

錬成問題

▪ 'A'のように単一引用符'で文字を囲んだ式を、と呼ぶ。その型は型である。なお、単一引用符を表すは、である。

▪ 演算子*、/, %, +=のように、演算子と代入という二つの働きをもった演算子の総称は、演算子である。

▪ 演算子++と--は、それぞれ2種類がある。++aや--aのようにオペランドの左側に置いて利用する演算子の名称が、増分(減分)演算子であり、a++やa--のようにオペランドの右側に置いて利用する演算子の名称が、増分(減分)演算子である。

▪ 繰返しのことをカタカナ3文字でと呼び、入れ子構造となっている繰返しのことを と呼ぶ。

▪ 以下に示す各プログラム部分の実行結果を示せ。

```
int a = 135;
cout << "a *= 10 : " << (a *= 10) << '\n';
cout << "a /= 10 : " << (a /= 10) << '\n';
cout << "a %= 10 : " << (a %= 10) << '\n';
cout << "a += 10 : " << (a += 10) << '\n';
cout << "a -= 10 : " << (a -= 10) << '\n';
```

```
a *= 10 :
a /= 10 :
a %= 10 :
a += 10 :
a -= 10 :
```

```
int a = 1, b = 3, c = 5;
cout << "a++ : " << (a++) << '\n';
cout << "++a : " << (++a) << '\n';
cout << "b-- : " << (b--) << '\n';
cout << "--b : " << (--b) << '\n';
cout << "c : " << (c = a++ + --b) << '\n';
```

```
a++ :
++a :
b-- :
--b :
c :
```

▪ 以下に示すのは、変数nに対して、キーボードから偶数のみを読み込むためのプログラムと、3桁の奇数のみを読み込むためのプログラムである(期待しない値が入力された場合は再入力させる)。

```
int n;
 {
    cout << "偶数 : ";
    cin >> n;
}  ();
```

```
int n;
 {
    cout << "3桁の奇数 : ";
    cin >> n;
}  (!);
```

▪ 以下に示すのは、5から0までの値を1行に1個ずつ表示するプログラムである。

```
int n = 5;
 {
    cout << n << '\n';
}  ();
```

- 以下に示すのは、1 から n までの値（たとえば n が 5 であれば、1, 2, 3, 4, 5）を、1 行に 1 個ずつ表示するプログラムである。

```
int i = 1;
(18) (i <= n) {
    cout << (19) << '\n';
    (20)++;
}
```

```
(21) (int i = 1; i <= n; (22)++)
    cout << (23) << '\n';
```

- 以下に示すのは、 n 以下のすべての奇数（たとえば n が 8 であれば、1, 3, 5, 7）を小さいほうから順に 1 行に 1 個ずつ表示するプログラムである。

```
int i = 1;
(18) (i <= n) {
    cout << (24) << '\n';
    i (25) 2;
}
```

- 以下に示すのは、 n 以下の 10 の倍数（たとえば n が 45 であれば、10, 20, 30, 40）を小さいほうから順にすべて 1 行に 1 個ずつ表示するプログラムである。

```
(26) (int i = (27); i <= n / 10; i (28) )
    cout << (29) << '\n';
```

- 以下に示すのは、3 桁の整数を当てさせる数当てゲームである。正解するまでキーボードから整数を読み込み、何回で当たったのかを最後に表示する。

```
srand((30)(NULL));
int no = (31); // 当てるべき数：100~999の乱数を生成
int counter = 0; // 入力回数
int x; // キーボードから読み込んだ値

cout << "数当てゲーム開始!!\n";
cout << "100~999の数を当ててください。 \n";

(32) {
    cout << "いくつかな：";
    cin >> x;
    counter(33);

    if (x > no)
        cout << "\aもっと小さな数だよ。 \n";
    else if (x < no)
        cout << "\aもっと大きな数だよ。 \n";
} (34) (x (35) no);

cout << "正解です。" << (36) << "回で当たりましたね。 \n";
```

- 以下に示すのは、0 と 1 を交互に合計 n 個連続して表示する（たとえば n が 5 であれば『01010』と表示する）プログラムである。

```
(37) (int i = 0; i < (38); i++)
    cout << (39);
```

```
(37) (int i = 1; i <= (40); i++)
    cout << (41);
```

▪ **do** 文と **while** 文と **for** 文の総称は (42) である。**do** 文では、ループ本体は (43)。
while 文では、ループ本体は (44)。**for** 文では、ループ本体は (45)。

▶ (43) ~ (45) の選択肢：(a)一度は必ず実行される (b)一度も実行されないことがある

▪ 以下に示すのは、 n から 1 までの値 (たとえば n が 5 であれば、5, 4, 3, 2, 1) を、1 行に 1 個ずつ表示するプログラムである (n は非負であるとする：次問および次々問も)。

```
int i = 1;
while (i <= n) {
    cout << (46) << '\n';
    (47)++;
}
```

```
int i = n;
while ((48)) {
    cout << i << '\n';
    (49)--;
}
```

```
for (int i = 0; i < n; (50)++)
    cout << (51) << '\n';
```

```
for ( ; (52); (53)-- )
    cout << (54) << '\n';
```

```
do
    cout << (55) << '\n';
while (--(56) > 0);
```

▪ 以下に示すのは、2 の n 乗 (たとえば n が 3 であれば 8) を変数 x に求めて表示するプログラムである。

```
int x = (57);
int c = n;
while ((58) > 0)
    x (59) 2;
```

```
int x = (60);
for (int i = 1; i <= (61); i++)
    x (59) 2;
```

▪ 以下に示すのは、記号文字 '-' を n 個連続して表示するプログラムである。

```
int i = 0;
(62) {
    cout << '-';
    i++;
} while ((63));
```

```
int i = 1;
(62) {
    cout << '-';
    i++;
} while ((67));
```

```
int i = 0;
while ((64)) {
    cout << '-';
    i++;
}
```

```
int i = 1;
while ((68)) {
    cout << '-';
    i++;
}
```

```
while ((65) >= 1)
    cout << '-';
```

```
while ((69) >= 0)
    cout << '-';
```

```
for (int i = 0; (66); i++)
    cout << '-';
```

```
for (int i = 1; (70); i++)
    cout << '-';
```

▪ 以下に示すのは、1234567890...の数字を循環的に合計 n 個連続して表示する (たとえば n が 15 であれば「123456789012345」と表示する) プログラムである。

```
for (int i = 0; i < (71); i++)
    cout << "1234567890";
for (int i = 0; i < (72); i++)
    cout << (73);
```

```
for (int i = 1; i <= n; i++)
    cout << (74);
```

- 右に示すプログラム部分の実行結果を示せ。

```
for (int i = 1; i < 3; i++);
cout << "こんにちは。";
```

(75)

- 右に示すのは、正の整数値 n の全桁の合計を変数 sum に求めるプログラムである (たとえば n が 513 であれば、 sum の値は $5 + 1 + 3$ すなわち 9 となる)。

```
sum = 0;
while (n > (76)) {
    sum += (77);
    n /= (78);
}
```

- 以下に示すのは、1 から n までの全整数の積を $fact$ に求めるとともに、その計算式と結果を表示するプログラムである (たとえば n が 5 であれば、 $fact$ の値を 120 にするとともに、『 $1 * 2 * 3 * 4 * 5 = 120$ 』と表示する)。なお、 n には 1 以上の正の整数値が与えられているものとする。

```
int fact = (79);
for (int i = 1; i < (80); i++) {
    cout << (81);
    fact (82) i;
}
cout << (83);
fact (84) n;
cout << (85) << '\n';
```

- 以下に示すのは、九九の表を出力するプログラムである。

```
#include <(86)>
#include <iostream>
// ...
cout << " |";
for (int i = 1; i <= 9; i++)
    cout << setw((87)) << (88);
cout << "\n--+";
for (int i = 1; i <= (89); i++)
    cout << '-';
cout << '\n';
for (int i = 1; i <= (90); i++) {
    cout << (91);
    for (int j = 1; j <= (92); j++)
        cout << setw((93)) << (94);
    cout << '\n';
}
```

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

`setw` のように入出力の書式の指示を行うものは (95) と呼ばれ、この他に `dec` や `oct` などがある。

- 以下に示すのは、記号文字 '*' を並べて、 $height$ 行 $width$ 列の長方形を表示するプログラムである (実行例に示すのは $height$ が 4 で $width$ が 9 の場合の出力)。

```
for (int i = 0; i < (96); i++) {
    for (int j = 0; j < (97); j++)
        cout << '*';
    cout << (98);
}
```

```
*****
*****
*****
*****
```

▪ **do** 文が行うのは (99) 判定繰返しであり、**while** 文と **for** 文が行うのは (100) 判定繰返しである。

▪ 以下に示すのは、記号文字 '+' を、 w 個ごとに改行しながら、計 n 個表示するプログラムである (n と w は正の整数値が与えられるものとする)。最後の '+' の出力の後に改行文字を 1 個だけ出力するものとし、2 個以上出力してはならない (実行例に示すのは n が 15 で w が 6 の場合の出力)。

```
for (int i = 1; i <= n; i++) {
    cout << '+';
    if ((101))
        cout << '\n';
}
if ((102))
    cout << '\n';
```

```
+++++
+++++
+++
```

▪ 以下に示すのは、**int** 型整数 n の値を 15 桁の幅で右寄せに表示するプログラムである (実行例に示すのは n が 1234567890 の場合と、 n が -1234567890 の場合である。なお、実行例中の □ は空白文字である)。

▶ **int** 型が表現できる範囲は処理系によって異なるため、**int** 型で 1234567890 を表現できるとは限りません (次章で学習します)。

```
int d = ((103)) ? 1 : 0;
int x = ((104)) ? n : -n;
while (x > 0) {
    d ((105));
    x ((106)) 10;
}
for (int i = 0; i < ((107)); i++)
    cout << ' ';
cout << ((108));
```

```
□□□□□1234567890
```

```
□□□□-1234567890
```

▪ 以下に示すのは、記号文字 '+' を、 w 個ごとに改行しながら、計 n 個表示するプログラムである。表示においては、3 行おきに空の行を出力する。最後の '+' の出力の後に改行文字を 1 個だけ出力するものとし、2 個以上出力してはならない (n と w は正の整数値が与えられるものとする)。実行例に示すのは n が 45 で w が 6 の場合の出力)。

```
int p = ((109));
int q = ((110));
for (int i = 1; i <= ((111)); i++) {
    for (int j = 1; j <= w; j++)
        cout << '+';
    cout << '\n';
    if (i % 3 == 0 && ((112)))
        cout << '\n';
}
for (int i = 1; i <= ((113)); i++)
    cout << '+';
if (q != 0) cout << '\n';
```

```
+++++
+++++
+++++
+++++
+++++
+++++
+++++
+++++
+++++
```

■ 以下に示すのは、 n 段の逆向きピラミッドを表示するプログラムである。 i 段目に表示するのは、 i の最下位桁の数字である（実行例に示すのは n が13の場合の出力）。

```
for (int i = 0; i < (114); i++) {
    for (int j = 1; j < (115); j++)
        cout << ' ';
    for (int j = 1; j < (116); j++)
        cout << (117);
    cout << '\n';
}
```

```
11111111111111111111111111111111
222222222222222222222222222222
333333333333333333333333333333
444444444444444444444444444444
555555555555555555555555555555
666666666666666666666666666666
777777777777777777777777777777
888888888888888888888888888888
999999999999999999999999999999
00000000
11111
222
3
```

■ 以下に示すのは、 $height$ 行 $width$ 列の長方形を数字文字で表示するプログラムである。表示する数字は、1行目は1234567890...で、2行目は2345678901...である（実行例に示すのは $height$ が13で $width$ が12の場合の出力）。

```
for (int i = 1; i <= (118); i++) {
    for (int j = 1; j <= (119); j++)
        cout << (120);
    cout << '\n';
}
```

```
123456789012
234567890123
345678901234
456789012345
567890123456
678901234567
789012345678
890123456789
901234567890
012345678901
123456789012
234567890123
345678901234
```

■ 以下に示すのは、面積が a である長方形の辺の長さを列挙するプログラムである。なお、縦の長さ h と横の長さ w は整数で、 $h \leq w$ とする（実行例に示すのは a が12の場合の出力）。

```
for (int i = 1; i < a; i++) {
    if (i * i > a) (121);
    if (a % i != 0) (122);
    cout << "縦" << i << "×横" << (123) << '\n';
}
```

```
縦1×横12
縦2×横6
縦3×横4
```

■ 以下に示すのは、 $height$ 行5列の平行四辺形（上辺と下辺は水平で、段が下るたびに1文字ずつ左にずれていくもの）を表示するプログラムである（実行例に示すのは $height$ が3の場合の出力）。

```
for (int i = 0; i < (124); i += (125)) {
    cout << (126) << "*****"
        << (127);
}
```

```
*****
*****
*****
```

■ $x \ \&\& \ y$ と $!(x \ || \ !y)$ とが等しく、 $x \ || \ y$ と $!(x \ \&\& \ !y)$ とが等しいことを、(128)の法則と呼ぶ。

■ 代入の左辺にも右辺にも置ける式を(129)式と呼び、右辺には置いて左辺には置けない式を(130)式と呼ぶ。

▪ 演算子 ++ は、オペランドの値を 1 だけ増やす演算子である。その名称は (131) 演算子であり、オペランドの値を 1 だけ増やすことは『(132) する』と呼ばれる。

演算子 -- は、オペランドの値を 1 だけ減らす演算子である。その名称は (133) 演算子であり、オペランドの値を 1 だけ減らすことは『(134) する』と呼ばれる。

▪ 以下に示す各文字を表す拡張表記を示せ。

警報 … (135) 後退 … (136) 逆斜線 … (137) 単一引用符 … (138)
改行 … (139) 復帰 … (140) 水平タブ … (141) 二重引用符 … (142)

▪ 以下に示す各プログラムの実行結果を示せ。

```
int y = 3;
int z = 3;
while (y >= 0)
    cout << y-- << " " << ++z << '\n';
```

(143)

```
int n = 1;
for (int i = 1; i < 10; i++) {
    if (i == 3) continue;
    cout << i;
    if (i == 5) goto abc;
}
cout << 'A';
abc:
cout << 'B';
cout << 'C';
```

(144)

goto 文（上のプログラムの場合での網かけ部）が実行されると、プログラムの流れは (145) 文へと飛ぶ。

上のプログラムでの (145) 文が、どこからどこまでであるかを示せ… (146)。

▪ 右に示すのは、「"ABC"」と表示するプログラムである。

```
cout << (147);
```

▪ 以下に示す各プログラムの実行結果を示せ。

```
cout << "\aa\bbc\n";
cout << "ABC\rDEF\n";
```

(148)

```
for (int i = 1; i < 5; i++)
    cout << setw(i) << i << '\n';
```

(149)

```
for (int i = 1, j = 10; i < 5; i++, j--)
    cout << i, j;
cout << '\n';
```

(150)

```
for (int i = 0; cout << i, i < 3; i++)
    ;
cout << '\n';
```

(151)