第5回条件による分岐

1

今回の目標

- 式、文（単文、ブロック）を理解する。
- 条件分岐の仕組みを理解する。
- 関係演算子、論理演算子の効果を理解する。

☆ $ax + c = 0$ の型の方程式の解を求めるプログラムを作成する。

2

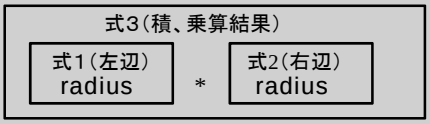
式と単文

式：定数、変数、関数呼び出し
　と、それらを演算子で結合したもの。

式の例

```
3.14
age=20
radius * radius
area =3.14 * radius * radius
```

この文字列には、3種類の式がある。



単文：式＋’;’

単文の例

C言語では、セミコロンが重要な役割を果たす。

```
3.14;
age=20;
radius * radius;
area =  3.14 * radius * radius;
```


文と複文

文：単文、複文、・・・

単文

```
*****;
```

複文
（ブロック）

文をならべて、
中括弧で囲んだもの。

```
{
    ****;
    *********;
}
```

中の文は、
一段字下げして
左端をそろえる事。
（スタイル規則参照）

中括弧の後にはセミコロンはつけない。

C言語のプログラムは、
このような文（単文、複文、・・・）から構成される。

if文

C言語で、条件式によって、
文を選択して実行する文

書式

```
if(条件式)
    {
        選択実行部分1
    }
```

選択実行部分は、ブロックにする。

条件式が **真** なら選択実行部分1を **実行する**。

条件式が **偽** なら選択実行部分1を **実行しない**。

5

式と真偽

C言語には真と偽を表す専用の型はなく、
int型の値で代用する。

真：1（0以外）

偽：0

if文の条件式には、
真偽を表す整数型の式（論理式）
を書く。
（スタイル規則参照）

必ず、中括弧を書く。
（スタイル規則参照）

```
int   bool;

bool=1;
if(bool)
   {
       …
   }
```

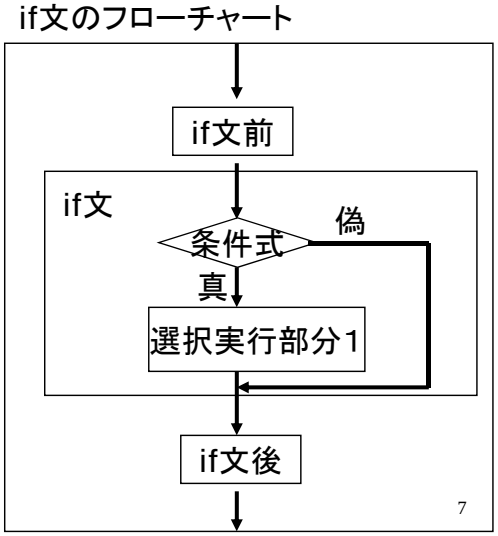
この例では、
この部分は実行
されます。

6

if文の動作1（フローチャート）

書式

```
if(条件式)
    {
        選択実行部分1
    }
```



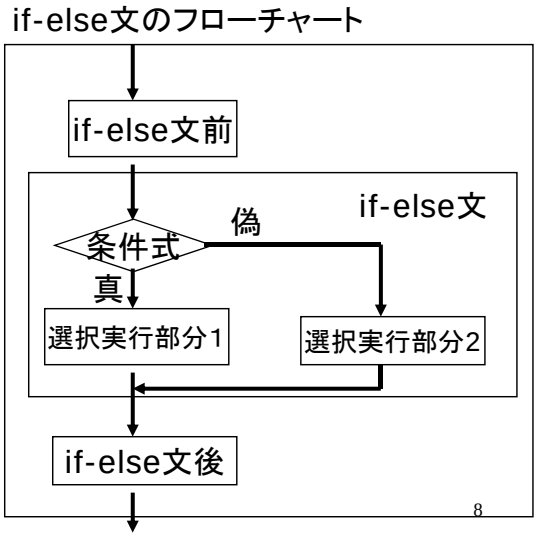
7

if-else文

条件によって2つの文のどちらかを選択して実行する。

書式

```
if(条件式)
    {
        選択実行部分1
    }
else
    {
        選択実行部分2
    }
```



8

条件の表現1（関係演算子）

a = = b	a がbと等しい時に真
a ! = b	a がbと等しくない時に真
a < b	a がbより真に小さいとき真
a > b	a がbより真に大きいとき真
a < = b	a がb以下のとき真
a > = b	a がb以上のとき真

関係演算子を使った式は、真偽値を表すint型の値を返す。
本演習では、関係演算子を使った式は論理式として扱い、
算術式とは明確に区別すること。

9

条件の表現2（論理演算子）

演算子	演算の意味	演算結果
!A	A の否定 (NOT A)	A が真のとき !A は偽。 A が偽のとき !A は真。
A && B	AかつB (A AND B)	A と B が共に真のとき A&&B は真。 それ以外のときは偽。
A || B	AまたはB (A OR B)	A と B が共に偽のとき A||B は偽。 それ以外のときは真。

論理演算子の被演算項（AやB）
は論理式だけを記述する。
よって、AやBは真偽値を表す。

10

式1 ＆＆ 式2 ＆＆ ・・・＆＆式n

式1から式nまですべてが真なら真。
それ以外の場合は偽。

式1 ｜｜ 式2 ｜｜ ・・・｜｜式n

式1から式nまですべてが偽なら偽。
それ以外の場合は真。

AND と OR が混在するような複雑な論理式を用いるときには、括弧をうまく用いて表現する。

11

プログラム例1（条件分岐の練習）

```
/* bunki.c      条件分岐の練習( コメント省略 ) */
#include<stdio.h>
int  main()
{
        int a;
        int b;
        int c;
        printf("3つの整数を入力して下さい¥n");
        printf("a=");
        scanf("%d", &a);
        printf("b=");
        scanf("%d", &b);
        printf("c=");
        scanf("%d", &c);
/*  次のページに続く  */
```

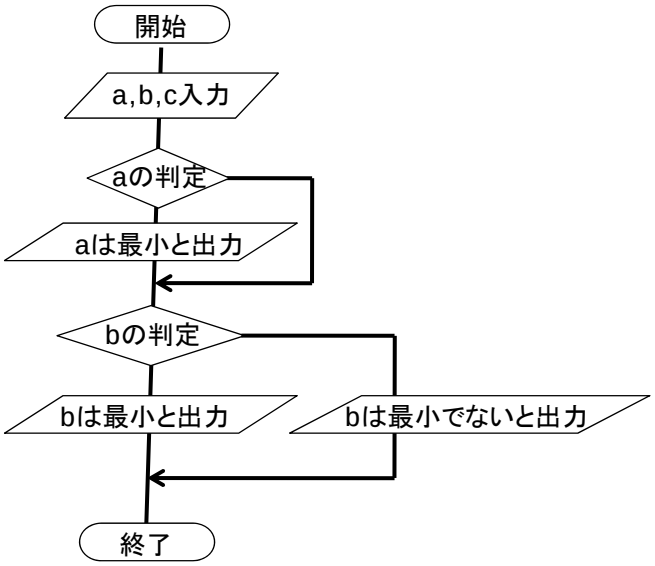
12

```
/*続き*/
        if((a<=b) && (a<=c))
            {
                printf("aは最小です。¥n");
            }

        if((b<=a) && (b<=c))
           {
                printf("bは最小です。¥n");
           }
        else
            {
                printf("bは最小ではありません。¥n");
            }
        return 0;
}
```

13

プログラムbunki.cのフローチャート



14

多分岐（elseの後にif文）

書式

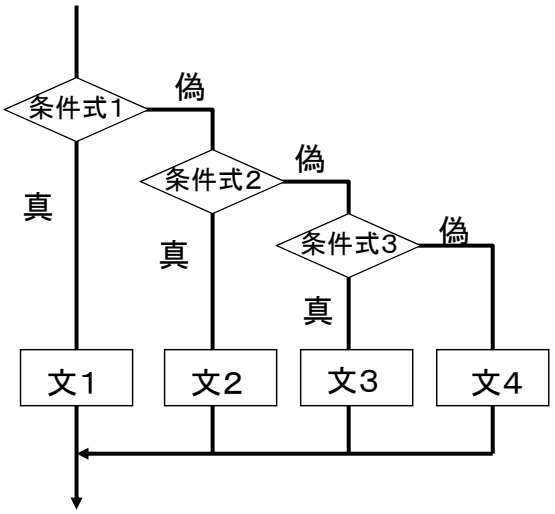
```
if(条件式1)
    {
        選択実行部分1
    }
else   if(条件式2)
    {
        選択実行部分2
    }
.
.
.
else  if(条件式n)
    {
        選択実行部分n
    }
else
   {
        選択実行部分(n+1)
   }
```

式は上から評価されて、
真になった条件式に対応する
選択実行部分が実行される。

すべての条件式が偽なら、
最後のelseの選択実行部分が
実行される。

15

多分岐のフローチャート




```
if(条件式１ )
    {
        文1
    }
else  if(条件式2)
    {
        文2
     }
else  if (条件式3)
    {
        文3
     }
else
    {
        文4
     }
```

16

多重分岐（if文の中にif文）

選択実行部分中にも、if文を書く事ができる。

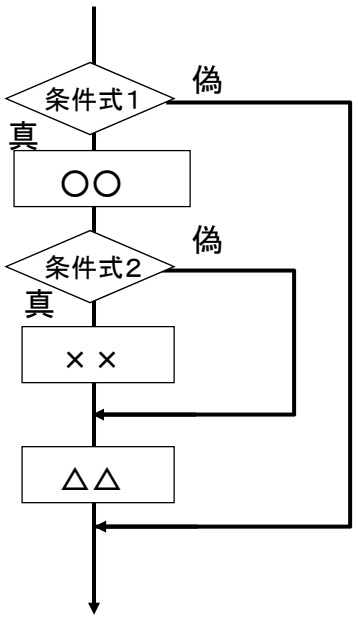
```
if(条件式)
    {
       選択実行部分1
    }
```

ここに、
またif文を書ける。

```
    if(a%2==1)
        {
            printf("aは奇数です。¥n");
            if(a>0)
                {
                    printf("aは正の奇数です。¥n");
                }
        }
```

17

多重分岐のフローチャート




```
if(条件式1)
    {

        〇〇

        if(条件式2)
            {
                ××
            }

        △△

    }
```

18

プログラム例2の原理：方程式と解の分類（1）

方程式 $ax+b=0$ の解は次のように分類される。

(1) $a \neq 0$ ∧ $b \neq 0$ の場合

解は一意であり、 $x=-\frac{b}{a}$ が解である。

(2) $a \neq 0$ ∧ $b=0$ の場合

解は一意であり、 $x=0$ が解である。

(3) $a=0$ ∧ $b=0$ の場合

解は不定であり、任意の実数 x が式を満たす。

(4) $a=0$ ∧ $b \neq 0$ の場合

解は不能であり、どんな実数 x も式を満たさない。

19

プログラム例2：
$ax+b=0$ 型の方程式を解くプログラム（多分岐版）

```
/*
        作成日：yyyy/mm/dd
        作成者：本荘太郎
        学籍番号：B00B0xx
        ソースファイル：equation1.c
        実行ファイル：equation1
        説明：
                方程式ax+c=0の解を求めるプログラム。(多分岐版)
        入力：
                標準入力から2つの係数a,bをこの順序に入力する。
        出力：
                標準出力に解の種別、解の値を出力する。
*/
/*      次のページに続く        */
```

20

```
/* 続き */
#include <stdio.h>
int  main()
{
        /*      変数宣言                */
        double a;       /*1次の係数*/
        double b;       /*定数項*/
        double x;       /*解*/

         /*係数入力*/
        printf("ax+b=0型の方程式を解きます。¥n");
        printf("1次の項の係数を入力して下さい。a=？¥n");
        scanf("%lf", &a);
        printf("定数項を入力して下さい。b=？¥n");
        scanf("%lf", &b);

/*続く*/
```

21

```
/* 続き */
        printf("方程式:(%4.1f) x+(%4.1f)=0.0)を解きます。¥n",a,b);
        if(a!=0.0  && b!=0.0)
           { /*この場合は一次方程式*/
                x=-b/a;/*a!=0.0より、割り算できる。*/
                printf("解は一意で、x=%6.2f  が解です。¥n",x);
           }
        else if(a!=0.0 && b==0.0)
           {/*この場合は一次方程式*/
                printf("解は一意で、x=0.0 が解です。¥n");
           }
        else if(a==0.0 && b==0.0)
           {/*この場合は恒真式*/
                printf("解不定、全ての実数xが式を満たします。¥n");
           }
        else    /*条件判定をしなくても、a==0.0 && b!=0.0*/
           {/*この場合は恒偽式*/
                printf("解不能、どんな実数xも式を満たしません¥n");
            }
        return  0;
}
```

22

プログラム例2の実行結果

```
$ ./equation1
ax+b=0型の方程式を解きます。
1次の項の係数を入力して下さい。a=？
2.0
定数項を入力して下さい。b=？
1.0
方程式:(( 2.0) x+( 1.0)=0.0)を解きます。
解は一意で、x=  0.50  が解です。
$
```

23

プログラム例3の原理:方程式と解の分類(2)

方程式 $ax+b=0$ の解は次のように分類される。

(1) $a \neq 0$ の場合

解は一意であり、 $x=-\frac{b}{a}$ が解である。

(2) $a=0$ の場合

(2-1) $b=0$ の場合

解は不定であり、任意の実数 x が式を満たす。

(2-2) $b \neq 0$ の場合

解は不能であり、どんな実数 x も式を満たさない。

24

プログラム例3:
$ax+b=0$ 型の方程式を解くプログラム(多重分岐版)

```
/*
        作成日:yyyy/mm/dd
        作成者:本荘太郎
        学籍番号:B00B0xx
        ソースファイル:equation2.c
        実行ファイル:equation2
        説明:
                方程式ax+c=0の解を求めるプログラム。(多重分岐版)
        入力:
                標準入力から2つの係数a,bをこの順序に入力する。
        出力:
                標準出力に解の種別、解の値を出力する。
*/
/*      次のページに続く        */
```

25

```
/* 続き */
#include <stdio.h>
int  main()
{
        /*      変数宣言                */
        double a;       /*1次の係数*/
        double b;       /*定数項*/
        double x;       /*解*/

         /*係数入力*/
        printf("ax+b=0型の方程式を解きます。¥n");
        printf("1次の項の係数を入力して下さい。a=?¥n");
        scanf("%lf", &a);
        printf("定数項を入力して下さい。b=?¥n");
        scanf("%lf", &b);

/*続く*/
```

26

```
/* 続き */
        printf("方程式:(%4.1f) x+(%4.1f)=0.0)を解きます。¥n",a,b);
        if(a!=0.0) /* この場合一次方程式 である。*/
           {
                x=-b/a;/*a!=0.0より、割り算できる。*/
                printf("解は一意で、x=%6.2f  が解です。¥n",x);
            }
        else  /* この場合、恒等式か恒偽式になる。*/
           {
                if(b==0.0)
                   {
                        printf("解不定、全ての実数xが式を満たします。¥n");
                   }
                else
                   {
                        printf("解不能、どんな実数xも式を満たしません¥n");
                   }
           }
        return  0;
}
```

27

プログラム例3の実行結果

```
$ ./equation2
ax+b=0型の方程式を解きます。
1次の項の係数を入力して下さい。a=？
0.0
定数項を入力して下さい。b=？
0.0
方程式:(( 0.0) x+( 0.0)=0.0)を解きます。
解不定、全ての実数xが式を満たします。
$
```

28