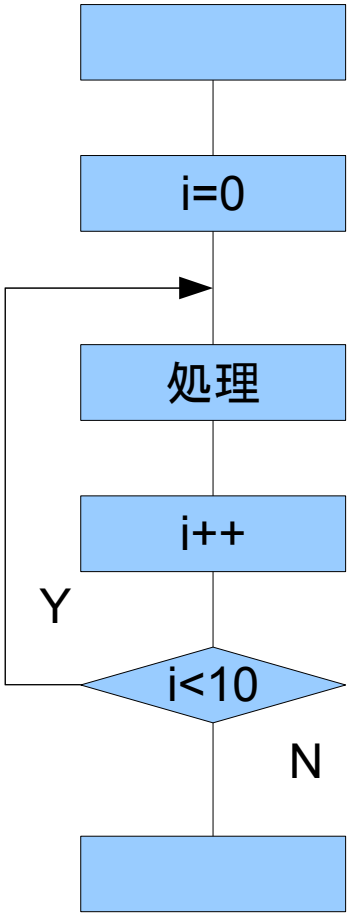



```c
#include <stdio.h>

int main(void)
{
    int i;

    i=0;
    while(i<10)
    {
        printf("%d¥n", i);
        i=i+1;
    }
    return 0;
}
```

構造としては、forとそっくり。
ただし、while文自体は、条件判断のみを行うので、
実際には、カウントが要らない処理などに適する。

do while文

- 繰り返しを行う文その3

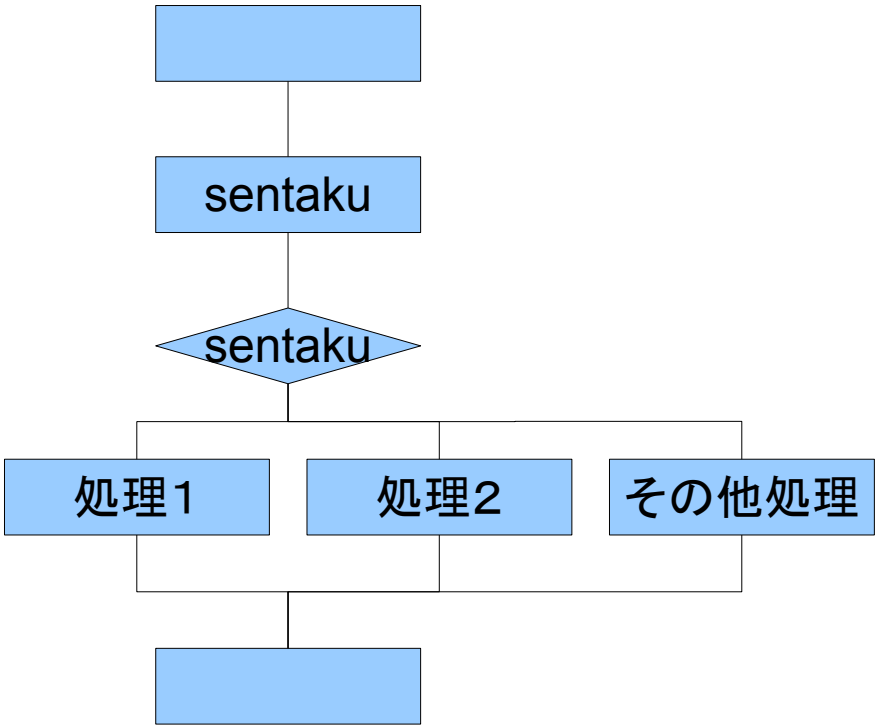



```c
#include <stdio.h>

int main(void)
{
    int i;

    i=0;
    do {
        printf("%d¥n",i);
        i=i+1;
    } while(i<10);
    return 0;
}
```

基本的に、while文と同じ。違いは、条件判断を、処理を実行した後に行うか否かという点。

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
≦	<=	=>ではだめ
≧	>=	同上
＜	<	
＞	>	
＝	==	
≠	!=	
（、｛、［	(	中・大括弧は別な用途に利用するので、通常の括弧のみで記述する。
）、｝、］	)	

AND	&&	「AかつB」などを示す。　「かつ」
OR	||	「AまたはB」などを示す。「または」

変数の型

- C言語では、変数の型により、扱える値が異なる。

型の名前	扱うデータ	サイズ(Byte)	扱える範囲
Int	整数	4(昔は2)	-2147483648 ～ 2147483647
float	小数	4	
long	整数	4	-2147483648 ～ 2147483647
double	小数	8	
char	文字（のコード）	1	-128 ～ 127

問題1

- 変数nに値を入力し、1～nまで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
  - 自分の誕生日の1桁目+2桁目の倍数の合計
    (21日生まれ:2+1=3の倍数)

発展

  - 素数の合計

問題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≧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。

次回の話題

- データについて学ぶ
- 配列
  同じ種類の変数を連続的に使用可能。
  番号でアクセスできるようになります。
- 文字列登場！
  C言語における文字列とは？
- 構造体？
  複数種類の変数をまとめて扱う。

解答例

解答

```c
#include<stdio.h>
int main (void)
{
	int sum1,sum2,n,i;

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

	//計算用変数の初期化
	sum1=0;
	sum2=0;
	//１～ｎまでの繰り返し
	for(i=1;i<=n;i=i+1)
	{
		sum1=sum1+i;			//すべての整数の合計計算
		if(i%3==0)
		{
			sum2=sum2+i;	//特定倍数の合計計算
		}
	}

	//表示
	printf("1～%dの¥n総和=%d¥n3の倍数の和%d¥n",n,sum1,sum2);

	return 0;
}
```

※二桁の整数から片方の値を取得する方法など、
　教えてもいないのに実装してくれた人が居たようです。今後に期待でしょうか。

解答例

解答

```c
#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{			//一回は計算が必要であるから、do whileを用いる。
		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;
}
```