Homework 1 Report

Name	Student ID	Grades
楊英豪	B10803207	

Task 1: 三角形 (Basics -必做)

1. **輸入3個邊長**，整數值，須查驗不得為負值，並介於（1～M，M自定成100）之間。
2. 判斷是否可成為三角形(兩邊之和大於第三邊)。
3. 若判定成三角形成立，則計算並輸出該三角形面積（有小數點2位）。

```
#include <iostream>
using namespace std;
#include <cmath>
#include <cstdlib>
#include <iomanip>
#include <vector>
//楊英豪 B10803207

void calculateArea(int &x, int &y, int &z) {
  float s = (x + y + z) / 2;
  float area = sqrt(s * (s - x) * (s - y) * (s - z));
  cout << "These values can form a triangle!, "
       << "Area of triangle is " << fixed << setprecision(2) << area << "\n\n";
}

void isTriangle(int &x, int &y, int &z) {
  // checks if 3 sides can be a triangle
  if (x + y <= z || x + z <= y || y + z <= x) {
    cout << "These values can't form a triangle! \n\n";
  } else {
    calculateArea(x, y, z);
  }
}

void checkSideValues(int &x, int &y, int &z) {
  if (x > 99 || y > 99 || z > 99) {
    cout << "Value more than '100' detected, Please enter a value between "
            "0~100 \n";
  } else if (x < 0 || y < 0 || z < 0) {
    cout << "Negative value detected, Please enter a value between 0~100 \n\n";
  } else {
    isTriangle(x, y, z);
  }
}
```

```
void print(vector<int> const &random_number)
{
    for (auto it =random_number.cbegin(); it != random_number.cend(); it++) {
        cout << *it << ' ';
    }
}

void generateRandomSides(int &x, int &y, int &z) {
  srand(time(0));
  vector<int> random_number;
  for (int i = 0; i < 9; i++) {
    x = 1 + (rand() % 100);
    y = 1 + (rand() % 100);
    z = 1 + (rand() % 100);
    cout << x << ", " << y << ", " << z << "\n";
    checkSideValues(x, y, z);

    random_number.insert(random_number.end(), {x, y, z});
  }
  // print(random_number);
}


void manualInput(int &x, int &y, int &z) {
  for (int i = 0; i < 9; i++) {
    cout << "Loop " << i << " | Insert 3 sides of triangle:";
    cin >> x >> y >> z;
    checkSideValues(x, y, z);
  }
}

int main() {
  int x, y, z;
  int mode;
  cout << "Please Enter Mode by number: 1.Manual Input 2.Random" << endl;
  cin >> mode;
  if (mode == 1) {
    manualInput(x, y, z);
  } else {
    generateRandomSides(x, y, z);
  }
}
```

Output

This is how the manual input mode will look like.

main.cpp

```
isTriangle
1  #include <iostream>
2  using namespace std;
3  #include <cmath>
4  #include <cstdlib>
5  #include <iomanip>
6  #include <vector>
7  //楊英豪 B10803207
8
9  void calculateArea(int &x, int &y, int &z) {
10   float s = (x + y + z) / 2;
11   float area = sqrt(s * (s - x) * (s - y) * (s - z));
12   cout << "These values can form a triangle!, "
13        << "Area of triangle is " << fixed << setprecision(2) << area <<
     "\n\n";
14 }
15
16 void isTriangle(int &x, int &y, int &z) {
17   // checks if 3 sides can be a triangle
18   if (x + y <= z || x + z <= y || y + z <= x) {
19     cout << "These values can't form a triangle! \n\n";
20   } else {
21     calculateArea(x, y, z);
22   }
23 }
24
25 void checkSideValues(int &x, int &y, int &z) {
```

Console / Shell

```
> sh -c make -s
> ./main
Please Enter Mode by number: 1.Manual Input 2.Random
1
Loop 0 | Insert 3 sides of triangle:-1 4 5
Negative value detected, Please enter a value between 0~100

Loop 1 | Insert 3 sides of triangle:101 56 74
Value more than '100' detected, Please enter a value between 0~100

Loop 2 | Insert 3 sides of triangle:3 4 5
These values can form a triangle!, Area of triangle is 6.00

Loop 3 | Insert 3 sides of triangle:7 24 25
These values can form a triangle!, Area of triangle is 84.00

Loop 4 | Insert 3 sides of triangle:
```

TASK 2: 有技巧的解題方式 (skillful) - Recommended

The numbers here were generated randomly

generateRandomSides

```
1  #include <iostream>
2  using namespace std;
3  #include <cmath>
4  #include <cstdlib>
5  #include <iomanip>
6  #include <vector>
7  //楊英豪 B10803207
8
9  void calculateArea(int &x, int &y, int &z) {
10   float s = (x + y + z) / 2;
11   float area = sqrt(s * (s - x) * (s - y) * (s - z));
12   cout << "These values can form a triangle!, "
13        << "Area of triangle is " << fixed << setprecision(2) << area <<
     "\n\n";
14 }
15
16 void isTriangle(int &x, int &y, int &z) {
17   // checks if 3 sides can be a triangle
18   if (x + y <= z || x + z <= y || y + z <= x) {
19     cout << "These values can't form a triangle! \n\n";
20   } else {
21     calculateArea(x, y, z);
22   }
23 }
24
```

```
> sh -c make -s
> ./main
Please Enter Mode by number: 1.Manual Input 2.Random
2
2, 4, 59
These values can't form a triangle!

39, 5, 65
These values can't form a triangle!

72, 75, 7
These values can form a triangle!, Area of triangle is 232.16

74, 86, 16
These values can form a triangle!, Area of triangle is 421.20

76, 62, 22
These values can form a triangle!, Area of triangle is 578.00

32, 84, 31
These values can't form a triangle!

76, 67, 31
These values can form a triangle!, Area of triangle is 1035.30

14, 9, 11
These values can form a triangle!, Area of triangle is 49.48

22, 40, 76
These values can't form a triangle!

>
```

Output

```
tructure *my1_p = &my1;
1x1 = 0;
1y1 = 0;
nber = 0;
ndistance = 0;

tance;
 y1;
 = 0, y2 = 0;

tu2022 外送員訂單:: Get "
rder:\n";

nter (X,Y) location below: " << endl;
 nCount) {
 ++i << " 訂單(X,Y)> ";
1 >> y1;
 = abs(x1 - x2) + abs(y1 - y2);

 0 && y1 == 0) {

a : packing "minx1, miny1, number,mindistance" as a structure
b : call a refactored function. pass-by-reference
ance, x1, y1, &my1, i);

"曼哈頓距離= " << distance << " at (" << x1 << "   " << y1 << ")\n"
```

```
> sh -c make -s
> ./main
stu2022 外送員訂單:: Get Order:
Enter (X,Y) location below:
1 訂單(X,Y)> 10 20
曼哈頓距離= 30 at (10 , 20)

2 訂單(X,Y)> 20 30
曼哈頓距離= 50 at (20 , 30)

3 訂單(X,Y)> 5 20
曼哈頓距離= 25 at (5 , 20)

4 訂單(X,Y)> 10 10
曼哈頓距離= 20 at (10 , 10)

5 訂單(X,Y)> 20 20
曼哈頓距離= 40 at (20 , 20)

6 訂單(X,Y)> 1 2
曼哈頓距離= 3 at (1 , 2)

7 訂單(X,Y)> 3
4
曼哈頓距離= 7 at (3 , 4)

8 訂單(X,Y)> 5 6
曼哈頓距離= 11 at (5 , 6)

9 訂單(X,Y)> 8 9
曼哈頓距離= 17 at (8 , 9)

10 訂單(X,Y)> 19 2
曼哈頓距離= 21 at (19 , 2)


最短距離為6訂單(X,Y)>  (1,2)
距離為 3>
```

TASK 3 : 曼哈頓距離（HW0）一必做

```
#include <cmath>
#include <iostream>
using namespace std;
// optional : remove the parameter name, just keep its type :: for simplifying &
// convention

struct myStructure {
  double minx1;
  double miny1;
  int number;
  double mindistance;
};

void min(double distance, double x1, double y1, struct myStructure *my1_p,
         int i);

int main() {
  int nCount = 10;
  struct myStructure my1;
  struct myStructure *my1_p = &my1;
  my1_p->minx1 = 0;
  my1_p->miny1 = 0;
  my1_p->number = 0;
  my1_p->mindistance = 0;

  double distance;
  double x1, y1;
  double x2 = 0, y2 = 0;
  int i = 0;

  cout << "stu2022 外送員訂單:: Get "
       << "Order:\n";

  cout << "Enter (X,Y) location below: " << endl;
  while (i < nCount) {
    cout << ++i << " 訂單(X,Y)> ";
    cin >> x1 >> y1;
    distance = abs(x1 - x2) + abs(y1 - y2);

    if (x1 == 0 && y1 == 0) {

      break;
    }
    // TODO2a : packing "minx1, miny1, number,mindistance" as a structure
    // TODO2b : call a refactored function. pass-by-reference
    min(distance, x1, y1, &my1, i);

    cout << "曼哈頓距離= " << distance << " at (" << x1 << " , " << y1 << ")\n"
         << endl;
  }

  if (i == 1) { // Q: why ?  more intuitive way is needed.
    return 0;
```

```
  }

  cout << "\n最短距離為" << my1_p->number << "訂單(X,Y)> "
       << " (" << my1_p->minx1 << "," << my1_p->miny1 << ")" << endl;
  cout << "距離為 " << my1_p->mindistance;
}

void min(double distance, double x1, double y1, struct myStructure *my1_p,
         int i) {
  if (distance < my1_p->mindistance || i == 1) {
    my1_p->minx1 = x1;
    my1_p->miny1 = y1;
    my1_p->mindistance = distance;
    my1_p->number = i;
  }
}
```

Task 4 大哉問 (Optional)

Q1 - 差異 int[] vs. int*

Q1::宣告[int a或int* a],a=100及 [int t1,t2,t3或int* t1,t2,t3],觀察以下3種情況是否出錯?

```
#include <iostream>
using namespace std;
int main() {
    int a=100;
    int t1;
    t1 = a + 1;
    cout<<t1;
}
```

ANSWER: t1 would not have any problem using no pointers as it would be a pass-by-value method, but there will be an error if you add pointer to the declaration and initialization of variable a

```
#include <iostream>
using namespace std;
    int main() {
    int a=100;
    int *t2;
    t2 = &a + 1;
    cout<<t2;
}
```

ANSWER: Here we need to declare that t2 is a pointer, so that during that operation when we use t2 it will be considered as the address rather than the value, &a refers to the address of the value a, so if t2 is not declared as a pointer, there will be a pointer and integer type error

```
#include <iostream>
using namespace std;
    int main() {
    int a=100;
    int t3;
    t3 = *&a + 1;
    cout<<t3;
}
```

ANSWER: Here we need to add an address before the variable, so that it translates to "point to the value of the address of a" , without the ampersand symbol there will be a pointer error

Q2: (講義) SWAP 函式差異

```
Q2::W3講義第一頁的swap2A及swap2B 有甚麼差異? (Function definition & its call usage)
```

Difference is during the function body, 2A use pointers while 2B uses array, both are similar. The difference is pointer points to a location in a memory using address, while array refers to the place of the first element in the array

Q3: function 參數用法差異 - int* vs. int[]

You can change the value of the pointer (int*) and sizeof(array) return the whole size of the array not just size of the pointer. But when it is used as function arguments then they are both viewed as int

#心得

I have never wrote any C++ code before this class so having this experience was kind of cool. There were a lot of small detailed things that I never think of when programming in other languages such as pass-by-reference, because usually pass-by-value is the most commonly used one. Knowing the difference between vector and arrays is also surprising, as when I'm using python it's usually a list