**\*\*\*\*大学数学与计算机科学学院**

**实践能力系列1**

**姓名：\*\*\* 学号：2307030\*\* 班级：计算机科学与技术**

**字符图形绘制系统**，在main函数中根据不同的输入选择，绘制不同形状的字符图形，绘制不同的图形时，根据输入参数来决定图形的大小和厚度等。例如：

1. **a.绘制直角等腰三角形（正立）**

输入高度（宽度）和厚度，请注意合法值检查

高度（宽度）=8， 厚度=2，图形为：

##

##

###

####

## ##

## ##

##### ##

###### ##

高度（宽度）=5， 厚度=1，图形为：

#

##

# #

# #

#####

高度（宽度）=2， 厚度=10, 参数非法

**2）b.绘制直角等腰三角形（倒立）**

输入高度（宽度）和厚度，请注意合法值检查

高度（宽度）=8， 厚度=2，图形为：

########

########

## ##

## ##

####

###

##

##

高度（宽度）=5， 厚度=1，图形为：

#####

# #

# #

# #

#

高度（宽度）=2， 厚度=10, 参数非法, 高度（宽度）必须大于等于厚度

高度（宽度）=-2， 厚度=10, 参数非法, 高度（宽度）必须大于0

**3）绘制正方形**

输入高度（宽度）和厚度，请注意合法值检查

高度（宽度）=6， 厚度=2，图形为：

######

######

## ##

## ##

######

######

高度（宽度）=5， 厚度=1，图形为：

#####

# #

# #

# #

#####

高度（宽度）=-2， 厚度=3, 参数非法

**4）绘制长方形**

输入高度（宽度）和厚度，请注意合法值检查

高度=6， 宽度=9 ， 厚度=2，图形为：

#########

#########

## ##

## ##

#########

#########

高度=5，宽度=6， 厚度=1 ，图形为：

######

# #

# #

# #

######

高度=16， （宽度）=2，厚度=3, 参数非法

**5）绘制正立金字塔**

输入高度和层数，请注意合法值检查。

高度=3 层数=2

/\

/ \

/\_\_\_\_\

/\ /\

/ \ / \

/\_\_\_\_\/\_\_\_\_\

高度=3 层数=3

/\

/ \

/\_\_\_\_\

/\ /\

/ \ / \

/\_\_\_\_\/\_\_\_\_\

/\ /\ /\

/ \ / \ / \

/\_\_\_\_\/\_\_\_\_\ /\_\_\_\_\

**6）其他图形，如a）实心三角形，菱形，长方形等 b）圣诞树等**

**1.（部分）源代码：**

**#include<stdio.h>**

**#include <stdlib.h>**

**#include<windows.h>**

**#include<graphics.h>**

**#include<conio.h>**

**int inputInt(int low, int high, const char description[30]);**

**void Draw\_diammond();**

**void Draw\_front\_lr\_Triangle();**

**void Draw\_inverted\_lr\_Triangle();**

**void Draw\_Square();**

**void Draw\_Rectangle();**

**void Draw\_pyramid();**

**void Draw\_cordiform();**

**void Draw\_Chrustmas\_trees();**

**void Draw\_Stare();**

**void Draw\_playground();**

**void main()**

**//void x()**

**{**

**while (1)**

**{**

**int choice = 0;**

**printf("1.菱形\t 2.直角等腰三角形（正立）\t3.直角等腰三角形（倒立） 4.正方形\n5.长方形 6.正立金字塔\t\t\t7.爱心\t\t\t 8.圣诞树\n9.五角星 10.操场（彩绘）\t0.退出\n请选择要绘制的图形：");**

**choice = inputInt(0, 10, "选项");**

**switch (choice)**

**{**

**case 0:exit(0); break;**

**case 1:Draw\_diammond(); break;**

**case 2:Draw\_front\_lr\_Triangle(); break;**

**case 3:Draw\_inverted\_lr\_Triangle(); break;**

**case 4:Draw\_Square(); break;**

**case 5:Draw\_Rectangle(); break;**

**case 6:Draw\_pyramid(); break;**

**case 7:Draw\_cordiform(); break;**

**case 8:Draw\_Chrustmas\_trees(); break;**

**case 9:Draw\_Stare(); break;**

**case 10:Draw\_playground(); break;**

**default:break;**

**}**

**system("pause");**

**system("cls");**

**}**

**}**

**int inputInt(int low, int high, const char description[30])**

**{**

**int temp, flag = 0;**

**flag = scanf\_s("%d", &temp);**

**while (flag == 0 || temp<low || temp>high)**

**{**

**printf("输入%s必须大于等于%d且小于等于%d\n请重新输入：", description, low, high);**

**fflush(stdin);**

**flag = scanf\_s("%d", &temp);**

**}**

**return temp;**

**}**

**void Draw\_diammond()**

**{**

**int size;**

**printf("请输入菱形的大小：1 3 5 7 ...:");**

**size = inputInt(3, 199, "大小");**

**for (int i = 0; i <= size / 2; i++)**

**{**

**for (int j = 0; j <= size / 2 - i; j++)printf(" ");**

**for (int k = 1; k <= i \* 2 + 1; k++)printf("#");**

**printf("\n");**

**}**

**for (int i = 1; i <= size / 2; i++)**

**{**

**for (int j = 0; j <= i; j++)printf(" ");**

**for (int k = 1; k <= size - i \* 2; k++)printf("#");**

**printf("\n");**

**}**

**}**

**void Draw\_front\_lr\_Triangle()**

**{**

**int highness = 2, thickness = 1;**

**do**

**{**

**if (highness < thickness)printf("厚度必须小于高度！\n");**

**printf("请输入等腰直角三角形的高（大小限制在2-120）：");**

**highness = inputInt(2, 120, "高（宽）");**

**printf("请输入等腰直角三角形的厚度（大小限制在1-5）：");**

**thickness = inputInt(1, 5, "厚度");**

**} while (highness < thickness);**

**for (int i = 1; i <= highness; i++)**

**{**

**int k = 1;**

**if (i <= thickness) {**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else if (i + thickness <= highness)**

**{**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**while (k + thickness <= i)**

**{**

**printf(" ");**

**k++;**

**}**

**while (k <= i)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else**

**{**

**while (k <= i)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_inverted\_lr\_Triangle()**

**{**

**int highness = 2, thickness = 0;**

**do**

**{**

**if (highness < thickness)printf("厚度必须小于高度！\n");**

**printf("请输入等腰直角三角形的高度（大小限制在2-120）：");**

**highness = inputInt(2, 120, "高（宽）");**

**printf("请输入等腰直角三角形的厚度（大小限制在1-5）：");**

**thickness = inputInt(1, 5, "厚度");**

**} while (highness < thickness);**

**for (int i = highness; i >= 1; i--)**

**{**

**int k = 1;**

**if (i <= thickness) {**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else if (i + thickness <= highness)**

**{**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**while (k + thickness <= i)**

**{**

**printf(" ");**

**k++;**

**}**

**while (k <= i)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else**

**{**

**while (k <= i)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_Square()**

**{**

**int size = 2, thickness = 1;**

**do**

**{**

**if (size < thickness)printf("厚度必须小于边长！\n");**

**printf("请输入正方形的边长（大小限制在2-120）：");**

**size = inputInt(2, 120, "边长");**

**printf("请输入正方形的厚度（大小限制在1-6）：");**

**thickness = inputInt(1, 6, "厚度");**

**} while (thickness > size);**

**int i, k;**

**for (i = 1; i <= size; i++)**

**{**

**k = 1;**

**if (i <= thickness)**

**{**

**while (k <= size)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else if (i + thickness <= size)**

**{**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**while (k + thickness <= size)**

**{**

**printf(" ");**

**k++;**

**}**

**while (k <= size)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else**

**{**

**while (k <= size)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_Rectangle()**

**{**

**int highness = 2, width = 1, thickness = 1;**

**do**

**{**

**if (highness < thickness || width < thickness)printf("长度和宽度必须都小于厚度！\n");**

**printf("请输入长方形的长度（大小限制在2-120）：");**

**highness = inputInt(2, 120, "长度");**

**printf("请输入长方形的宽度（大小限制在2-120））：");**

**width = inputInt(2, 120, "宽度");**

**printf("请输入长方形的厚度（大小限制在1-5）：");**

**thickness = inputInt(1, 5, "厚度");**

**} while (highness < thickness || width < thickness);**

**int i, k;**

**for (i = 1; i <= width; i++)**

**{**

**k = 1;**

**if (i <= thickness)**

**{**

**while (k <= highness)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else if (i + thickness <= width)**

**{**

**while (k <= thickness)**

**{**

**printf("#");**

**k++;**

**}**

**while (k + thickness <= highness)**

**{**

**printf(" ");**

**k++;**

**}**

**while (k <= highness)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**else**

**{**

**while (k <= highness)**

**{**

**printf("#");**

**k++;**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_pyramid()**

**{**

**int highness, size;**

**printf("请输入金字塔的高度（大小限制在1-30）：");**

**size = inputInt(1, 30, "高度");**

**printf("请输入金字塔的层数（大小限制在1-100）：");**

**highness = inputInt(1, 120, "层数");**

**for (int i = 1; i <= size \* highness; i++)**

**{**

**for (int j = 1; j <= size \* highness + i; j++)**

**{**

**int outnum = (i - 1) / size + 1, innum = i % size ? i % size : size, kon = (highness - outnum) \* size + size - innum;**

**if ((j - kon) % (2 \* size) == 1 && j < highness \* size + i && j - kon>0)printf("/");**

**else if ((j + kon) % (2 \* size) == 0 && j > highness \* size - i)printf("\\");**

**else if (i % size == 0 && j > highness \* size - i && j < highness \* size + i)printf("\_");**

**else printf(" ");**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_cordiform()**

**{**

**int size;**

**printf("请输入心形的大小（大小限制在4-100）：");**

**size = inputInt(4, 100, "心形的大小");**

**int i, j, col = (size + size % 2 + (size - 3) \* 4)\*2 + 3, row = (size - 2) \* 2 + (col + 1) / 4;**

**for (int i = 1;i<=row; i++)**

**{**

**int innum,kon,m\_kon,cnum;**

**if (i <= size - 2)innum = i, kon = (size - 2) \* 2 - 1 - (innum - 1) \* 2, m\_kon = (size-2) \* 4 - 3 - (innum - 1) \* 4, cnum = size % 2 ? size + 1 + (innum - 1) \* 4:size + (innum - 1) \* 4;**

**else if (i <= 2 \* size-4)innum = i - size-2, kon = 0, m\_kon = 0,cnum=col;**

**else innum = i - (2 \* size-4), kon = (innum \* 2) - 1, m\_kon = 0, cnum = col - (innum - 1) \* 4 - 2;**

**if (i == 1 && size % 2 == 1)kon++, cnum = size;**

**for (int j = 1; j <= col; j++)**

**{**

**if (j <= kon)printf(" ");**

**else if (j <= kon + cnum)printf("\*");**

**else if (m\_kon)**

**{**

**if (j > kon + cnum + m\_kon && j <= kon + cnum \* 2 + m\_kon)printf("\*");**

**else printf(" ");**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_Chrustmas\_trees()**

**{**

**int size = 4, highness = 3;**

**//do**

**{**

**if (highness > size)printf("层数必须小于大小");**

**printf("请输入圣诞树的大小（大小限制在4-100）：");**

**size = inputInt(4, 100, "圣诞树的大小");**

**printf("请输入圣诞树的层数（大小限制在3-10）：");**

**highness = inputInt(3, 10, "圣诞树的层数");**

**} //while (highness > size);**

**int row= size \* highness + size, col = (size - 1 + highness - 1) \* 4 + 1;**

**for (int i = 1; i <= row; i++)**

**{**

**for (int j = 1; j <= col; j++)**

**{**

**int outnum = (i - 1) / size + 1, innum = i % size ? i % size : size, cnum , kon;//outnum 第几层 ,innum 这层第几行, cnum = \*的个数 ，Kon 左边空格**

**if (outnum <= highness)cnum = (innum - 1 + outnum - 1) \* 4 + 1, kon = (col - cnum) / 2;//树枝，每层size行，每层第一行从1加4，**

**else cnum = ((highness / 2) + 1) \* 2 + 1, kon = (col - cnum) / 2;**

**if (j <= kon)printf(" ");**

**else if (j <= kon + cnum)printf("\*");**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_Stare()**

**{**

**int size;**

**printf("请输入五角星的大小：（大小限制在4-100）");**

**size = inputInt(4, 100, "大小");**

**for (int i = 1; i <= size + (size\*4+2)/3 +1; i++)**

**{**

**int l=1, r=1, kon=0,cnum=0;**

**if (i <= size)kon = 3 \* size - i + 1, cnum = 2 \* i - 1;**

**else**

**{**

**l = (i - size - 1) \* 4 + 1;**

**r = 6 \* size - (i - size - 1) \* 4 + 1;**

**if (l >= 3 \* size - i + 1)cnum = 0;**

**else cnum = 1;**

**}**

**for (int j = 1; j <= size \* 6 + 1; j++)**

**{**

**if (i <= size)**

**{**

**if (j <= kon)printf(" ");**

**else if (j <= kon + cnum)printf("\*");**

**}**

**else**

**{**

**if (cnum)**

**{**

**if (j >= l && j <= r)printf("\*");**

**else printf(" ");**

**}**

**else**

**{**

**if (i == size + (size \* 4 + 2) / 3 + 1 && (j == 3 \* size - i + 2 || j== 3 \* size + i))printf("\*");**

**else if ((j > 3 \* size - i + 1 && j <= r) || (j <= 3 \* size + i && j >= l))printf("\*");**

**else printf(" ");**

**}**

**}**

**}**

**printf("\n");**

**}**

**}**

**void Draw\_playground()**

**{**

**initgraph(640, 480);**

**setbkcolor(0xFFAF31);**

**cleardevice();**

**setfillcolor(0xE63C0E);//跑道，深蓝**

**fillcircle(200, 200, 120);**

**fillcircle(450, 200, 120);**

**fillroundrect(200, 80, 450, 320, 0, 0);**

**for (int i = 120 - 8; i > 80; i -= 8)//跑道线**

**{**

**circle(200, 200, i);**

**circle(450, 200, i);**

**solidrectangle(200, 193 - i, 450, 207 + i);**

**line(200, 200 - i, 450, 200 - i);**

**line(200, 200 + i, 450, 200 + i);**

**}**

**setfillcolor(0xFFAF31);//操场草坪，浅蓝**

**fillcircle(200, 200, 80);**

**fillcircle(450, 200, 80);**

**setfillcolor(0x32E03E);//浅绿**

**fillroundrect(200, 120, 450, 280, 0, 0);**

**setfillcolor(0x1C7D23);//深绿**

**for (int i = 200; i < 450; i += 25)**

**fillroundrect(i, 120, i + 12, 280, 0, 0);**

**//足球场**

**circle(231, 200, 21);**

**setfillcolor(0x32E03E);**

**solidrectangle(238, 125, 244, 275);**

**solidrectangle(213, 125, 224, 275);**

**setfillcolor(0x1C7D23);**

**solidrectangle(201, 125, 211, 275);**

**solidrectangle(226, 125, 236, 275);**

**circle(418, 200, 21);**

**setfillcolor(0x1C7D23);**

**solidrectangle(405, 125, 411, 275);**

**solidrectangle(426, 125, 436, 275);**

**setfillcolor(0x32E03E);**

**solidrectangle(413, 125, 424, 275);**

**solidrectangle(438, 125, 449, 275);**

**rectangle(206, 126, 443, 274);**

**rectangle(206, 154, 244, 246);**

**rectangle(206, 179, 219, 221);**

**rectangle(405, 154, 443, 246);**

**rectangle(430, 179, 443, 221);**

**line(325, 126, 325, 274);**

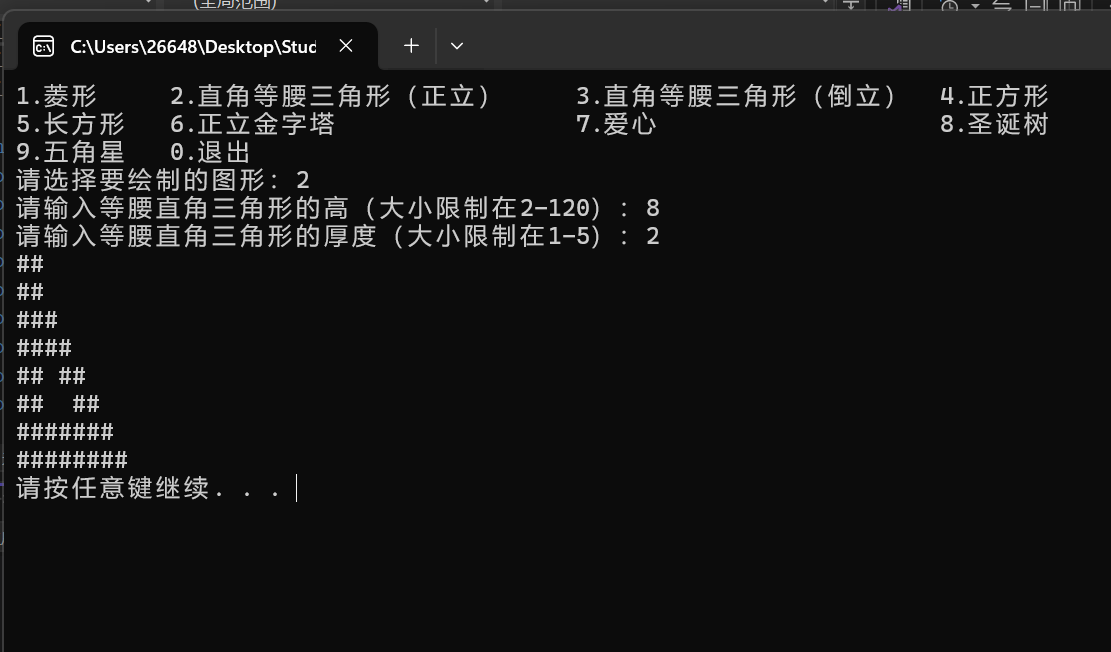
**circle(325, 200, 21);**

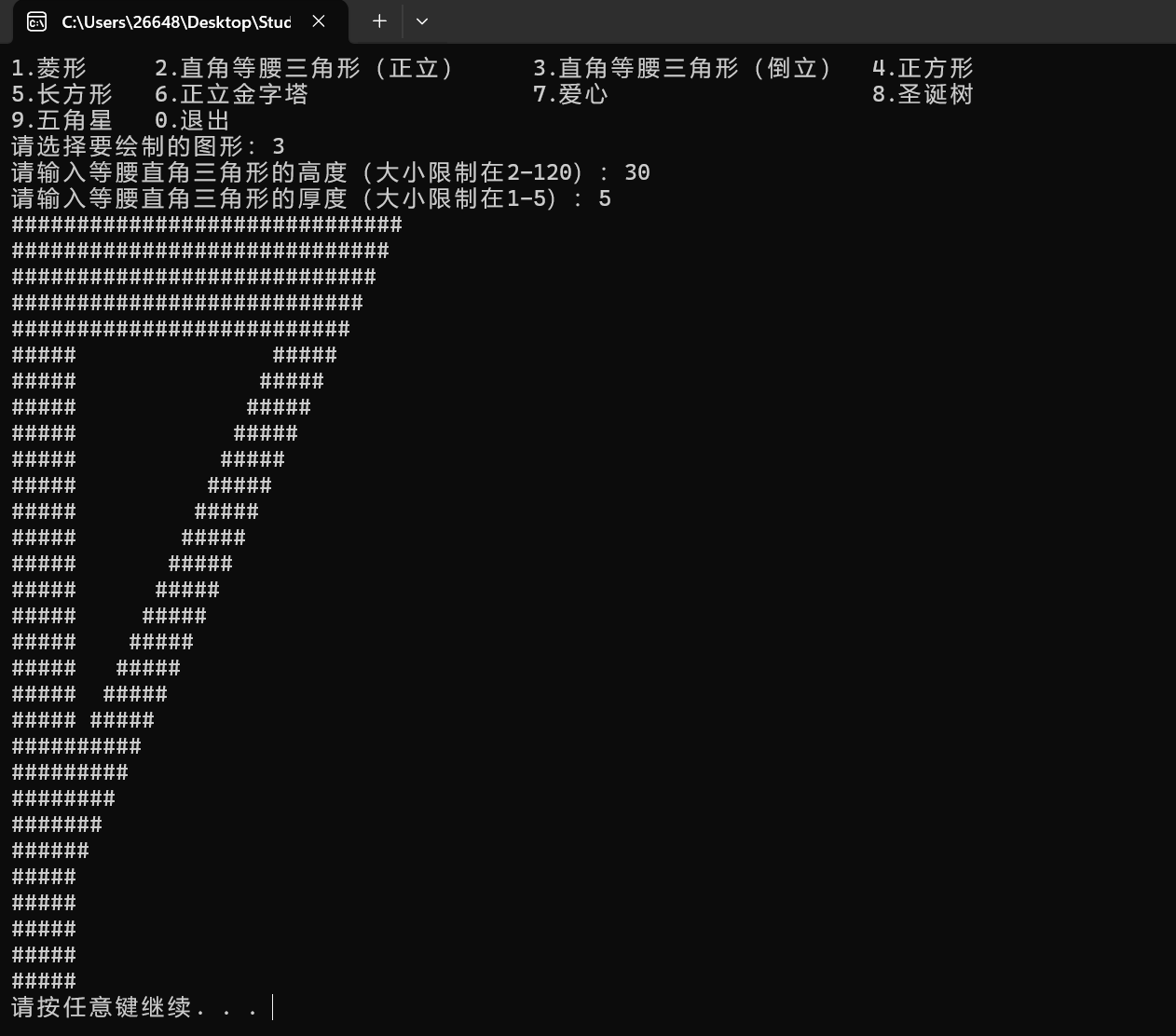
**\_getch();**

**closegraph();**

**}**

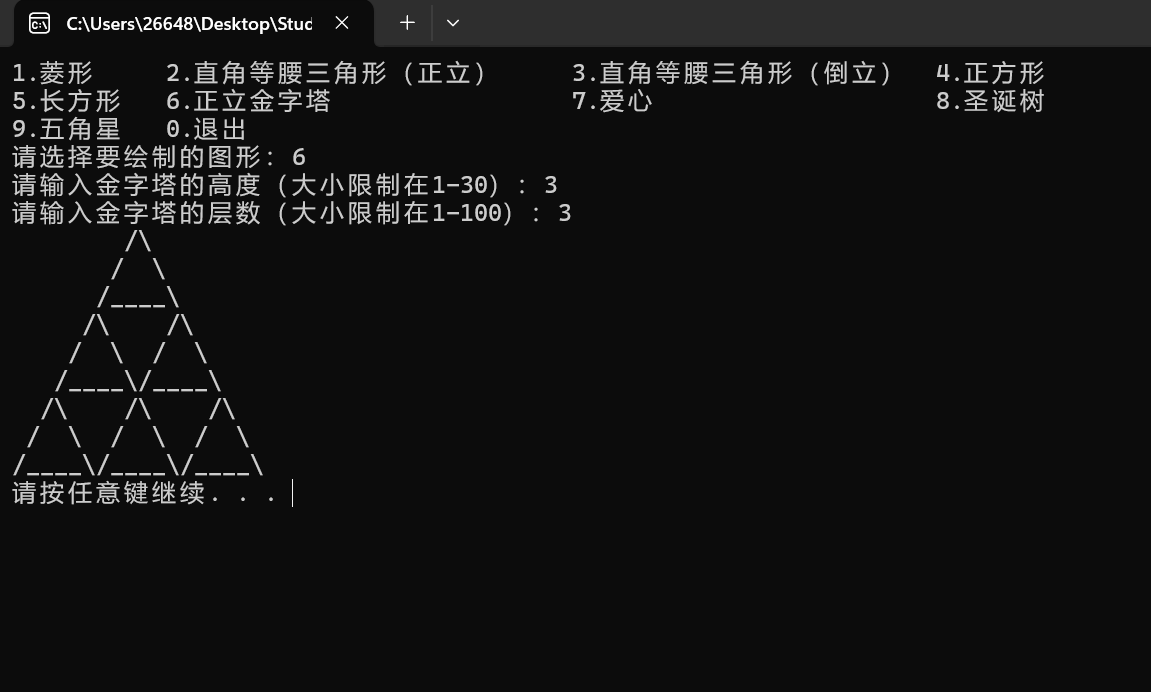
**2.运行结果截图：**

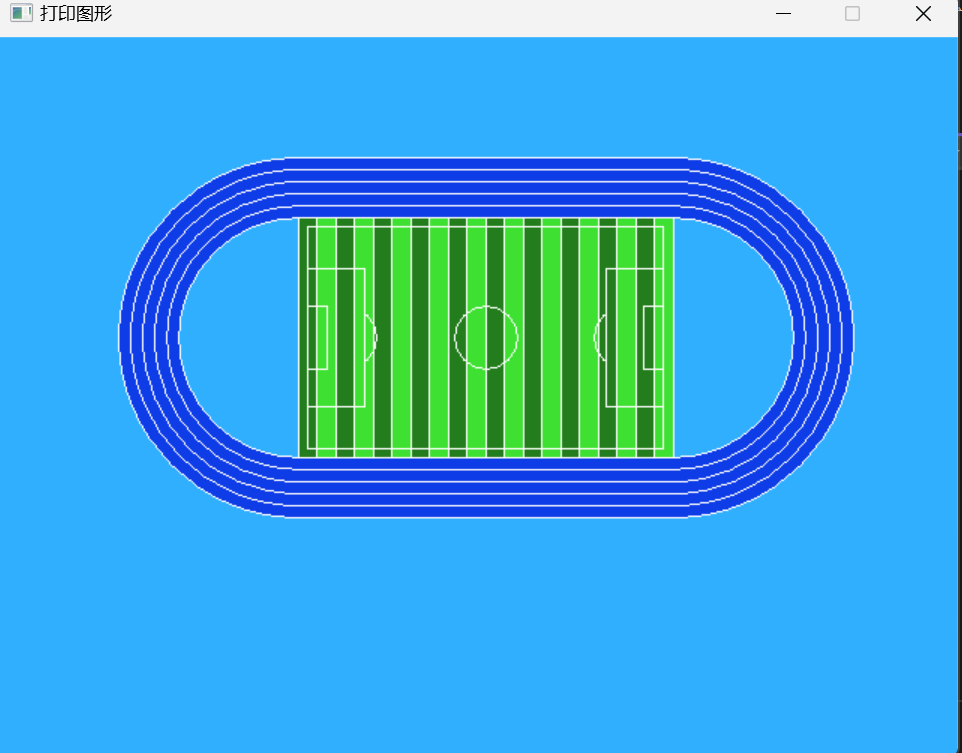
****

****

****

****

**------------**



**3.主要流程图**

**4.总结：**