Project Title: Inheritance and Combination of the Class Graph

1. Requirement Analysis

Write a C++ program to allow users to draw different graphs on a specified location, such as upright isosceles triangle, inverted isosceles triangle, rectangle, square, diamond, and polygon. The graphs are made up of randomly selected symbol “*” or “#”. The so-called specified location means the number of characters between the leftmost character of graph and the left border of screen. (Note that due to the asymmetry of size and gap of characters, the square pattern may be not standard.). The programming requirements are listed as follows:

- Design a class for each graph, and the relationship between classes may be inheritance and combination.
- Each class can store the data members of graph objects, and has member functions to change the data members and to print graphs with characters on screen.
- According to user’s choice, the program can create a new graph object, set initial values for this object, and print it out.

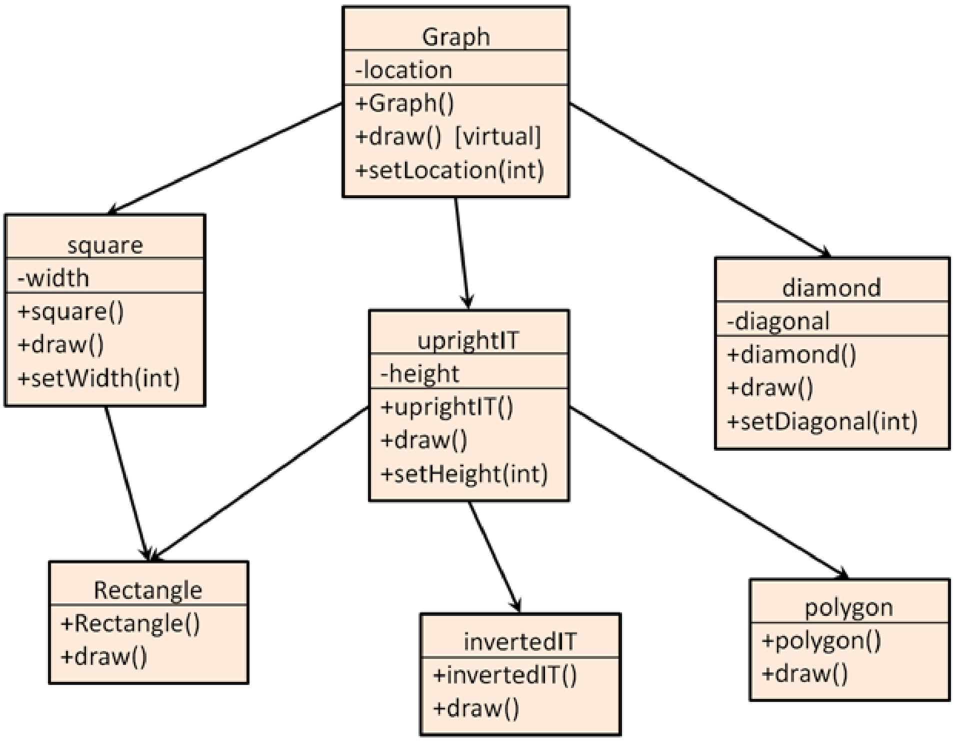
For example, a run-time result for this program is shown as follows:

```
Input 1 to choose upright isosceles triangle;
Input 2 to choose inverted isosceles triangle;
Input 3 to choose rectangle;
Input 4 to choose square;
Input 5 to choose diamond;
Input 6 to choose polygon;
Input 0 to exit.

Your choice (0~6): 1

Please input height: 3
Please input location: 1

   #
  #**
 #****

Your choice (0~6): 2

Please input height: 4
Please input location: 2
```

```
  #*###*#
   #*###*
    ###
     *

Your choice (0~6): 5

Please input diagonal (must be odd number): 5
Please input location: 0

  #
 **#
*****
 #*#
  *

Your choice (0~6): 3

Please input height: 4
Please input width: 10
Please input location: 3

   **##*##***
   ***#**#*##
   ***#####**
   *#*#*##***

Your choice (0~6): 4

Please input width: 6
Please input location: 0

##**#*
****##
#####*
#*#**#
**#**#
#*#*#*

Your choice (0~6): 6

Please input height: 3
Please input location: 0

  #
  #**
#****
##**#
****#
#####
```

```
##**#

Your choice (0~6): 0
```

2. Design Description



3. Debug Analysis

1. For the robustness and correctness of the program, troublesome conditions should be carefully taken into consideration. For instance, while prompting for a menu selection from the user, if a meaningless number is inputted, the program should stop to ask again until he inputs a correct one.

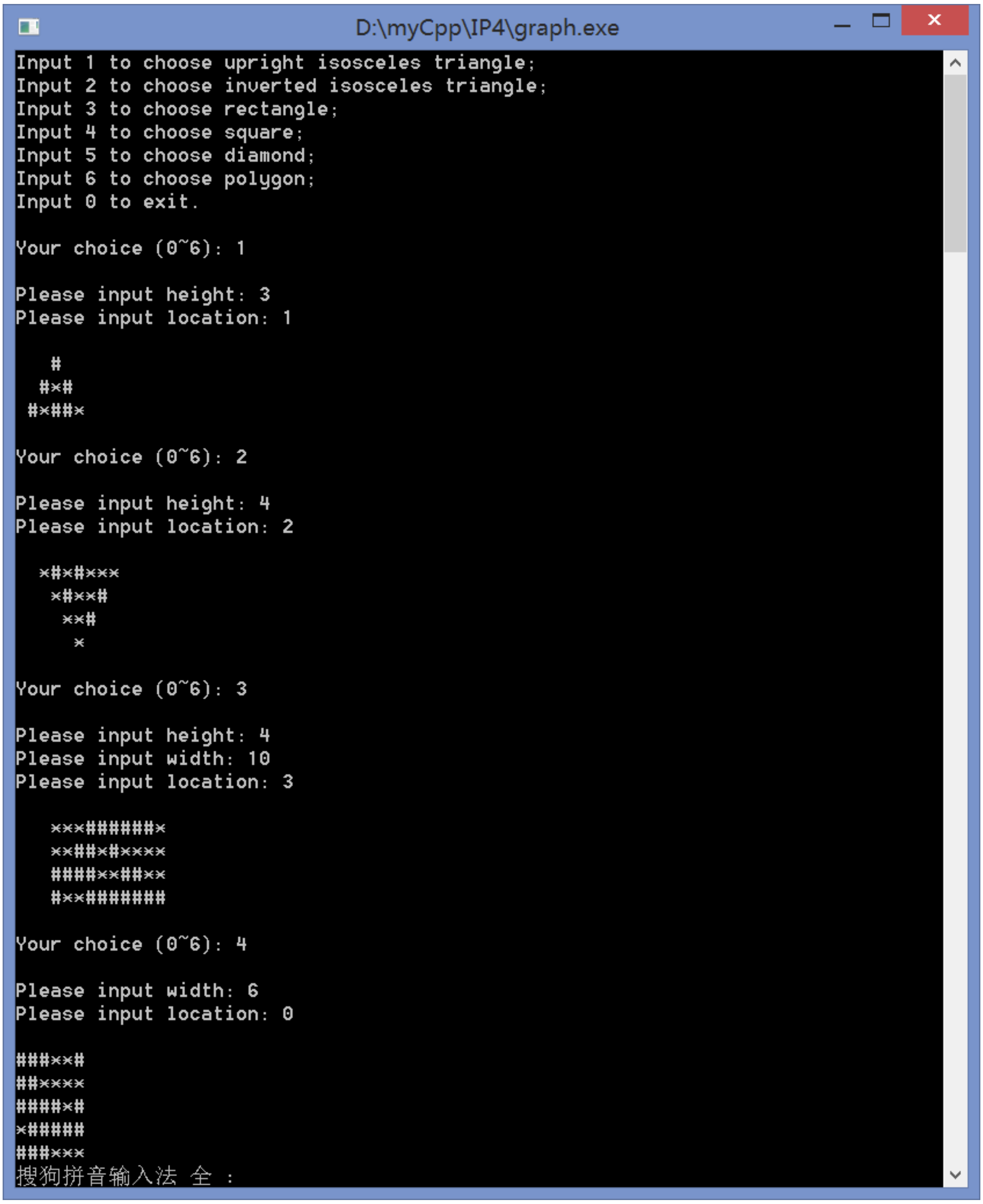
2. While using virtual functions, the explicit implement should be exactly the same with function declaration in the base class, otherwise it would be assumed as another function, and the derived class would be considered an abstract class.

3. Under the circumstance where a class is derived from 2 different classes which share a common base class, virtual inheritance is required in order to avoid the member of the new class from getting ambiguous.

4. Test Result

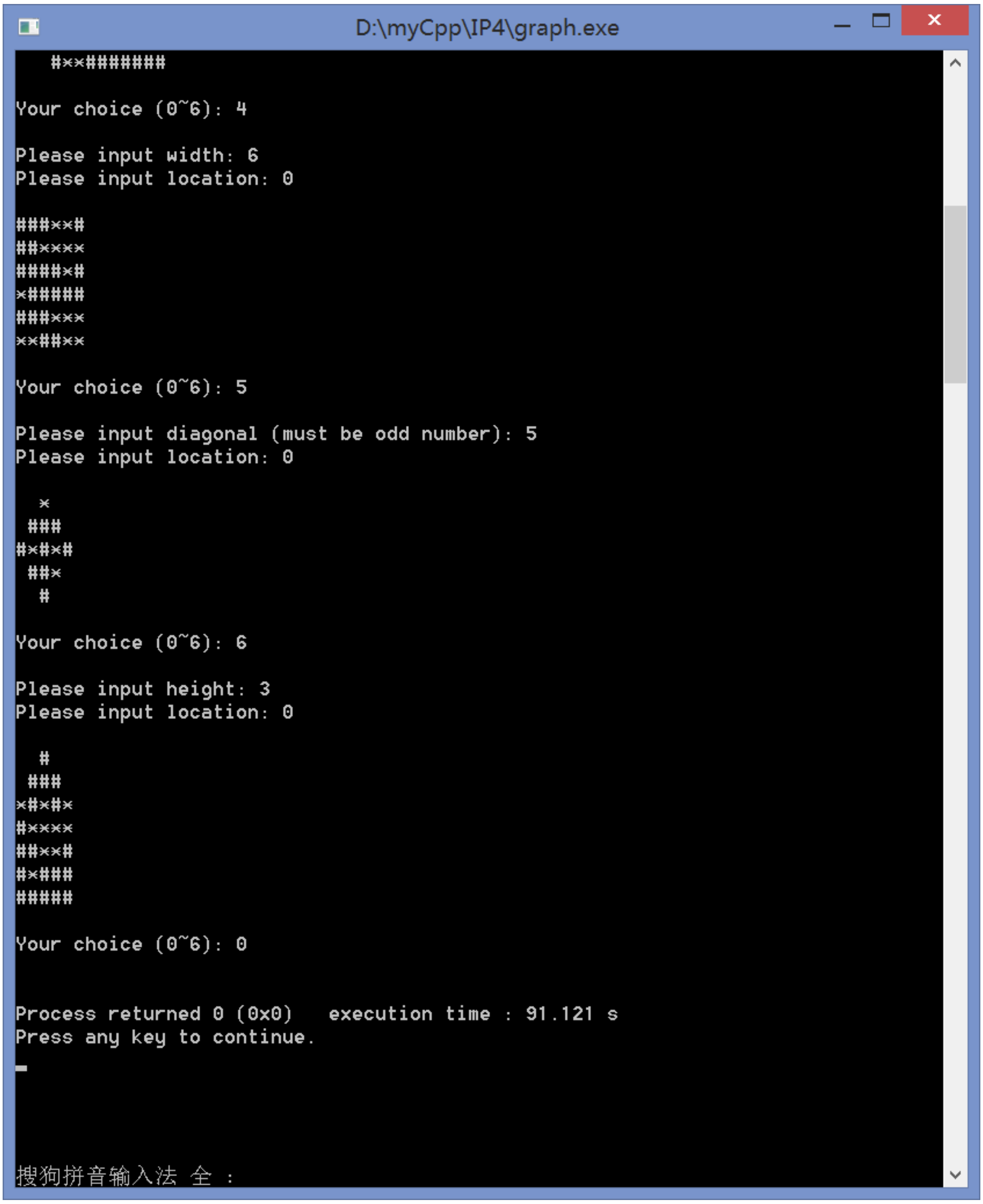
The screenshots are a bit big and each consumes a whole page.

a)



```
Input 1 to choose upright isosceles triangle;
Input 2 to choose inverted isosceles triangle;
Input 3 to choose rectangle;
Input 4 to choose square;
Input 5 to choose diamond;
Input 6 to choose polygon;
Input 0 to exit.

Your choice (0~6): 1

Please input height: 3
Please input location: 1

   #
  #*#
 #*##*

Your choice (0~6): 2

Please input height: 4
Please input location: 2

  *#*#***
   *#**#
    **#
     *

Your choice (0~6): 3

Please input height: 4
Please input width: 10
Please input location: 3

   ***######*
   **##*#****
   ####**##**
   #**#######

Your choice (0~6): 4

Please input width: 6
Please input location: 0

###**#
##****
####*#
*#####
###***
搜狗拼音输入法 全 ：
```

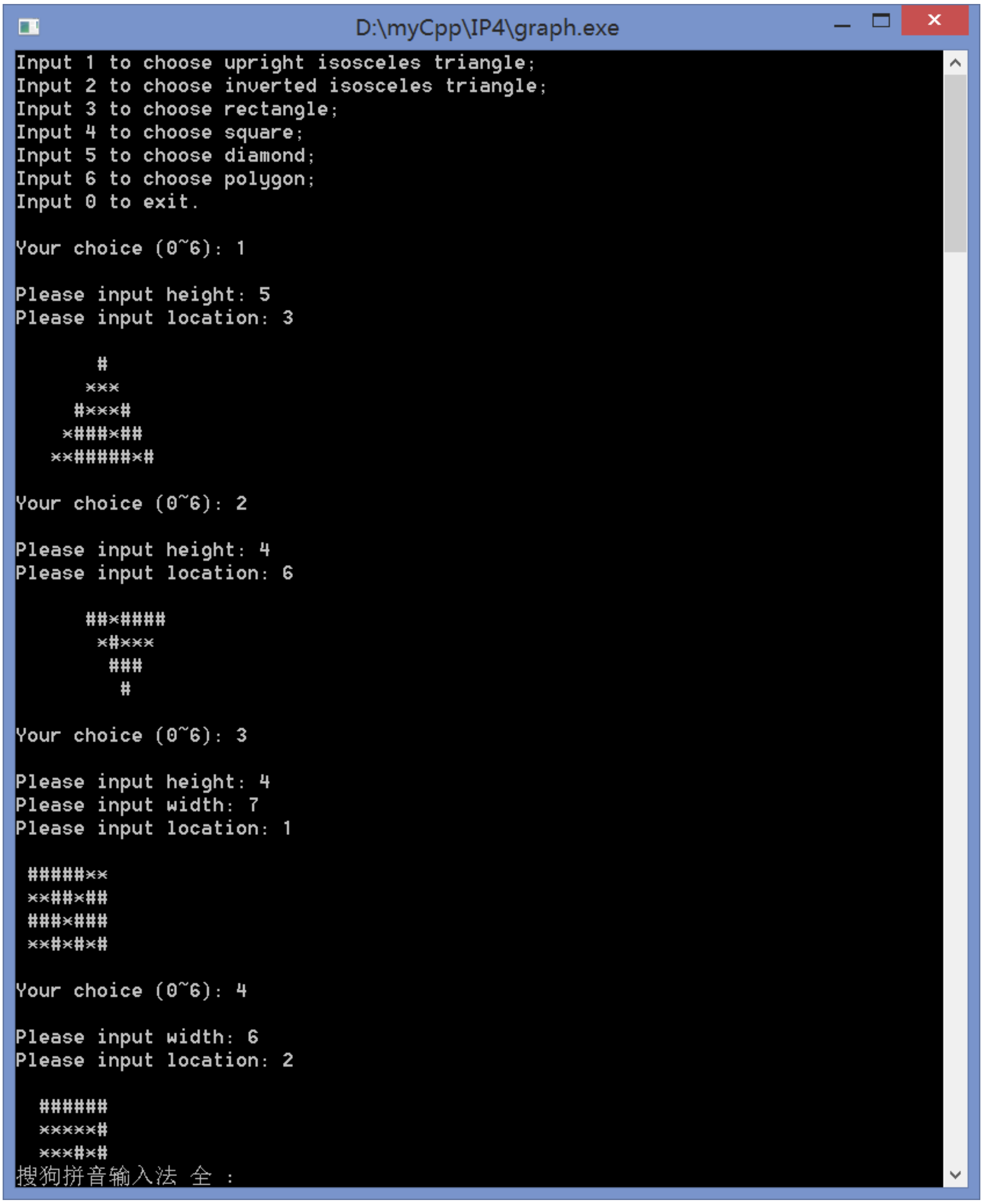
a_1



```
   #**#######

Your choice (0~6): 4

Please input width: 6
Please input location: 0

###**#
##****
####*#
*#####
###***
**##**

Your choice (0~6): 5

Please input diagonal (must be odd number): 5
Please input location: 0

  *
 ###
#*#*#
 ##*
  #

Your choice (0~6): 6

Please input height: 3
Please input location: 0

  #
 ###
*#*#*
#****
##**#
#*###
#####

Your choice (0~6): 0


Process returned 0 (0x0)   execution time : 91.121 s
Press any key to continue.
```

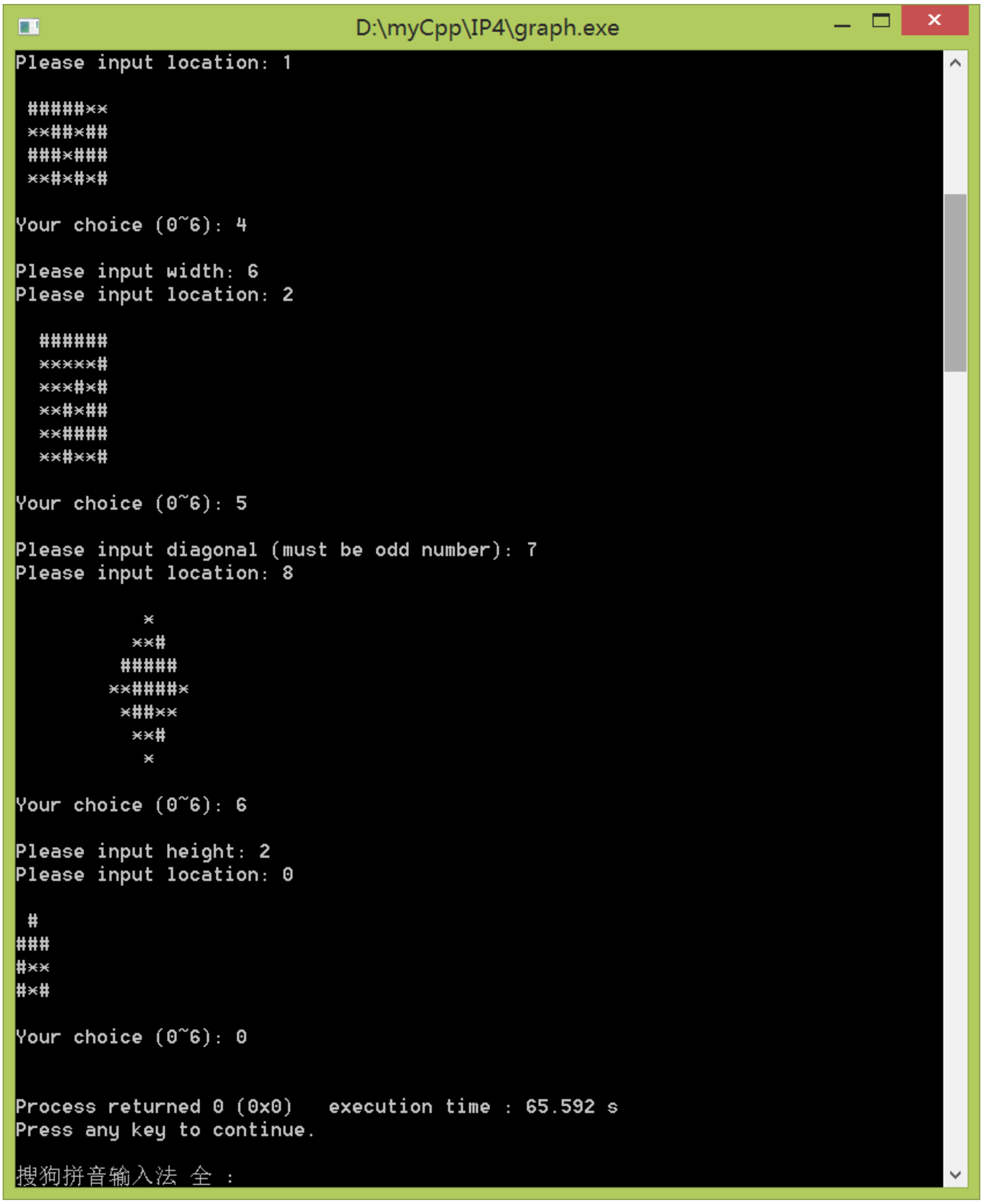
a_2

b)



```
Input 1 to choose upright isosceles triangle;
Input 2 to choose inverted isosceles triangle;
Input 3 to choose rectangle;
Input 4 to choose square;
Input 5 to choose diamond;
Input 6 to choose polygon;
Input 0 to exit.

Your choice (0~6): 1

Please input height: 5
Please input location: 3

       #
      ***
     #***#
    *###*##
   **#####*#

Your choice (0~6): 2

Please input height: 4
Please input location: 6

      ##*####
       *#***
        ###
         #

Your choice (0~6): 3

Please input height: 4
Please input width: 7
Please input location: 1

 #####**
 **##*##
 ###*###
 **#*#*#

Your choice (0~6): 4

Please input width: 6
Please input location: 2

  ######
  *****#
  ***#*#
搜狗拼音输入法 全 :
```

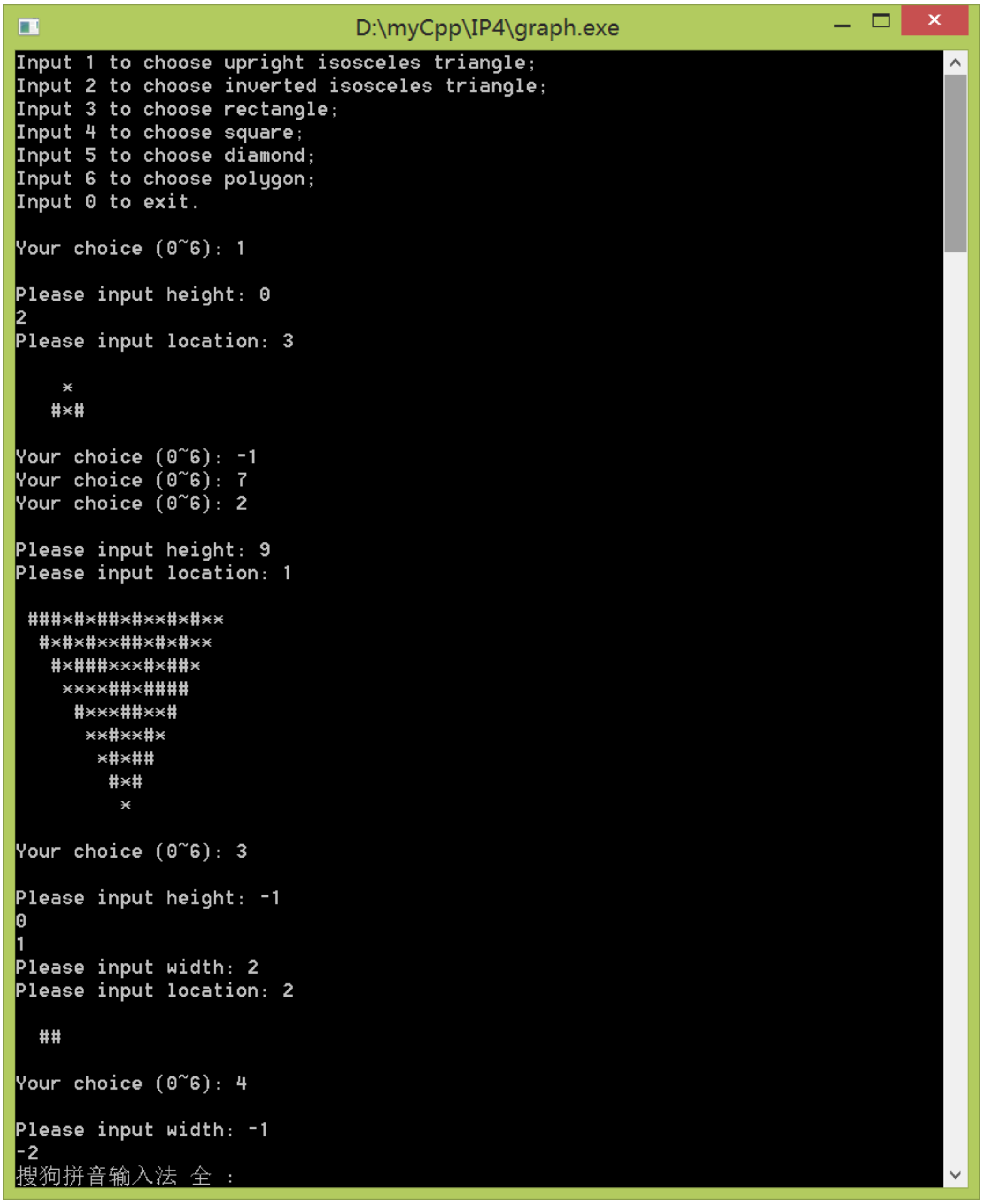
b_1



```
Please input location: 1

 #####**
 **##*##
 ###*###
 **#*#*#

Your choice (0~6): 4

Please input width: 6
Please input location: 2

  ######
  *****#
  ***#*#
  **#*##
  **####
  **#**#

Your choice (0~6): 5

Please input diagonal (must be odd number): 7
Please input location: 8

           *
          **#
         #####
        **####*
         *##**
          **#
           *

Your choice (0~6): 6

Please input height: 2
Please input location: 0

 #
###
#**
#*#

Your choice (0~6): 0


Process returned 0 (0x0)   execution time : 65.592 s
Press any key to continue.

搜狗拼音输入法 全 ：
```

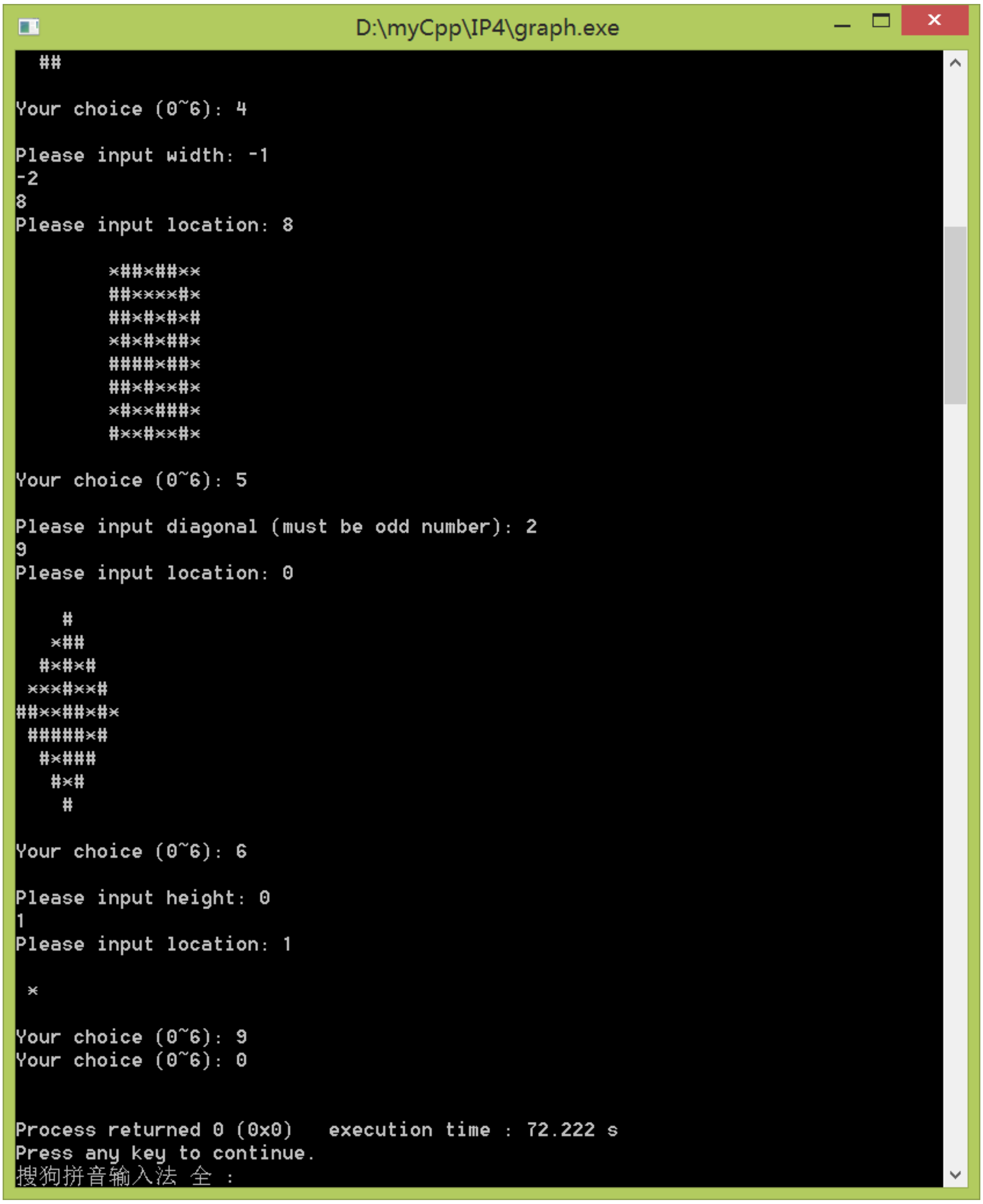
b_2

c)



```
Input 1 to choose upright isosceles triangle;
Input 2 to choose inverted isosceles triangle;
Input 3 to choose rectangle;
Input 4 to choose square;
Input 5 to choose diamond;
Input 6 to choose polygon;
Input 0 to exit.

Your choice (0~6): 1

Please input height: 0
2
Please input location: 3

    *
   #*#

Your choice (0~6): -1
Your choice (0~6): 7
Your choice (0~6): 2

Please input height: 9
Please input location: 1

 ###*#*##*#**#*#**
  #*#*#**##*#*#**
   #*###***#*##*
    ****##*####
     #***##**#
      **#**#*
       *#*##
        #*#
         *

Your choice (0~6): 3

Please input height: -1
0
1
Please input width: 2
Please input location: 2

  ##

Your choice (0~6): 4

Please input width: -1
-2
搜狗拼音输入法 全 ：
```

c_1



```
  ##

Your choice (0~6): 4

Please input width: -1
-2
8
Please input location: 8

        *##*##**
        ##****#*
        ##*#*#*#
        *#*#*##*
        ####*##*
        ##*#**#*
        *#**###*
        #**#**#*

Your choice (0~6): 5

Please input diagonal (must be odd number): 2
9
Please input location: 0

    #
   *##
  #*#*#
 ***#**#
##**##*#*
 #####*#
  #*###
   #*#
    #

Your choice (0~6): 6

Please input height: 0
1
Please input location: 1

 *

Your choice (0~6): 9
Your choice (0~6): 0


Process returned 0 (0x0)   execution time : 72.222 s
Press any key to continue.
搜狗拼音输入法 全 ：
```

c_2

5. Appendix

The source code file is packed together with this report in the compressed files. However, the code as well as annotation is still provided in this report in case.

```
//File name:graph.cpp
//This program accomplishes and tests the application of the class Graph and its descendants..
#include <iostream>    //For console input and output.
#include <cstdlib>    //To generate a random number.
#include <ctime>    //To provide the random number generator with a seed.
using namespace std;
void randPrint();    //To print a character randomly chosen from '*' and '#'.
class Graph    //The base class Graph
{
protected:
	int location;
public:
	Graph():location(0){};
	virtual void draw()const=0;    //It will be explicitly defined in each derived class.
	void setLocation(int n){location=n;}
};
class uprightIT:virtual public Graph    //Upright isosceles triangle
{
protected:
	int height;
public:
	uprightIT():Graph(),height(0){}
	void draw() const;
	void setHeight(int n){height=n;}
};
class invertedIT:public uprightIT    //Inverted isosceles triangle
{
public:
	invertedIT():uprightIT(){}
	void draw() const;
};
class square:virtual public Graph    //Square
{
protected:
	int width;
public:
	square():Graph(),width(0){}
	void draw() const;
```

```
    void setWidth(int n){width=n;}
};
class rectangle:public uprightIT,public square    //Rectangle
{
public:
    rectangle():Graph(),uprightIT(),square(){}
    void draw() const;
};
class polygon:public uprightIT    //Polygon
{
public:
    polygon():uprightIT(){}
    void draw() const;
};
class diamond:public Graph    //Diamond
{
protected:
    int diagonal;
public:
    diamond():Graph(),diagonal(0){}
    void draw() const;
    void setDiagonal(int n){diagonal=n;}
};
void printMenu();
int main()
{
    srand(time(NULL));
    int s,x;
    uprightIT g1;
    invertedIT g2;
    rectangle g3;
    square g4;
    diamond g5;
    polygon g6;
    printMenu();
    while (true)
    {
        cout<<"Your choice (0~6): ";
        cin>>s;
        while ((s<0)||(s>6))
        {
            cout<<"Your choice (0~6): ";
            cin>>s;
        }
```

```
cout<<endl;    //To prompt a selection
switch (s)
{
        case 1:cout<<"Please input height: ";
                      do{cin>>x;}while (x<=0);
                      g1.setHeight(x);
                      cout<<"Please input location: ";
                      do{cin>>x;}while (x<0);
                      g1.setLocation(x);
                      cout<<endl;
                      g1.draw();
                      cout<<endl;
                      break;
        case 2:cout<<"Please input height: ";
                      do{cin>>x;}while (x<=0);
                      g2.setHeight(x);
                      cout<<"Please input location: ";
                      do{cin>>x;}while (x<0);
                      g2.setLocation(x);
                      cout<<endl;
                      g2.draw();
                      cout<<endl;
                      break;
        case 3:cout<<"Please input height: ";
                      do{cin>>x;}while (x<=0);
                      g3.setHeight(x);
                      cout<<"Please input width: ";
                      do{cin>>x;}while (x<=0);
                      g3.setWidth(x);
                      cout<<"Please input location: ";
                      do{cin>>x;}while (x<0);
                      g3.setLocation(x);
                      cout<<endl;
                      g3.draw();
                      cout<<endl;
                      break;
        case 4:cout<<"Please input width: ";
                      do{cin>>x;}while (x<=0);
                      g4.setWidth(x);
                      cout<<"Please input location: ";
                      do{cin>>x;}while (x<0);
                      g4.setLocation(x);
                      cout<<endl;
                      g4.draw();
```

```
                   cout<<endl;
                   break;
            case 5:cout<<"Please input diagonal (must be odd number): ";
                   do{cin>>x;}while ((x<=0)||(x%2==0));
                   g5.setDiagonal(x);
                   cout<<"Please input location: ";
                   do{cin>>x;}while (x<0);
                   g5.setLocation(x);
                   cout<<endl;
                   g5.draw();
                   cout<<endl;
                   break;
            case 6:cout<<"Please input height: ";
                   do{cin>>x;}while (x<=0);
                   g6.setHeight(x);
                   cout<<"Please input location: ";
                   do{cin>>x;}while (x<0);
                   g6.setLocation(x);
                   cout<<endl;
                   g6.draw();
                   cout<<endl;
                   break;
            default:break;
        }   //Conduct different orders accordingly
        if (s==0) break;
    }
}
void printMenu()
{
    cout<<"Input 1 to choose upright isosceles triangle;\n";
    cout<<"Input 2 to choose inverted isosceles triangle;\n";
    cout<<"Input 3 to choose rectangle;\n";
    cout<<"Input 4 to choose square;\n";
    cout<<"Input 5 to choose diamond;\n";
    cout<<"Input 6 to choose polygon;\n";
    cout<<"Input 0 to exit.\n\n";
}   //To print the selection menu
void randPrint()
{
    if (rand()%2) cout<<'#';else cout<<'*';
}   ////To print a character randomly chosen from '*' and '#'.
void uprightIT::draw() const
{
    int i,j;
    for (i=0;i<height;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<height-i-1;++j) cout<<' ';
        for (j=0;j<2*i+1;++j) randPrint();
        cout<<endl;
    }
}   //Draw the upright isosceles triangle
void invertedIT::draw() const
{
    int i,j;
    for (i=0;i<height;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<i;++j) cout<<' ';
        for (j=0;j<2*height-2*i-1;++j) randPrint();
        cout<<endl;
    }
}   //Draw the inverted isosceles triangle
void rectangle::draw() const
{
    int i,j;
    for (i=0;i<height;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<width;++j) randPrint();
        cout<<endl;
    }
}   //Draw the rectangle
void square::draw() const
{
    int i,j;
    for (i=0;i<width;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<width;++j) randPrint();
        cout<<endl;
    }
}   //Draw the square
void polygon::draw() const
{
    int i,j;
    for (i=0;i<height-1;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<height-i-1;++j) cout<<' ';
        for (j=0;j<2*i+1;++j) randPrint();
        cout<<endl;
    }
    for (i=0;i<2*height-1;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<2*height-1;++j) randPrint();
        cout<<endl;
    }
}   //Draw the Polygon
void diamond::draw() const
{
    int i,j,h=diagonal/2;
    for (i=0;i<h;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<h-i;++j) cout<<' ';
        for (j=0;j<2*i+1;++j) randPrint();
        cout<<endl;
    }
    for (j=0;j<location;++j) cout<<' ';
    for (i=0;i<2*h+1;++i) randPrint();
    cout<<endl;
    for (i=0;i<h;++i)
    {
        for (j=0;j<location;++j) cout<<' ';
        for (j=0;j<i+1;++j) cout<<' ';
        for (j=0;j<2*h-2*i-1;++j) randPrint();
        cout<<endl;
    }
}   //Draw the diamond
```