補足資料：関数の作成

自分でオリジナルの関数を作成する場合は、次の書式を使用します。

書式：関数の作成

```
戻り値の型　関数名( 引数の型　引数) {
    文;
}
```

戻り値（return value）とは「関数の実行後に戻される値」です。例えば、引数で渡された値の 2 倍を計算して、値を戻す関数の場合は次のように書きます。

List：2 倍の値を計算する関数

```
int twice(int num){
    return num * 2;
}
```

この例では、2 倍した値を戻すので〈戻り値の型〉は「int」（整数型）になります。「twice」は関数名です。関数名は他の重複しない名前であれば自由に決めることができます（慣例として小文字からはじまる名前にします）。

twice の後に続く括弧の中の「int」が〈引数の型〉です。引数である変数 num の型をここに書きます。変数 num はこの関数の引数です。

関数の中に記述されている『**return**』は return 文と呼ばれる文です。return 文には次の 2 つの役割があります。

・return 文の後ろに書かれた値を関数の戻り値に設定する
・この関数を呼び出している元のプログラムに処理を戻す

上記の例の場合、return 文は「num * 2」の計算結果をこの関数の戻り値に設定し、処理を元のプログラムに戻します。例えば、引数に「2」が設定された場合は「4」（2 * 2）を戻り値に設定して、処理を戻します。なお、return 文が実行されると処理が戻されてしまうので、return 文の後にコードを書いても、それらは処理されません。

このように関数を作成することを「**関数の定義（definition）**」といいます。

□ 作成した関数の使用方法

作成した関数を使用する場合は、次のようにコードを書きます。

List：twice()関数の作成と実行

```
#import <Foundation/Foundation.h>

int twice(int num){
    return num * 2;
}

int main(int argc, const char * argv[])
{
    @autoreleasepool{

        int result = twice(2);
        NSLog(@"%d", result);
    }
    return 0;
}
```

結果：twice()関数の実行結果

```
4
```

上記のコードでは「int result = twice(2);」の行で、twice()関数を実行しています。このコードは「twice(2)の実行結果を int 型の変数 result に代入せよ」という命令です。引数に「2」が設定されているので、twice()関数の戻り値は「4」（2 * 2 の計算結果）となり、変数 result には「4」が代入されます。

twice()関数は「int main(int argc, const char * argv[])」よりもの上に書きます。

本書内で「コンピュータは命令を上から下に処理する」という説明をしましたが、みなさんが自由に名前をつけた twice()関数がコードの途中でいきなり出てくるとコンピュータはどう処理していいかわかりません。そこで、前もって twice()関数が何であるのかをコンピュータに教えるために main より上に書きます。twice()関数を呼び出す命令の上であればよいということなら main の中でも良さそうですが、これはエラーになってしまいます。