第1章 「まずは慣れよう」のまとめ

1.1 「まずは表示を行う」のまとめ

プログラムとコンパイル (教 p.2)

とは、ソースファイル（人間が読む／書く形式、C 言語の場合拡張子は ___）を実行ファイル（CPU が直接理解できる形式、Windows 上では拡張子は `.exe`）などに、翻訳することである。

注釈（コメント） (教 p.4)

とは、ソースファイル中の人間向けのメッセージで、コンパイラーは無視する部分である。C 言語では「___」から「___」までが注釈である。さらに新しい C 言語の仕様 (C99) では「`//`」から行末までという形も利用できる。

`printf` 関数 (教 p.6)

は、表示を行うための関数である。関数とは定義済みのプログラム部品である。 関数呼出しは処理の依頼であり、その時に渡すデータを_____という。

文 (教 p.7)

の末尾には、通常セミコロン「___」が必要である。「`{`」 と「`}`」の間に置かれた文は上から（同一行に複数文があるときは左から）順次実行される。

書式文字列と変換指定 (教 p.8) (教 p.378)

`printf` の第 1 引数のなかで、`%d` や `%f` などの「`%`」から始まる部分は変換指定と言い、第 2 引数以降の値に順に置き換えられる。整数（10 進数）を表示するための変換指定は「___」であり、浮動小数点数の表示は「`%f`」を使う。

文字列リテラル (教 p.11)

とは、一連の文字を二重引用符「"」～「"」で囲んだものであり、文字の並びを表す。一重引用符「'」～「'」は別の用途があるため、この目的には使用できない。

拡張表記 (教 p.11)

文字列中の「`\n`」は_____、「`\a`」は警報（ベル）、「`\t`」は_____を表す。このような、「`\`」を使った書き方を拡張表記という。（「`\`」は日本語環境では「`¥`」と表示されることがある。）その他の拡張表記については、教科書 p.250 を参照すること。

1.2 「変数」のまとめ

変数と宣言 (教 p.12)

とは、数値などのデータを収納するための「箱」（メモリの中の場所）である。C 言語では、変数を使うためには事前に宣言が必要である。

```c
int vx;         /* int 型の変数 vxの宣言 */
double fx;      /* double 型の変数 fxの宣言 */
int vx, vy;     /* int 型の変数 vxと vyの宣言 */
```

Q 1.2.1 C 言語の変数にはどのような名前をつけることができるか調べよ。（→教 p.108）

代入 (教 p.13)

とは、変数の値を書き換えることである。「変数 = 式」という形式で、右辺の式の値を左辺の変数に代入する。（代入される変数は必ず左辺に書く。「57 = vx」とは書けない。）

初期化 (教 p.14)

変数の生成（宣言）のときに値を入れることを初期化という。変数は次の形で初期化することができる。（初期化されないときは"不定値"が入る。）

```
型名 変数名1 = 初期化子1, … ,変数名n = 初期化子n;
```

初期化子 (initializer) は今のところ"式" (expression) と思っておいて良い。（あとで配列を紹介するときに、初期化子として式でない形が出てくる。）

1.3 「読み込みと表示」のまとめ

scanf 関数 (教 p.16)

とは、キーボードから数値などデータを読み込むための関数である。

```c
/* キーボードから整数を変数 noに読み込む */
scanf("____", &no);
```

「&」は「アンパサンド」と読む。詳しくは、ポインターのところで説明する。

(書き方に注意)

ところで scanf 関数は、セキュリティー上問題が多いことが指摘されている。そのため、 scanf の代わりに scanf_s 関数を使え、というメッセージを出すコンパイラーもある。しかし scanf_s 関数は、C 言語の標準仕様で必

須とされていないので、サポートしていない C コンパイラーも多い。そのため、本授業では `scanf_s` 関数は使用せず、`scanf` 関数を使う。

`puts` 関数 ⟪教 p.18⟫

は、文字列を出力し、最後に**改行**を行う。`printf` と異なり、書式変換は行わない。

文法のまとめ

式 (expression) とは

これまでのところ、

分類	一般形	補足説明
変数		`x`, `i` など
整数リテラル		`1`, `0`, `100`, `0xff` など
文字列リテラル	`"～"`	`"Hello\n"` など
関数呼出し	関数名（**式**，…，**式**）	`printf("Hello\n")` など

宣言 (declaration) とは

これまでのところ、

分類	一般形	補足説明
変数宣言	型 変数名 = 初期化子，…， 変数名 = 初期化子;	型は `int`, `double` など
