第3章 「プログラムの流れの分岐」のまとめ

3.1 「if 文」のまとめ

if 文 (教 p.44)

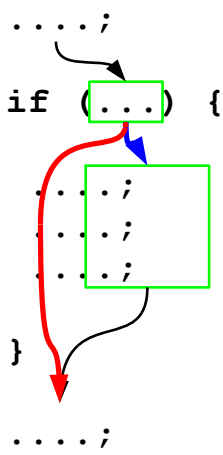
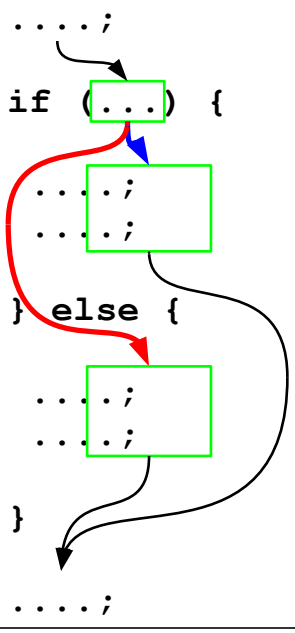
```
if ( 式1 ) 文1
```

という形のこと、式$_1$ を評価して、その値が＿＿(すなわち真) であれば、＿＿を実行する。式$_1$の値が＿＿(すなわち偽) であるときは、 何もしない。

else 付きの if 文 (教 p.46)

```
if ( 式1 ) 文1 else 文2
```

という形のこと、式$_1$ を評価して、その値が＿＿(すなわち真) であれば、＿＿を実行する。式$_1$ の値が＿＿(すなわち偽) であるときは、＿＿を実行する。

if 文	if ～ else 文
	

Q 3.1.1 次のプログラムの断片の出力は?（出力なしのときは「無」と書く。）
① `if (0) printf("X");` 答: ＿＿
② `if (0) printf("Y"); else printf("Z");` 答: ＿＿

等価演算子 (教 p.50)

「==」は両辺の値が等しければ＿＿（つまり真）を、等しくなければ＿＿（つまり偽）を返す演算子である。「==」と逆に等しくないかどうかを判定する演算子

は「　　」である。

関係演算子 (教 p.52)

以下の4つがある。

<	左辺が右辺よりも小さいとき真
>	左辺が右辺よりも大きいとき真
<=	左辺が右辺よりも小さいか等しいとき真
>=	左辺が右辺よりも大きいか等しいとき真

「≦」とか「≧」はもちろん、「=<」とか「=>」という演算子は**ない**ので注意する。

入れ子になったif文 (教 p.52)

```
1  if (no == 0)
2      puts("その数は 0です。");
3  else if (no > 0)
4      puts("その数は正です。");
5  else
6      puts("その数は負です。");
```

これは、単に else の次の文が、また if 文になっているだけのことである。

Q 3.1.2 もし、次のようになっていれば、no が 0 のときの出力はどうなるか? 改行は ↵ で表せ。

```
1  if (no == 0)
2      puts("その数は 0です。");
3  if (no > 0)
4      puts("その数は正です。");
5  else
6      puts("その数は負です。");
```

答: _____

評価 (教 p.55)

式の値を調べる（ために実行する）ことを　　する (evaluate) という。

条件演算子 (三項演算子) (教 p.58)

```
式1 ? 式2 : 式3
```

まず 式$_1$ を評価し、その値が、

非 0 (真) であれば、　　を評価して、その値を返す。　　は評価しない。
0 (偽)　であれば、　　を評価して、その値を返す。　　は評価しない。

複合文 (ブロック) (教 p.60)

文の並びを波括弧 (ブレース ― 「 」 と 「 」 ―) で囲んだものを複合文またはブロックという。 (文のほかにいくつかの宣言があってもよい。) 複合文は構文上単一の文と見なされる。 複合文中の文は上 (左) から順に一つずつ実行される。

通常、if 文の制御する文 (後述の do 文、 while 文、for 文などでも同様) は、たとえ一つの文でも (間違いを避けるため)波括弧で囲んでブロックにする。

△ 望ましくないスタイル	◎ 望ましいスタイル

`if (n1 > n2)`
`  printf("hello");`
`else`
`  printf("hi");`
	`if (n1 > n2) {`
	`  printf("hello");`
	`} else {`
	`  printf("hi");`
	`}`

教科書の例題は望ましいスタイルでないものが多いので、特に注意する。 この授業の課題の解答は「望ましいスタイル」で提出すること。 （教科書 p.61 下のほうの ▷） (教 p.61)

繰り上がりの計算 (addtime.c)

```c
#include <stdio.h>

int main(void) {
    int hour1, minute1, hour2, minute2,
        hour3, minute3;

    printf("hour1 を入力して下さい:");
    scanf("%d", &hour1);
    printf("minute1 を入力して下さい:");
    scanf("%d", &minute1);
    printf("hour2 を入力して下さい:");
    scanf("%d", &hour2);
    printf("minute2 を入力して下さい:");
    scanf("%d", &minute2);

    hour3   = hour1 + hour2;
    minute3 = minute1 + minute2;

    if (minute3 >= 60) {
        hour3   = hour3 + 1 ;
        minute3 = minute3 - 60;
    }
    printf("その和は、%d 時間 %d 分です。\n",
           hour3, minute3);
    return 0;
}
```

Q 3.1.3 次のように入力したとき、出力はどうなるか

```
hour1 を入力して下さい: 1
minute1 を入力して下さい: 40
hour2 を入力して下さい: 2
minute2 を入力して下さい: 30
```

2 つの数を大きい順に並べる (maxswap.c)

```c
#include <stdio.h>

int main(void) {
    int n1, n2, tmp;

    printf("整数 1 を入力して下さい:"); scanf("%d", &n1);
    printf("整数 2 を入力して下さい:"); scanf("%d", &n2);

    if (n2 > n1) {    /* n1 と n2 を入れ換える */
        tmp = n1;
        n1 = n2;
        n2 = tmp;
    }
    printf("大きい方は %d です。小さい方は %d です。\n",
           n1, n2);
    return 0;
}
```

Q 3.1.4 10 ～ 12 行めが、もし

```
    n2 = n1;
    n1 = n2;
```

だったら、整数 1 が 3、整数 2 が 4 のときどう出力されるか？

（発展）ぶら下がりの else (dangling else)

```c
if (h < 12) if (h < 6) printf("A"); else printf("B");
```

は、どのように文法的に解釈されるか?

解釈 1:

```c
if (h < 12) {
    if (h < 6) {
        printf("A");
    } else {
        printf("B");
    }
}
```

解釈 2:

```c
if (h < 12) {
    if (h < 6) {
        printf("A");
    }
} else {
    printf("B");
}
```

Q 3.1.5 上の解釈 1, 解釈 2 は `h` が以下の値のとき、どのように出力するか?
（出力なしのときは「無」と書く。）

	解釈 1	解釈 2
h = 3		
h = 9		
h = 15		

さらに次の文の空欄を埋めよ。

ぶら下がりの else は、解釈 ＿ と同等である。つまり、 ＿＿＿＿ の if と対応する。

論理演算子 (教 p.62)

は以下のような演算子である。左右非対称である — つまり左オペランドを評価して値が決まれば、 ＿＿＿＿＿＿ は評価しない — ことに注意する（短絡評価）。

演算子	呼び方	説明
&&	論理 AND 演算子 かつ	左オペランドを評価して、0 (偽) であれば、 1 を返す。非 0 (真) であれば、 右オペランドを評価して、0 であれば 0 を、非 0 であれば 1 を返す。
||	論理 OR 演算子 または	左オペランドを評価して、非 0 (真) であれば 1 を返す。0 (偽) であれば、 右オペランドを評価して、0 であれば 0 を、非 0 であれば 1 を返す。

Q 3.1.6 次のプログラム（の一部）は誤っている（作成者の意図に反している）可能性が高い。このままだと、どう出力するか答えよ。 (教 p.65)

1.
```
int a = 1;
if (a == 0);
    printf("hello");
```

＿＿＿＿＿＿＿＿＿＿＿＿

2.
```
int a = 0;
if (a = 0)
    printf("hello");
```

＿＿＿＿＿＿＿＿＿＿＿＿

3.
```
int a = 2, b = 2, c = 2;
if (a == b == c)
    printf("hello");
```

＿＿＿＿＿＿＿＿＿＿＿＿

4.
```
int a = 10;
if (-1 <= a <= 1)
    printf("hello");
```

＿＿＿＿＿＿＿＿＿＿＿＿

5.
```
int a = 2;
if (a & -1 <= a)
```

```
    printf("hello");
```

3.2 「switch 文」のまとめ

switch 文と break 文 (教 p.66)

ある式の値 (整数型) によって、プログラムの流れを複数に分岐するときに使う。

```
switch ( 式1 ) 文1
```

文$_1$は、通常、複合文（ブロック）である。 switch 文は式$_1$ を評価して、文$_1$ の中の _____ と「:」の間に書かれた定数と一致するところにジャンプする。 (どの case にも一致しないときは、 _____ にジャンプする。) ただし、break 文に出会うと、一気に switch 文を飛び出る。逆に break 文がなければ、そのまま次の文を実行する。

`case〜:` や `default:` のようにプログラムの飛び先を示す目印を _____ (名札) と呼ぶ。

> default の綴りを間違ってもコンパイルエラーにならないので注意すること。

Q 3.2.1 次のプログラムの断片の出力は?

```
1  int no = 2;
2
3  switch (no) {
4      case 1:  printf("A");
5      case 2:  printf("B");
6      case 3:  printf("C");
7      default: printf("D");
8  }
```

文法のまとめ

文 (statement)

に以下を追加、

分類	一般形	補足説明
if 文	`if` （式）文	(教 p.44)
else 付き if 文	`if` （式）文 `else` 文	(教 p.46)
複合文(ブロック)	{ 宣言 … 文 … }	(教 p.60)
switch 文	`switch` （式）文	(教 p.66)
ラベル付き文	`case` *整数リテラル* : 文 `default` : 文	(教 p.67)
break 文	`break` ;	(教 p.67)

式 (expression)

に以下を追加、

分類	一般形	補足説明
三項演算子	式 ? 式 : 式	(教 p.58)
