

東北大学全学教育科目

情報基礎 A

Linux の利用

C言語によるプログラミングその2

担当: 大学院情報科学研究科

塩浦 昭義

復習：C言語プログラムの実行のやりかた

C言語のプログラムは、そのままではコンピュータは理解できない
→機械語(コンピュータの理解できる言語)に翻訳する必要あり
(「**コンパイル**」という作業)

手順

1. 端末にて “gcc xxx.c” と入力, 実行(エンターキーを押す)
(xxx.c は実行したいプログラム名)
 2. エラーメッセージが出なかったら
→端末にて “a.out” を実行するとプログラムが動く
エラーメッセージが出たら
→プログラム内の間違いを探す
-

プログラムその1: printf

“Hello.” と表示するプログラム

```
#include <stdio.h>

main()
{
    printf("Hello.¥n");
}
```

#include ... や main() は
C言語のプログラムで
毎回使う「おまじない」

プログラム作成時には、
中括弧の中身のみ書き換える

命令の最後には必ず
セミコロン ";" をつける

printf --- 文字や数字を画面に表示するための命令
printf("xxxxxx") という形で使う。
改行を入れたいときは「¥n」を入れる。

演習問題その1

```
#include <stdio.h>

main()
{
    printf("Hello.¥n");
}
```

- プログラム名は ex1.c としてください
- 問題1-1, 1-2 を続けて解いた後のプログラムを1つだけ提出してください(プログラムを2つ提出する必要はありません)

問題1-1:

命令 printf の中の ¥n を削除してプログラムを実行してみよ.

問題1-2:

“Hello.” と表示させた後, printf をもう一回使って, 次の行に “Good Bye!” と表示するプログラムに修正せよ.

プログラムその2:変数

実際に入力して実行
してみることに

23と9の足し算を行うプログラム

```
#include <stdio.h>

main()
{
    int x, y, z;
    x = 23;
    y = 9;
    z = x + y;
    printf("%d ¥n", z);
}
```

#include ... や main() は
C言語のプログラムで
毎回使う「おまじない」

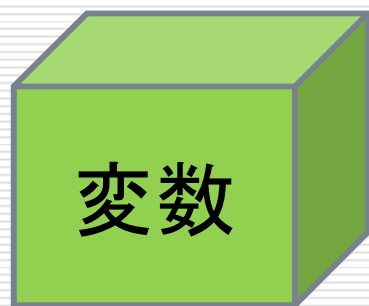
プログラム作成時には、
中括弧の中身のみ書き換える

命令の最後には必ず
セミコロン ";" をつける

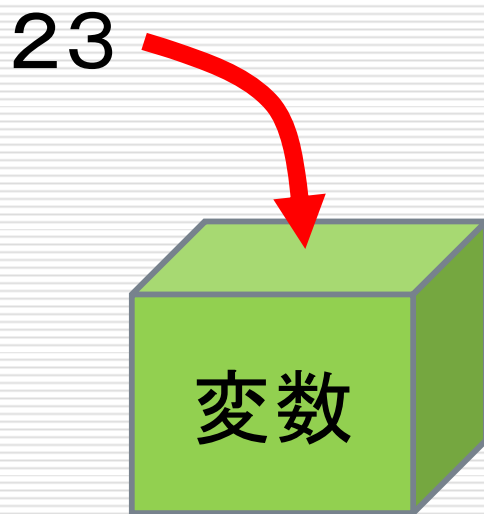
変数の利用

コンピュータは「変数」を使って数値データを覚えることができる

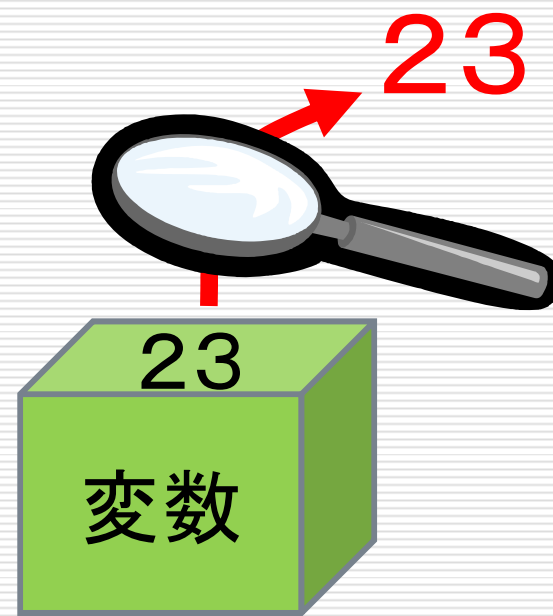
□ できること



データを入れる
「入れ物(変数)」
を準備



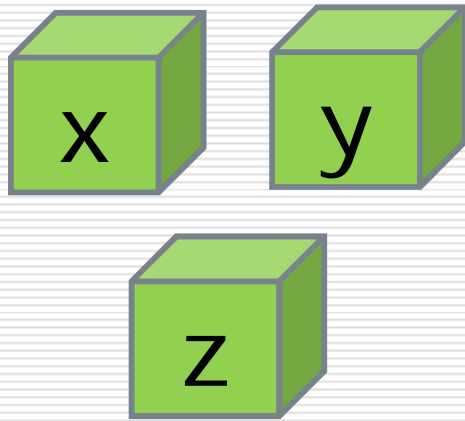
「入れ物」に数字
を入れる(覚えさ
せる)



「入れ物」に入っ
ている数字を見る

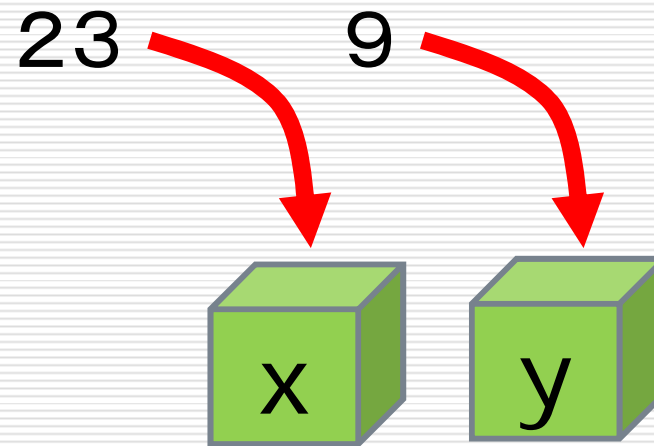
プログラム2の解説

```
int x, y, z;
```



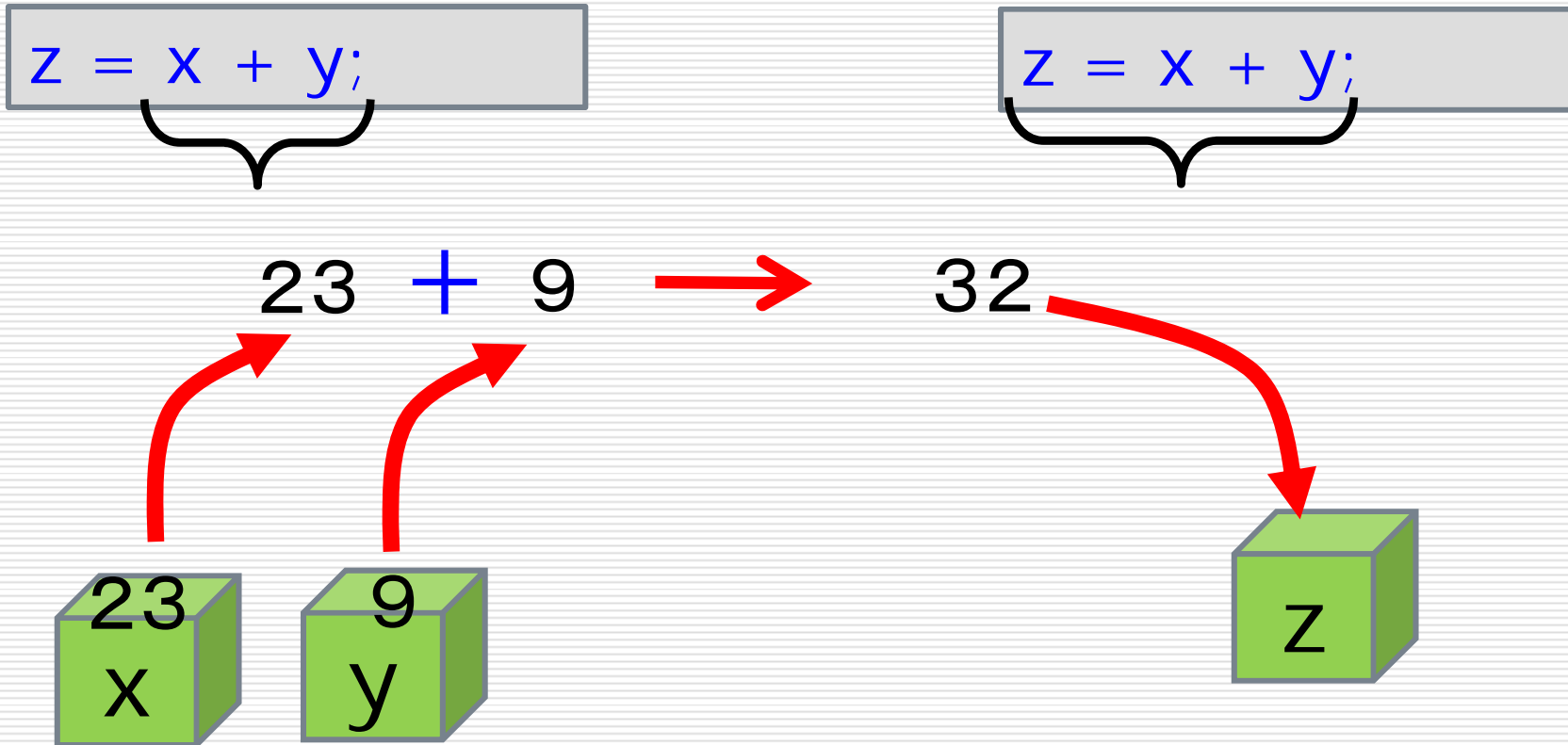
データを入れる
変数を3つ準備
名前は x, y, z とする

```
x = 23;  
y = 9;
```



変数xに数字23を入れる
変数yに数字9を入れる

プログラム2の解説



変数xとyの中身を調べて、
足し算する

足し算の結果32を
変数zに入れる

プログラム2の解説

```
printf("%d ¥n", z)
```

printf を使って変数 z の中身を表示する

```
printf("%d ¥n", z)
```

まず変数 z の中身を調べる
→32

```
printf("32 ¥n")
```

変数 z の中身を「%d」のところに入る

端末に「32」が表示される

演習問題その2

- プログラム名は ex2.c としてください
- 問題2-1, 2-2, 2-3 を続けて解いた後のプログラムを1つだけ提出してください(プログラムを3つ提出する必要はありません)

```
#include <stdio.h>

main()
{
    int x, y, z;
    x = 23;
    y = 9;
    z = x + y;
    printf("%d ¥n", z);
}
```

問題2-1:

命令 printf の行を
printf("answer = %d ¥n", z);
に置き換えてみよ。

問題2-2:

変数 x, y の値を変更してみて, 結果
がどのように変わるか確認せよ。

問題2-3:

「z = x + y;」を削除し, 命令 printf の行を
printf("answer = %d ¥n", x + y);
に置き換えてみよ。

プログラムその3: scanf

足し算に使う値を自分でキーボードから入力したい

→ 命令 scanf を使う

```
#include <stdio.h>

main()
{
    int x, y;
    x = 23;
    scanf("%d", &y);
    printf("sum = %d ¥n", x + y);
}
```

このプログラムの動き:
キーボードから数値を入力
→ 23と入力された数値
の合計(和)を出力

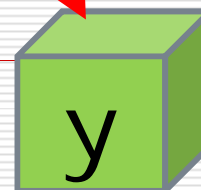
scanf の使い方

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

ここに変数の名前を入れる
他は変更の必要なし

- 「scanf("%d", &y);」が実行される
- ➔ 端末は、キーボードから数値が入力されるのを待つ
- ➔ キーボードから数値が入力されると、その値が変数 y に入る

キーボードから
入力された数値
32



変数 y に入る

演習問題その3

問題3-1:

命令 `scanf` をもう一回使って、変数 `x` の値もキーボードから入力できるように変更せよ。

問題3-2:

`x`と`y`の足し算の結果だけでなく、引き算の結果も表示するように変更せよ。

- プログラム名は `ex3.c` としてください
 - 問題3-1, 3-2 を続けて解いた後のプログラムを1つだけ提出してください(プログラムを2つ提出する必要はありません)
-

プログラムその4:if ... else ...

条件判定, 場合分けをしたいときには if ... else ... を使う

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

```
main()
```

```
{
```

```
    int x;
```

```
    printf("input x =");
```

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

```
    if (x >= 0) {
```

```
        printf("x = %d is non-negative\n", x);
```

```
    } else {
```

```
        printf("x = %d is negative\n", x);
```

```
    }
```

```
}
```

このプログラムの動き:

- 端末に「input x=」と表示される
- キーボードから数値を入力
- 数値が非負ならば「x = *** is non-negative」と表示される
- 数値が負ならば「x = *** is negative」と表示される

if ... else ... の使い方

ここに条件式を入れる

条件式の形

- 「 $x \geq y$ 」(x は y 以上)
- 「 $z < 0$ 」(z は 0 未満)
- 「 $x == y$ 」(x は y に等しい)
- 「 $x != 0$ 」(x は 0 に等しくない)

```
if (x >= 0) {  
    printf("x = %d is non-negative\n", x);  
} else {  
    printf("x = %d is negative\n", x);  
}
```

条件式が成り立たない場合に実行する命令を、2番目の中括弧の中に書く複数でも可

条件式が成り立つ場合に実行する命令を、最初の中括弧の中に書く複数でも可

演習問題その4

問題4-1:

変数 x と y の値をキーボードから入力して、
 x が y より大きかったら「 x is larger than y 」と表示し、
 x が y 以下だったら「 x is not larger than y 」と表示する
プログラムを作成せよ。

- プログラム名は `ex4.c` としてください
-

今日の課題レポート

- 演習問題1～4を解いて、作ったプログラム ex1.c, ex2.c, ex3.c, ex4.c を授業援助システムを使って提出
 - 締切: 次回の授業(7月第1週)(終わっていない場合は、不完全でも構わないので必ず提出)
 - 授業時間外でわからないところが出てきたら...
 - 簡単な質問はメールにて対応します
 - 現場でないと答えにくい質問については、テクニカルアシスタントの方に聞いてみてください(情報科目相談室(M102)にいます)
 - ただし、質問する場合は自分でしっかり考えてから、してください。授業を聞かないで質問するのも×
-