

补全代码后，在菜单中选择“生成”->“生成解决方案”，Visual studio 开始编译，如果编译成功（界面最下方的输出框显示成功），则说明没有语法错误，**如果没有成功，输出框会提示语法错误位置**，请大家从第一个错误提示显示的代码行数附近检查是否有语法错误。编译成功后，在菜单中选择“调试”->“开始执行（不调试）”运行程序，这时候可能会出现与预期不一致的运行结果，说明代码中存在**逻辑错误**，同学们可以通过在菜单中选择“调试”中的“调试”、“逐语句”、“逐过程”等子菜单调试程序，要学会利用“断点”、“自动窗口”、“监视”、“局部变量”、“内存”等工具调试代码中存在的逻辑错误。

实验二 zumaV1（链表的应用）

实验二目标：利用链表和 Easy-X 图形库实现球链的初始化和显示。

同样的，实验二我们给出一个新的程序框架，包含 main.cpp、LinkedList.cpp、LinkedList.h、ball.h（为了和 Easy-X 图形库兼容，*.c 文件全部改名为*.cpp 文件）。其中 LinkedList.cpp、LinkedList.h、ball.h 文件已经完成，main.cpp 只完成大部分，你的任务是补全 main.cpp 的代码（initBallList() 和 drawBallList() 两个函数），让整个代码可以编译并执行，如图 2-1 所示。

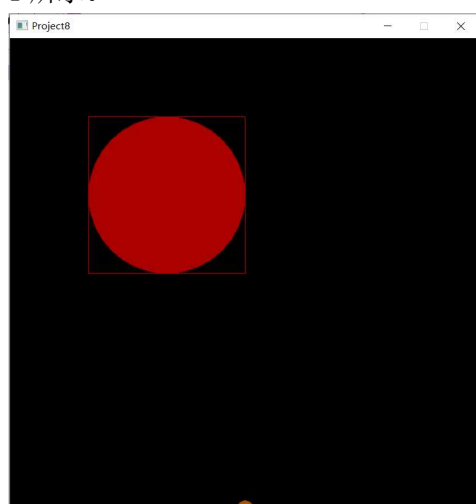


图 2-1（a）补全代码前编译显示结果

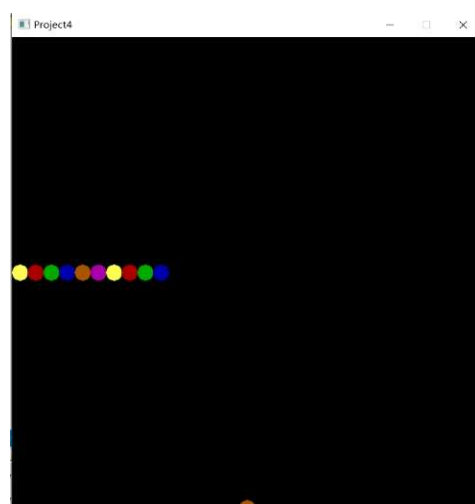


图 2-1（b）补全代码后编译显示结果

```
46 //绘制球链表，请补全该函数
47 void drawBallList(Node* head)
48 {
49
50 }
51
52 //初始化球链表，即给球链表中添加一些球，请补全该函数
53 void initBallList(Node* head)
54 {
55
56 }
57
58 int main()
59 {
60 // 初始化绘图窗口
61 initgraph(WINDOWWIDTH, WINDOWHEIGHT);
62 setbkcolor(BLACK);
63 cleardevice();
64
65 //创建一个空的球链表
66 Node* head = NULL;
67 head = CreateEmptyList();
68
69 //初始化球链表
70 initBallList(head);
71
72 //在窗口中绘制球链表
73 drawBallList(head);
74 //绘图示例函数，补全drawBallList()函数后请删除
75 drawExample();
76
77 }
```

图 2-1（c） 在 main.cpp 中要补全的两个函数，initBallList() 和 drawBallList()。

评价标准：实现图 2-1（b）类似的效果，由助教检查后评分。

1. LinkList.cpp、LinkList.h 文件说明

这两个文件是大家在实验一完成的单链表模块，在实验二中直接使用实验一完成的模块。由于 C 语言的局限性，我们在使用单链表模块时，要对 LinkList.h 文件进行一个小的修改，如图 2-2 中第 9 行和第 10 代码所示，我们将实验一的数据元素类型由 int 型改为构造类型 ball，结构体 ball 在 ball.h 中定义。

```
9      //typedef int DataType;
10     typedef ball DataType;
11
12     typedef struct Node
13     {
14         DataType data;
15         struct Node *next;
16     } Node;
```

图 2-2 LinkList.h 文件的修改

2. ball.h 文件说明

ball.h 文件的代码如图 2-3 所示，该文件中定义了 ball 结构体，该结构体中包含球的位置（int x, int y）和球的颜色（int c），另外该文件还定义了球的半径，如代码第 9 行所示。

```
1
2     #pragma once
3
4     #ifndef _BALL_H_
5     #define _BALL_H_
6
7     #include <graphics.h>
8
9     #define BALLRADIUS 10
10
11     struct ball
12     {
13         int x;
14         int y;
15         int c;
16     };
17
18
19     #endif
```

图 2-3 ball.h 文件代码

3. main.cpp 文件说明

在 main.cpp 文件的 main() 函数中，首先调用 Easy-X 图形库提供的函数初始化绘图窗口（第 52-55 行代码），然后再利用链表模块提供的函数创建一个空链表，接着初始化球链表，即往空的球链表中添加一些球，紧接着将这个球链表在窗口中绘制出来，**这两个函数（initBallList() 和 drawBallList()）需要同学们将其补全**。其中 drawExample 是一个示例函数，在这个示例函数中调用 Easy-X 图形库提供的函数在窗口中绘制了一个矩形和一个圆形，供大家参考，在补全 drawBallList() 函数后，可将 drawExample() 函数注释掉或者删除。

70-73 行代码是绘制碰撞球，76 行代码是等待用户按任意键，如果用户按下任意键，_getch() 函数将返回，紧接着执行剩下的代码，程序结束。需要注意的是，_getch() 函数是一个阻塞式的函数，当用户没有按下任意键时，这个函数会阻塞，即程序停留在该函数中不返回。

```
50  int main()
51  {
52      // 初始化绘图窗口
53      initgraph(WINDOWWIDTH, WINDOWHEIGHT);
54      setbkcolor(BLACK);
55      cleardevice();
56
57      //创建一个空的球链表
58      Node* head = NULL;
59      head = CreateEmptyList();
60
61      //初始化球链表
62      initBallList(head);
63
64      //在窗口中绘制球链表
65      drawBallList(head);
66      //绘图示例函数，补全drawBallList () 函数后请删除
67      drawExample();
68
69
70      //画碰撞球
71      ball cball;
72      cball.c = rand() % 6;
73      drawColBall(&cball, WINDOWWIDTH / 2, WINDOWHEIGHT);
74
75      //等待任意键退出
76      _getch();
77      //销毁球链表和绘图窗口
78      DestroyList(head);
79      closegraph();
80
81      return 0;
82  }
```

图 2-4 main.cpp 文件代码