

Homework 02

Question 3

for MEG304301

張皓鈞 B11030202



請以附上的 C++/Python 程式為例，
說明其中 assert 敘述的用法/用途，
及如何引用其概念在(防呆輸入)/避免
上犯檢查？

建議：
以 Q1Base 為基礎，加 assert，達成避免
"自己設計函式"的目的。

Code

C++

```
#include <iostream>
#include <cassert> // assert()
#include <cmath>    // abs()
#include <iomanip>  // setiosflags(), setprecision()
// URL:: https://www.cxybb.com/article/weixin_43379056/84983003

using namespace std;

int main()
{

    const double min_value = -1000;
    const double max_value = 1000;

    double x1, y1, x2, y2;
    cout << "Enter TWO points (Xi, yi), -1000 <= value <= 1000";
```

```

cout << "\n\tFOUR double values in ONE line: " << endl;
cin >> x1 >> y1 >> x2 >> y2;

assert(x1>=min_value && x1<=max_value);
assert(y1>=min_value && y1<=max_value);
assert(x2>=min_value && x2<=max_value);
assert(y2>=min_value && y2<=max_value);

double distance = abs(x1-x2) + abs(y1-y2);

cout << setiosflags(ios::fixed);
cout << setprecision(3) << distance << endl;

return 0;
}

```

Python

```

# MEGdemo_MDistance1..py for "Manhattan Distance-曼哈頓距離"
'''
#https://www.geeksforgeeks.org/taking-multiple-inputs-from-user-in-python/
# multiple input using split
'''

import math
a144 = math.sqrt(144.011)
# print (a144)

min_value , max_value = -1000, +1000

# Python function
def calcDistance ( px1, py1, px2, py2):
    dist = abs(float(px1)-float(px2)) +abs(float(py1)-float(py2))
    return dist
#
# single input ( return is a "string" )
x1 = float(input( "Enter point 1-X :"))
y1 = float(input( "Enter point 1-Y: "))
print(x1, y1, "type of x1=", type(x1))

# taking two inputs at a time
x2, y2 = input("Enter a two positive value(Boyes , Girls): ").split()
# ASSERT for python - send out error message

print("Type of X2-[Girls] is ", type(x2))
print("Number of boys: ", x2)
print("Number of girls: ", y2)

```

```

assert( float(x2) >0 ), "Boys numbers should be >0"
assert( float(y2) >0 ), "Girls numbers should be >0"
assert( float(x2) < max_value ), "Boys numbers should be <=1000"


zsum = float(x2) +float(y2)
print("SUM of two floating numbers =", zsum)
distance = abs(float(x1)-float(x2)) +abs(float(y1)-float(y2))
print("曼哈頓距離-Case1 :", distance)
dist2 = calcDistance( x1,y1,x2,y2)
print("Manhattan Distance: -Case2 :", dist2)
print()

def getTwoNumbers (message):
    # taking two inputs at a time
    ap, bp = input("Enter a two value: ").split()
    print(f"{message} - getTwoNumbers function. Got {ap} and {bp}")
    print("First number is {} and second number is {}".format(ap, bp))
    print()

## call it
getTwoNumbers("MEG304-曼哈頓:")


# taking multiple inputs at a time
# and type casting using list() function
x = list(map(int, input("Enter a multiple value: ").split()))
print("List of students: ", x)

```

說明

assert之意為斷言，指的是程式進行到某個時間點，斷定其必然是某種狀態，具體而言，也就是斷定該時間點上，某變數必然是某值，或某物件必具擁有何種特性值。

斷言測試Python中的用法為

```
assert <test>, <message>
```

test是狀態測試，而message是斷言失敗時所要呈現訊息。

斷言是用來斷定程式某個時間點的狀態，最基本的原則是，斷言執行前後，不可以改變任何程式狀態，也就是不可以產生任何邊際效應。斷言會在最佳化時被省略，也就是最後編譯出來的程式碼，不會包括assert陳述句，這可以在使用Python直譯器時，加上 `-O` 引數來達到，啟動Python直譯器時，若加上 `-O` 引數，則程式中**debug**會被設為False，也就不會包括或執行assert陳述句。

我認為assert可以用於判斷使用者輸入數值是否合法，但不建議，因為assert的設計目的只是用於debug期間的執行狀態斷定，會在最佳化時被省略，故不建議使用。