

### 問題 2.1.1

この問題は、数の分類（→**2.1.1項**）を理解しているかを問う問題です。解答は、

- 整数： $-100, -20, 0, 1, 70$
- 正の整数： $1, 70$

となります。整数は小数点が付かないような数、正の整数はそれらのうち  $0$  より大きい数のことを指します。

### 問題 2.1.2

この問題は、文字式とその書き方（→**2.1.2項**）を問う問題です。解答は、

- $A + B + C = 25 + 4 + 12 = 41$
- $ABC = 25 \times 4 \times 12 = 1200$

となります。ここで、 $ABC$  は  $A \times B \times C$  という意味であることに注意してください。

### 問題 2.1.3

この問題は、「3つの整数を入力し、この掛け算の値を出力してください」という意味です。したがって、次のようなプログラムを書くと正解が得られます。なお、数列のように番号を付した文字式に慣れていない人は、**2.1.4項・2.1.5項**に戻って復習しましょう。

```
#include <iostream>
using namespace std;

int main() {
    int A[4];
    cin >> A[1] >> A[2] >> A[3];
    cout << A[1] * A[2] * A[3] << endl;
    return 0;
}
```

※ Python などのソースコードは GitHub の codes フォルダをご覧ください。

### 問題 2.1.4 (1)

この問題は、2 進法を 10 進法に変換する方法（→**2.1.7項**）を問う問題です。以下の  
ようにして計算すると、答えが **9** であることが分かります。

- 8 の位の値は 1
- 4 の位の値は 0
- 2 の位の値は 0
- 1 の位の値は 1
- **位とその桁の数の足し算**は、 $(8 \times 1) + (4 \times 0) + (2 \times 0) + (1 \times 1) = 9$

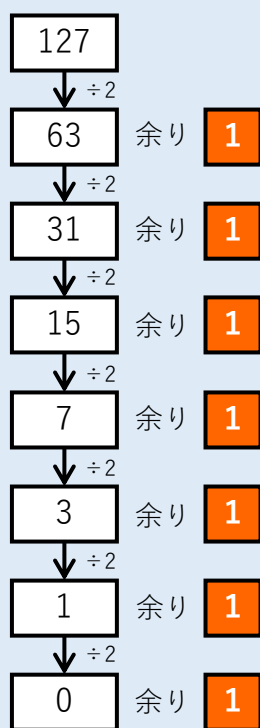
### 問題 2.1.4 (2)

この問題は、10 進法を 2 進法などに変換する方法（→**2.1.9項**）を問う問題です。  
一般に、10 進法を  $K$  進法に変換するときは、値がゼロになるまで  $K$  で割り続け、余  
りを後ろから読めば良いです。したがって、以下のようにして計算すると、

- 127 の 2 進法表記が **1111111**
- 127 の 3 進法表記が **11201**

であることが分かります。イメージ図は以下の通りです。

#### 127 を 2 進法に変換



#### 127 を 3 進法に変換

