

## 錬成問題

※ 空欄部を埋めてください（次章以降の錬成問題も同様です）。

- C++ は、 指向プログラミングをサポートする言語である。
- 私たち人間が文字の並びとして作成するプログラムは  と呼ばれる。  はそのままでは実行できないので、 やリンクなどの作業が必要である。

▪ 作成者を含め、その読み手に伝えたいことがらを、簡潔な言葉としてプログラムに書き込まれたものが、“注釈”すなわち  である。その有無や内容によってプログラムの動作が変わることは  。

複数行にわたることのできる  は、 と  とで囲んで表記する。一方、先頭に  を置く形式では、行末までが注釈となる。

▶  の選択肢：(a)ある (b)ない

▪ 82 や 17 といった整数値を表す定数を  と呼び、3.14 や 2.5 といった実数値を表す定数を  と呼ぶ。また、“ABC” のように二重引用符 " で囲んだ文字の並びを、 と呼ぶ。

▪ 以下に示す記号文字の読み方をカタカナで示せ。

; ...       : ...       . ...       , ...   
 { ...       ( ...       ' ...       " ...

▪ 整数や実数などの値を自由に出し入れできて、そのデータを保持するために利用するのが  であり、必要になった時点で  と名前を与えて宣言する。

の生成時に値を入れるのが  であり、いったん生成した後に値を入れるのが  である。

▪ 右に示すのは、『風林火山』の各文字を 1 行に 1 文字ずつ表示するプログラムである。

▪ 空白文字・改行文字・水平タブ文字・垂直タブ文字・書式送り文字の総称が  であり、 と注釈の総称が  である。

```
 <iostream>
using  std;
int  ()
{
     << "風\n";
     << "林\n";
     << "火\n";
     << "山\n";
}
```

▪  << "AB\nC" << "D\n\nEFG"; を実行すると  と表示される。

▪ 演算を行う + や \* などの記号を (31) と呼び、演算の対象となる式を (32) と呼ぶ。演算の対象となる (32) の個数は (31) によって異なる。少ないものは (33) 個で、多いもので (34) 個である。なお、= の名称は (35) であり、\*, /, %, +, - の総称は (36) である。

▪ ソースプログラムは、その構造に応じて左側に数行ごとの単位での余白を設けると読みやすくなる。その余白のことを (37) と呼ぶ。

▪ rand 関数が生成する乱数は、(38) である。

▶ (38) の選択肢：(a) 正の整数値 (b) 0 または正の整数値 (c) 実数値

▪ 以下に示すのは、キーボードから読み込んだ二つの int 型整数 x, y を加減乗除した値を表示するプログラムである。

```
(39) 二つの整数値を読み込んで加減乗除した値を表示
#include (40) ①
using namespace (41); ②
(42) main (43)
{
    int x (44) y;
    cout << "xとyを加減乗除します。\\n";
    cout << "xの値："; cin (45) x;
    cout << "yの値："; cin (45) y;
    cout << "xにyを加えた値は" << x (46) y << "です。\\n";
    cout << "xからyを減じた値は" << x (47) y << "です。\\n";
    cout << "xにyを乗じた値は" << x (48) y << "です。\\n";
    cout << "xをyで割った商は" << x (49) y << "です。\\n";
    cout << "xをyで割った剰余は" << x (50) y << "です。\\n";
}
```

なお、①、②、③は、それぞれ (51)、(52)、(53) である。

▪ 以下に示すのは、double 型変数 x の符号を反転した値 (x が 5.7 であれば -5.7) と、x と y の平均を表示するプログラム部分である。

▶ 前問の①、②に相当する部分や、③の枠組みなどは省略している。次章以降の問題でも、本問のように、プログラムの部分のみを示すことがある。

```
cout << "xの符号を反転した値は" << (54) << "です。\\n";
cout << "xとyの平均は" << (55) << "です。\\n";
```

▪ 以下に示すのは、文字を読み込んで、その文字を反復表示するプログラム部分である。

```
(56) ch; // 文字を読み込むための変数の宣言
cout << "文字：";
cin (45) ch;
cout << "入力された文字は" << ch << "です。\\n" (57)
```

- 以下に示す各プログラム部分の実行結果を示せ。

```
cout << 13 << 57 << "\n";
cout << 13 + 57 << "\n";
```

(58)

```
int x = 10;
cout << "x = " << "x" << "\n";
cout << "x = " << x << "\n";
cout << "x = " << +x << "\n";
cout << "x = " << -x << "\n";
cout << "x = " << x-x << "\n";
```

(59)

```
cout << 25 * 7 << "\n";
cout << 25 / 7 << "\n";
cout << 25 % 7 << "\n";
cout << 25 + 7 / 2 << "\n";
cout << (25 - 7) / 2 << "\n";
```

(60)

```
cout << "ABC" "DEF" << "\n";
cout << "ABC" /**/ "DEF" << "\n"
    "A";
```

(61)

- 以下に示すのは、キーボードから読み込んだ **double** 型値の半径をもつ円の円周の長さ  
と面積を表示するプログラム部分である。

```
(62) double PI = 3.1416; // 円周率
double r; // 半径

cout << "半径："; // 半径の入力を促す
cin (45) r; // 半径を読み込む

cout << "円周の長さは" << (63) << "です。 \n"; // 円周
cout << "面積は" << (64) << "です。 \n"; // 面積
```

なお、PI のように (62) を付けて宣言された変数は、(65) と呼ばれる。

- 以下に示すのは、姓と名を読み込んで、日本語と英語の両方で挨拶を行うプログラム  
である。たとえば、姓として「Shibata」が入力されて、名として「BohYoh」が入力さ  
れた場合は、まず『こんにちは、Shibata BohYohさん。』と表示し、それから『Hello,  
BohYoh Shibata.』と表示する。

```
#include (40)
#include (66)

using namespace (41);

(42) main (43)
{
    (67) first, last; // 姓と名
    cout << "姓は："; cin (68) last;
    cout << "名は："; cin (68) first;
    cout << "こんにちは、" << (69) << "さん。 \n";
    cout << "Hello, " << (70) << ". \n";
}
```

cout と cin は、外部の機器と接続された文字の流れる川のようなものであり (71) と呼ばれる。また、<< と >> の名称は、それぞれ (72) と (73) である。

- 以下に示すのは、キーボードから整数値を読み込み、その値を10で割った剰余を表示するプログラム部分である。

```
(74) x;
cout << "整数値：";
(75) >> x;
cout << "10で割った剰余は" << (76) << "です。\\n";
```

- 以下に示すのは、3桁の奇数(101, 103, ..., 999)を乱数として生成し、その乱数と、その乱数を10倍した値と、その乱数の最下位の桁を表示する(生成した乱数が537であれば537と5370と7を表示する)プログラムである(一部省略している)。

```
(77) 3桁の奇数の乱数を生成するプログラム (78)
#include <iostream>
#include <(79)>
#include <(80)>
// ... 中略 ...
{
    (81)(time((82)));
    int k = (83) + 2 * ((84)() % (85));
    cout << "生成した乱数は" << k << "です。\\n";
    cout << "10倍した値は" << (86) << "です。\\n";
    cout << "最下位の桁は" << (87) << "です。\\n";
}
```

- (88) と呼ばれる `iostream`, (79), (80) などには、ライブラリに関する情報が格納されている。

- 式  $x * y$  と  $x + y$  における  $*$  と  $+$  は (89) 演算子であり、式  $+1$  と  $-15$  における  $+$  と  $-$  は (90) 演算子である。

- 以下に示すプログラムのすべての誤りを指摘して正せ。… (91)

```
include <iostream>
using namespace std;
imt main()
[
    const int a = 5;
    const int b = 7;
    const double x = 3.7;
    const double y = 8.4;
    const int sum;

    sum = (a + b) / 2 // aとbの平均

    cont >> "a = " + a;
    cont >> "b = " + b;
    cont >> "x = " + x;
    cont >> "y = " + y;
    cont >> "aをbで割った剰余は" + a % b + "です。\\n";
    cont >> "aとbの平均は" + sum + "です。\\n";
    cont >> "xをyで割った剰余は" + x % y + "です。\\n";
]
```