

錬成問題

- 変数・リテラル・変数やリテラルを演算子で結合したものを (1) と呼ぶ。
 (1) にセミコロンが付いて文になったものは (2) であり、セミコロンだけの文は (3) である。また、任意の個数の文を { } で囲んだ文は (4) と呼ばれる。

- $a + b * c$ では、左側の加算よりも右側の乗算のほうが先に行われる。これは、演算子 $*$ の (5) が演算子 $+$ のそれよりも高いためである。なお、同じ (5) の演算子が連続する場合に、左右どちらの演算が先に行われるのかは、演算子の (6) に基づいて決定される。

- オペランドの大小関係を判定する演算子 $<$, $>$, $<=$, $>=$ の総称は (7) 演算子であり、等しいか等しくないかを判定する演算子 $==$, $!=$ の総称は (8) 演算子である。
 また、演算子 $\&\&$, $\|\|$ の総称は (9) 演算子であり、演算子 $!$ の名称は (10) 演算子であり、演算子 $? :$ の名称は (11) 演算子である。

- 論理演算の式全体の評価結果が、左オペランドの評価のみで明確になる場合に、右オペランドの評価が行われないことを (12) という。

- **if** や **else** などの語句は、特別な意味が与えられており、(13) と呼ばれる。変数や関数などに与えられる名前は (14) と呼ばれる。

- 変数を値の小さい順や大きい順で並べかえることを (15) と呼ぶ。

- アルゴリズムとは、問題を解くためのものであって、(16) に定義され、順序付けられた (17) の規則からなる集合のことである。

- 以下に示す真理値表の空欄を埋めよ。

a 論理積

x	y	$x \ \&\& \ y$
true	true	(18)
true	false	(19)
false	true	(20)
false	false	(21)

b 論理和

x	y	$x \ \ \ \ y$
true	true	(22)
true	false	(23)
false	true	(24)
false	false	(25)

- 以下に示すのは、変数 a の絶対値を表示するプログラムである。

```
cout << "aの絶対値は" << (26) << "です。\\n";
```

- 以下に示すのは、変数 a と変数 b の小さいほうの値と大きいほうの値の両方を表示するプログラムである。

```
cout << "aとbの小さいほうの値は" << (27) << "です。 \n";
cout << "aとbの大きいほうの値は" << (28) << "です。 \n";
```

- 以下に示すのは、変数 c の最下位桁が 0 であれば（たとえば 50 や 250 であれば）、『変数 c の最下位桁は 0 です。』と表示するプログラムである。

```
(29) (c (30) 10 == 0)
cout << "変数cの最下位桁は0です。 \n";
```

- 以下に示すのは、変数 x の値が 0 であれば『ゼロです。』と表示し、そうでなければ『非ゼロです。』と表示するプログラムである。

```
(29) (x == 0)
cout << "ゼロです。 \n";
(31)
cout << "非ゼロです。 \n";
```

```
cout << (x == 0 (32)) << "です。 \n";
```

- 以下に示すプログラム部分の実行結果を示せ。

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

```
a - b - c :
c - b - a :
a = b = c :
c = b = a :
```

(33)

```
int m = 0, n = 1;
(29) (!m) cout << "A"; (31) cout << "B";
(29) (!n) cout << "C"; (31) cout << "D";
```

(34)

- 右に示すのは、変数 n の値が 0 であれば『ゼロです。』と表示し、そうでなく 1 桁の値であれば『 1 桁です。』と表示し、そうでなく 2 桁であれば『 2 桁です。』と表示するプログラムである。

```
(29) (n (35) 0)
cout << "ゼロです。 \n";
(36) (n >= -9 (37) n <= 9)
cout << "1桁です。 \n";
(38)
cout << "2桁です。 \n";
```

- 以下に示すのは、変数 w の値が 0 であれば『晴れ』、 1 であれば『雨』、 2 であれば『曇り』と表示するプログラムである。

```
(39) (w) {
(40) 0: cout << "晴れ \n"; (41) ;
(40) 1: cout << "雨 \n"; (41) ;
(40) 2: cout << "曇り \n"; (41) ;
}
```

- 真と偽を表す論理値リテラルは、それぞれ (42) と (43) である。
- 加算を行う + 演算子は (44) 結合であり、代入を行う = 演算子は (45) 結合である。
- 以下に示すのは、変数 m の値が 3, 4, 5 のいずれかであれば『春です。』と表示し、そうでなければ『春ではありません。』と表示するプログラムである。

```
if ( (46) || (47) || (48) )
    cout << "春です。 \n";
else
    cout << "春ではありません。 \n";
```

```
if ( (49) && (50) )
    cout << "春です。 \n";
else
    cout << "春ではありません。 \n";
```

```
if (!( (51) || (52) ))
    cout << "春です。 \n";
else
    cout << "春ではありません。 \n";
```

- 以下に示すのは、変数 m の値を 3 で割った剰余に応じて、『3 で割り切れます。』『3 で割った剰余は 1 です。』『3 で割った剰余は 2 です。』のいずれかを表示するプログラムである。

```
if ( (53) )
    cout << "3 で割り切れます。 \n";
else (54)
    cout << "3 で割った剰余は1です。 \n";
else
    cout << "3 で割った剰余は2です。 \n";
```

```
if ( (55) )
    cout << "3 で割った剰余は" << mod << "です。 \n";
else
    cout << "3 で割り切れます。 \n";
```

```
switch ( (56) ) {
    case (57) cout << "3 で割り切れます。 \n";      break;
    case (58) cout << "3 で割った剰余は1です。 \n";  break;
    case (59) cout << "3 で割った剰余は2です。 \n";  break;
}
```

- 以下に示すのは、変数 a の値が b の値の倍数かどうかを判断し、その結果を表示するプログラムである。

```
if ( (60) )
    cout << "aはbの倍数です。 \n";
else
    cout << "aはbの倍数ではありません。 \n";
```

▪ 右に示すのは、変数 a と b の値が $a \geq b$ となるようにソートするプログラムである。

```
if ( (61) ) {
    int t = a; a = (62); b = (63);
}
```

▪ 右に示す **switch** 文は、**int** 型変数 n の値が 1, 2, 3 のときに、それぞれ (64)、(65)、(66) と表示する。

```
switch (n) {
    case 1: cout << "A";
    case 2: cout << "B"; break;
    default: cout << "C";
}
```

▪ 右に示すプログラムは、変数 n の値が 0, 1 のときに、それぞれ (67)、(68) と表示する。

```
if (n == 1)
    cout << "A";
```

▪ 右に示すプログラムは、変数 n の値が 0, 1 のときに、それぞれ (69)、(70) と表示する。

```
if (n == 0);
    cout << "A";
```

▪ 右に示すのは、変数 a , b , c の値がすべて等しいときに「★」と表示するプログラムである。

```
if ( (71) && (72) )
    cout << "★";
```

▪ 右に示すのは、変数 a , b , c の二つあるいは三つの値が等しいときに「★」と表示するプログラムである。

```
if ( (73) || (74) )
    cout << "★";
```

▪ 以下に示すのは、変数 a の値が b の値の平方根より小さければ『 a は b の平方根より小さい。』と表示するプログラムである。

```
if (a * (75) < (76))
    cout << "aはbの平方根より小さい。 \n";
```

▪ 右に示すのは、変数 a と b に読み込んだ整数値が両方とも奇数か、一方のみ奇数か、両方とも偶数であるかを表示するプログラムである。

▪ **int** 型の変数 n と k の値が 15 と 17 であるとする。以下に示す各式を評価して得られる型と値を示せ。

$n + 27$ … 型: (84) 値: (85)

$n < 52$ … 型: (86) 値: (87)

$n = 34$ … 型: (88) 値: (89)

$n == 8$ … 型: (90) 値: (91)

$n ? 5 : 3$ … 型: (92) 値: (93)

$n > k ? n : k$ … 型: (94) 値: (95)

```
(77) include <iostream>
using namespace std;
int main (78)
{
    int a, b;
    cout << "a : "; cin >> a;
    cout << "b : "; cin >> b;

    int c = 0;
    if ( (79) ) c = c + 1;
    if ( (80) ) c = c + 1;

    if ( (81) == 0 )
        cout << "両方とも奇数です。 \n";
    else if ( (82) )
        cout << "一方のみ奇数です。 \n";
    else if ( (83) )
        cout << "両方とも偶数です。 \n";
}
```

▪ **switch** 文内のプログラムの流れの飛び先である **case** ?? : や **default** : は、(96) と呼ばれる。なお、**case** の後ろに置かれる ?? は (97)、(98)。**if** 文と **switch** 文の総称は (99) である。

- ▶ (97) の選択肢 : (a)整数でも実数でもよく (b)整数でなければならない
- ▶ (98) の選択肢 : (a)変数でも定数でもよい (b)定数でなければならない

▪ 識別子として正しいものに○を、誤ったものに×を記入せよ。

case ... (100) case1 ... (101) pc98 ... (102) 98nx ... (103)
abc ... (104) abc ... (105) ab#c ... (106) ab\$c ... (107)

なお、識別子の大きい文字と小文字は区別 (108)。

- ▶ (108) の選択肢 : (a)される (b)されない

▪ 以下に示すのは、変数 a , b , c , d の最大値を求めて max に代入するプログラムである。

```
int max = a;
if ( (109) ) max = b;
if ( (110) ) max = c;
if ( (111) ) max = d;
```

```
int max1 = a > b ? (112) : (113);
int max2 = c > d ? (114) : (115);
int max = max1 > max2 ? (116) : (117);
```

▪ 以下に示すのは、変数 a , b を昇順 ($a \leq b$ となるよう) にソートするプログラムである。

```
if (a > b) {
    int t = (118);
    a = (119);
    b = (120);
}
```

```
if (!( (121) )) {
    int t = b;
    (122) = (123);
    (124) = (125);
}
```

▪ 以下に示すプログラムのすべての誤りを指摘して正せ。 … (126)

```
/** 小さいほうの値と大きいほうの値を求めて表示 */
#include <iostream>
using namespace std;
int main()
{
    int a, b; /* 読み込む値 */
    cout << "整数a:"; cin << a;
    cout << "整数b:"; cin << b;
    if (a = b) {
        cout << "二つの値は同じです。 \n";
    } else {
        int min, max; /* 小さいほう / 大きいほうの値
        if (a < b) { /* aがbより小さければ…
            min = a;
            max = b;
        } else { /* そうでなければ…
            min = b;
            max = a;
        }
    }
    cout << "小さいほうの値は" << min << "です。 \n";
    cout << "大きいほうの値は" << max << "です。 \n";
}
```