

## 【課題 1】（提出期限 2021 年 12 月 10 日（金）15：00）

それぞれが以下の機能をもつ getMax 関数を関数宣言を用いて 3 種類定義せよ：

- int 型 vector の最大要素を返却する
- double 型 vector の最大要素を返却する
- string 型 vector の最長文字列の要素を返却する

ただし、main 文は変更しないこととし、プログラム全体を ex10-1.cpp として提出すること。

```
int main() {  
    std::vector<int> veci = { -20, 20, 0, -10, 10 };  
    std::vector<double> vecd = { -5.5, 5.5, 0, -3.3, 3.0 };  
    std::vector<string> vecstr = { "ZERO", "ONE", "TWO", "THREE", "FOUR" };  
  
    cout <<"getMax on int: " << getMax(veci) << "¥n"  
        << "getMax on double: " << getMax(vecd) << "¥n"  
        << "getMax on string: " << getMax(vecstr) << "¥n";  
  
    return 0;  
}
```

## 実行例 1-1

```
% ./ex10-1
```

```
getMax on int: 20
```

```
getMax on double: 5.5
```

```
getMax on string: THREE
```

※補足：関数を多重定義（オーバーロード）するときには、引数の型または個数が異なるようにしておかなければならない。なぜならば、もしそうでなく（引数の型と個数が同じで）戻り値の型だけが違っているならば、関数呼び出しの際にどの関数を呼び出すのかを判断できないことになる。

【課題 2】（提出期限 12 月 12 日（日）23 : 59）

2 つの整数および 2 つの小数を入力し、それぞれにおいて最大値を返すような関数テンプレートを作成せよ。  
ただし、(2) (3) の箇所を除いて main 文は変更しないこととし、プログラム全体を ex10-2. cpp として提出すること。

```
//必要なものをインクルード
```

```
//(1) 関数テンプレート（関数名 : max 関数）
```

```
int main() {  
    int a, b; double da, db;  
    cout << "Input two integers: "; cin >> a >> b;  
    cout << "Input two decimals: "; cin >> da >> db;  
  
    //テンプレート引数 a, b を int 型として受け取った max 関数を呼び出す  
    cout << "The maximum value of two integers is " << (2) << ", and\n";  
    //テンプレート引数 a, b を double 型として受け取った max 関数を呼び出す  
    cout << "that of two decimals is " << (3) << ".\n";  
    return 0;  
}
```

実行例 2-1

% ./ex10-2

Input two integers: 1 3

Input two decimals: 2.5 6.6

The maximum value of two integers is 3, and  
that of two decimals is 6.6.

※各実行例の下線はユーザ入力を示している。

※補足：「関数のオーバーロード」「関数テンプレート」は、異なる複数の処理を、同じ関数名を使って利用できるようにした機能だと言えるが（1 つの名前が状況に応じて別々の機能をもつことを多態性またはポリモーフィズムという）、それぞれは、使える局面が異なるため、それぞれの特徴に注意すること。

- 関数のオーバーロード： 関数内の処理手続きが異なってもよい。
- 関数テンプレート： 関数内の処理手続きはまったく同じである必要がある。扱う型だけが異なる場合に使うことができる。

【課題3】（提出期限 12月12日（日）23:59）

2つの1以上の整数a,bを引数として受け取り、b以上a以下のすべての整数の和を求める関数sumを以下の実行例に合うように関数宣言を用いて作成せよ。ただし、bに対する実引数が省略されて呼び出された場合は、bを1とみなして合計を求めること。また、main文は変更しないこととし、プログラム全体をex10-3.cppとして提出すること。

```
//必要なものをインクルード
```

```
//sum関数の関数宣言
```

```
int main() {  
    int x, y;  
    cout << "Integer x: "; cin >> x;  
    cout << "Integer y: "; cin >> y;  
  
    cout << "The sum of all numbers from 1 to y is " << sum(y) << ", and¥n";  
    cout << "that from x to y is " << sum(y,x) << "¥n";  
  
    return 0;  
}
```

```
//b以上a以下のすべての整数の合計を求めるsum関数の定義
```

実行例 3-1

```
% ./ex10-3
```

```
Integer x: 3
```

```
Integer y: 10
```

```
The sum of all numbers from 1 to y is 55, and  
that from x to y is 52.
```

※各実行例の下線はユーザ入力を示している。

※デフォルト実引数は、後ろ側の引数から順に連続して設定する必要があることに注意。