

DSAP HW2

B05703100 財金三 郭仲嘉

第一題

(a)

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

int getArrayElement(const int array[], int len, int index) throw(out_of_range);

int main()
{
    const int LEN = 10;

    int array[LEN] = {0};
    for(int i = 0; i < LEN; i++)
        cin >> array[i];
    int index = 0;
    cin >> index;

    try
    {
        int result = getArrayElement(array, LEN, index);
        cout << result << endl;
    }
    catch(out_of_range e)
    {
        cout << "Bad Index!";
    }

    return 0;
}

int getArrayElement(const int array[], int len, int index) throw(out_of_range)
{
    if (index < 0 or index >= len) throw out_of_range("...");
}
```

```
        return array[index];
    }
}
```

(b)

```
template <typename arrayType>
arrayType getArrayElement(const arrayType array[], int len, int index) throw(out_of_range);
```

```
//main function...
```

```
template <typename arrayType>
arrayType getArrayElement(const arrayType array[], int len, int index) throw(out_of_range)
{
    if (index < 0 or index >= len) throw out_of_range("...");
    return array[index];
}
```

(c)

```
template <typename arrayType>
arrayType getArrayElement(const arrayType array[], int len, int index) throw(out_of_range)
{
    if (index < -len + 1 or index >= len) throw out_of_range("...");
    if (index >= 0) return array[index];
    else return array[len - 1 + index];
}
```

(d)

1. 若要處理物件的 `array`，應該傳入該 `array` 的 `reference`。若直接傳該 `array`，函數會生成另一個物件 `array`，較費時且若沒寫 `copy constructor` 可能 `shallow copy`
2. 函數若要回傳物件，也要注意 `shallow copy` 問題
3. 若要以一個物件變數接住回傳值，要 `overload` 等號