

・技術計算用言語

FORTRAN (Formula Translator)

1957年にアメリカのIBM社によって開発された科学技術計算用コンパイラ型(翻訳型)言語。非常に携帯性(どこの国のどのマシンでも動く)がよい。

C (C language)

ベル研究所のD.M リッチーとK.トンプソンによってUNIXを記述するためのシステム記述用言語として開発された(1970年)。現在はISO(国際標準化機構)によって規格化されている。

BASIC (N88BASIC)

1965年にアメリカのダートマス大学がゼネラルエレクトリック社と共同開発されたインタプリタ(会話型)言語。グラフィック処理が比較的簡単。

Visual BASIC

比較的簡単にWindowsアプリケーションが自分で作成できる。

C++ (C plus plus)

1985年AT&Tベル研究室テクニカルジャーナルBjarne Stroustrupによって開発された。最近のトレンドオブジェクト指向型モデリングが可能。

プログラミング言語はアルゴリズムを実現させるための手段。
アルゴリズムを完成させる方が最も大切であることを忘れない。

この講義では、マイクロソフト社の開発したWindows2000(基本ソフト:OS)上で、学習院大学の計算機センターで開発統合化された"Cleos"(アプリケーションソフトの名前)を用いて、C言語を用いたプログラミングについて学習する。

cプログラムの基礎の基礎

例題1(ウィンドウ上に画面に"Hello! HAL9000"と表示させる。)

```
/* ex-01 */
#include <stdio.h>
#include <math.h>

int main(void)
{
    printf("Hello! HAL9000¥n");
    return(0);
}
```

プログラム言語の基本は半角英数字。
漢字は慣れてから。
切り替えはキーボード左上の半角・全角キー

```
/* ex-01 */
#include <stdio.h>
#include <math.h>

int main(void)
{
    printf("Hello! HAL9000¥n");
    return(0);
}
```

/* ... */ はコメント文
プログラムの実行には無関係。
プログラムの見やすさを向上させる

standard Input Outputのおまじない

出力関数
文字を表示する時には" "
quotation markで囲う。¥nは改行の意味

おまじない

- ・ x 間違った文 `printf(...)` ...を表示しなさい
正しい文 `printf(...);` ...を表示しなさい。 セミコロンを付けて初めて文。
- ・ C言語は関数の集まり `printf(...)`も`return(...)`も関数
- ・ ;で区切られているのは"文"、{...}で囲まれた"文"の集まりを"複文"。
- ・ 出力文 <表現> `printf("%d", 5+3);` `printf("ABC¥n");`
<意味> 表示せよ 十進数で 5+3の値を表示せよ ABCと改行を
関数名 (引数, 引数);

例題2 (a と b の二つの定数の和、差、積、商、べき乗を出力)

```

/*  ex-02  */
#include <stdio.h>
#include <math.h>

int main(void)
{
    double a;
    double b;
    double m;
    double n;

    /* ---input--- */
    a=30.0;
    b=7.0;

    /* ---equation--- */
    m=a+b;
    n=a-b;

    /* ---output--- */
    printf("%f ¥n", m);
    printf("a-b= %f ¥n", n);

    return(0);
}

```

変数"a"を小数型として宣言。
doubleは小数
intは整数

小数として宣言した"a"という
変数に、値30.0を代入する。
小数点を忘れずに！

x=5;
は数学の「xは5に等しい」の
意味ではなく、変数xに値5を
代入している。

小数として宣言した"m"という変数に、"a+b"の
計算値を代入する。

mの値を小数(%f)として出力して改行(¥n)

nの値を小数(%f)とし
て出力する直前に
"a-b="の文字を表示

x+yの値をどんな形
(変換仕様)で出力
するかを指定する。

変換仕様 改行

```
printf("x + y= %f ¥n", x+y);
```

"..."で囲まれた部分が出力
x + y= 37.000

- ・ 変数と型 計算には値をしまっておく「箱」が必要で、これを変数といい、その名前を変数名という。
変数にはそれぞれ決められた型をもつ値を入力できる。

例題 2 a (a と b の二つの定数の和、差、積、商、べき乗を出力)

```
/* ex-02a */
#include <stdio.h>
#include <math.h>

int main(void)
{
    double a;
    double b;
    double m;
    double n;

    double kakezan, beki;

    /* ---input--- */
    a=30.0;
    b=7.0;

    /* ---equation--- */
    m=a+b;
    n=a-b;
    kakezan=a*b;

    beki=pow(a,b);

    /* ---output--- */
    printf("a+b= %f ¥n", m);
    printf("a-b= %f ¥n", n);
    printf("a*b= %f ¥n", kakezan);

    printf("a/b= %f ¥n", a/b);

    printf("a no b jyou ha %f ¥n", beki);
    return(0);
}
```

変数宣言は一文字以上でも可、まとめても

aのb乗は pow(…)関数を使う

直接計算式を書くことも可

- ・べき乗 <数学記号> A^B
<C言語> pow(A,B);

課題

$$A \cdot x^2 + B \cdot x + C = 0$$

の解を求め、出力するプログラムを書く。A, B, Cには適当な値を入れる。

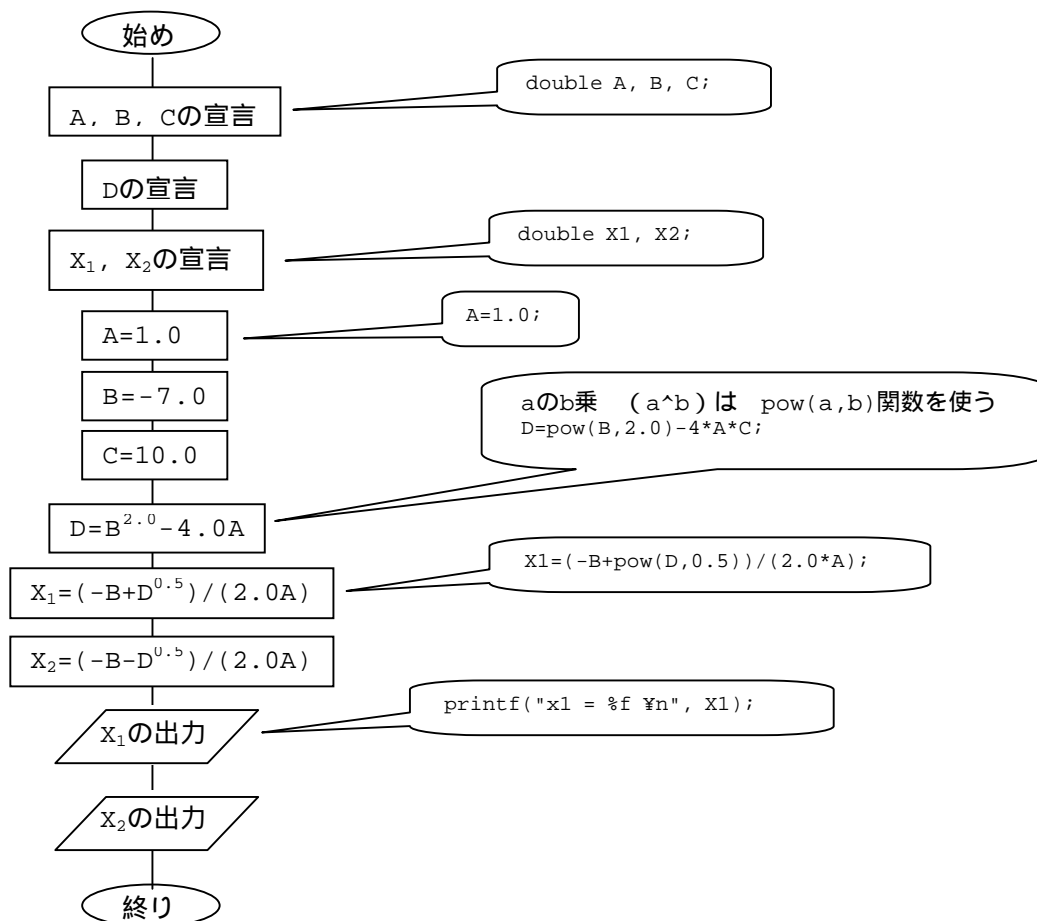
例えば A=1.0、B=-7.0、C=10.0

もちろん 二次方程式の解を求めるアルゴリズムは自分で考えてネ！
計算機は君たちの入力したアルゴリズムに従うだけ。
計算機が答えを出力できるか否かは、プログラマー次第！

二次方程式を解くプログラムのフローチャート

$$A \cdot x^2 + B \cdot x + C = 0$$

(A=1.0, B=-7.0, C=10.0)



A, B, Cにいろいろな値を入れて答えを確かめてみましょう。

出力は個人で好みが違うはず。自分の個性が活きる出力形式で答えを画面に出力してみましょう。

アルゴリズムさえフローチャート（例えば）で書ければ、後は、
1 処理を 1 文に翻訳するだけ。大切なのは「詳細」なアルゴリズム！