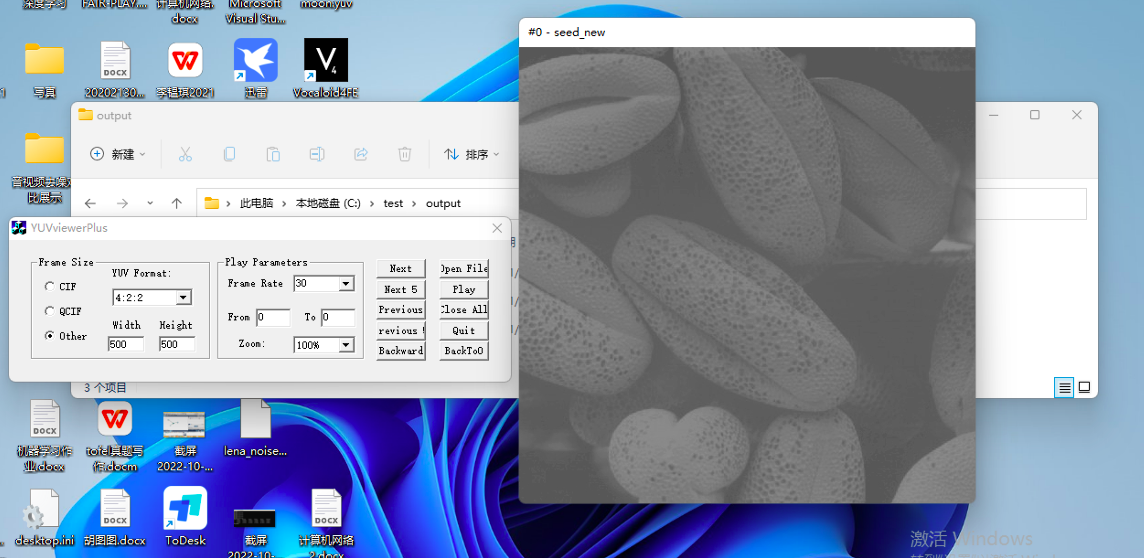
# 实验一 YUV文件的基本操作

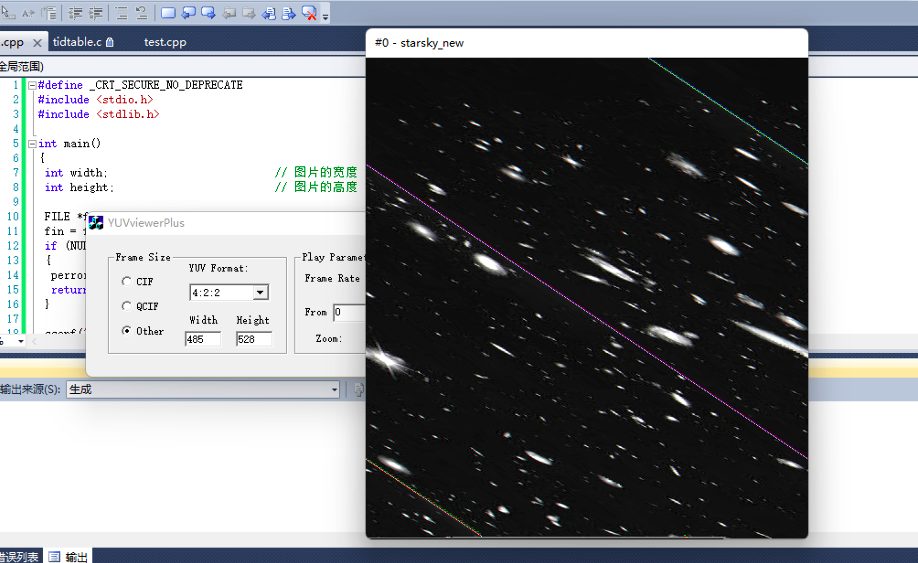
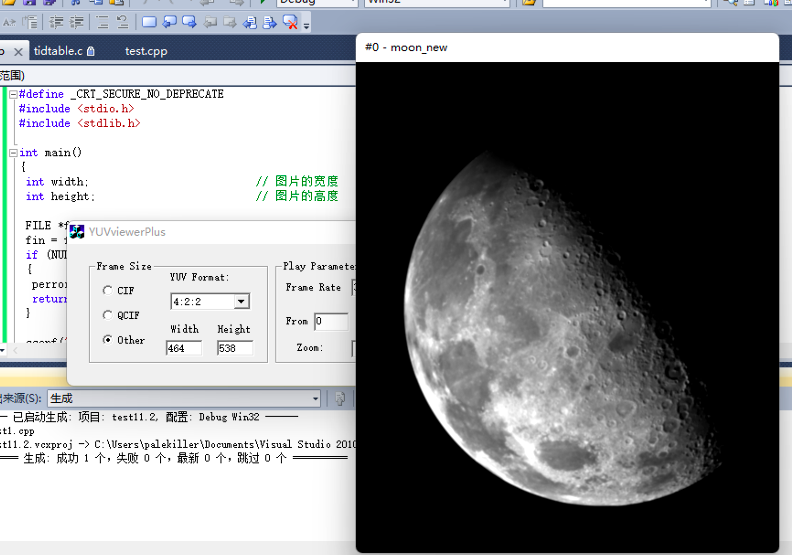
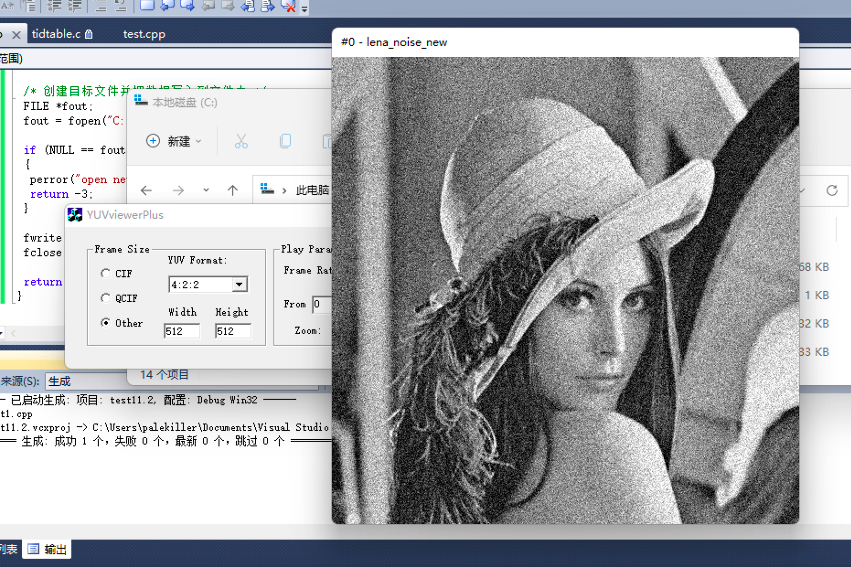
### 实验目的

1. 熟悉数字图像、像素的概念。
2. 熟悉YUV文件的存储格式和采样格式。
3. 熟练掌握C语言中对文件的基本操作以及循环操作。

### 实验任务

1. 编程实现将所给YUV图像的采样格式由4:4:4转化为4:2:2，然后通过播放器进行播放。





**代码：（以seed.yuv为例，其他图片同理）**

**#define \_CRT\_SECURE\_NO\_DEPRECATE**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int width; // 图片的宽度**

**int height; // 图片的高度**

**FILE \*fin;**

**fin = fopen("C:\\seed.yuv", "rb");**

**if (NULL == fin)**

**{**

**perror("open file is failed\n");**

**return -2;**

**}**

**scanf("%d,%d", &width, &height); // 图片宽高**

**int len = width \* height \*3; // 文件大小**

**unsigned char \*data = (unsigned char \*)malloc(len); // 转换后数据存放的位置**

**if (NULL == data)**

**{**

**fprintf(stderr, "malloc data failed\n");**

**fclose(fin);**

**return -3;**

**}**

**/\* 配置 y u v 数据位置 \*/**

**unsigned char \*y = data; // y数据存放的位置**

**unsigned char \*u = data + width \* height; // u数据存放的位置**

**unsigned char \*v = u + width \* height / 2; // v数据存放的位置**

**unsigned char \*buff = (unsigned char \*)malloc(width \* 3);//设置缓存区**

**if (NULL == buff)**

**{**

**fprintf(stderr, "malloc buff failed\n");**

**fclose(fin);**

**return -4;**

**}**

**/\*put your code here\*/**

**/\*提示：使用fread(buff, 1, ?, fin)\*/**

**int i, j;**

**for (i = 0; i < height; i++)**

**{**

**fread(buff, 1, width, fin); //读取fin，每次读取一个字节，读取width个项，最终放入buff中**

**for (j = 0; j < width; j += 2)**

**{**

**\*y++ = buff[j];**

**\*y++ = buff[j + 1];**

**}**

**}**

**for (i = 0; i < height; i++)**

**{**

**fread(buff, 1, width \* 3, fin);**

**for (j = 0; j < width \* 3; j += 6)**

**{**

**\*u++ = buff[j + 1];**

**\*v++ = buff[j + 5];**

**}**

**}**

**/\*end\*/**

**/\* 创建目标文件并把数据写入到文件中 \*/**

**FILE \*fout;**

**fout = fopen("C:\\test\\output\\seed\_new.yuv", "wb"); //其他图片同理，此处不展示代码。**

**if (NULL == fout)**

**{**

**perror("open newfile is failed\n");**

**return -3;**

**}**

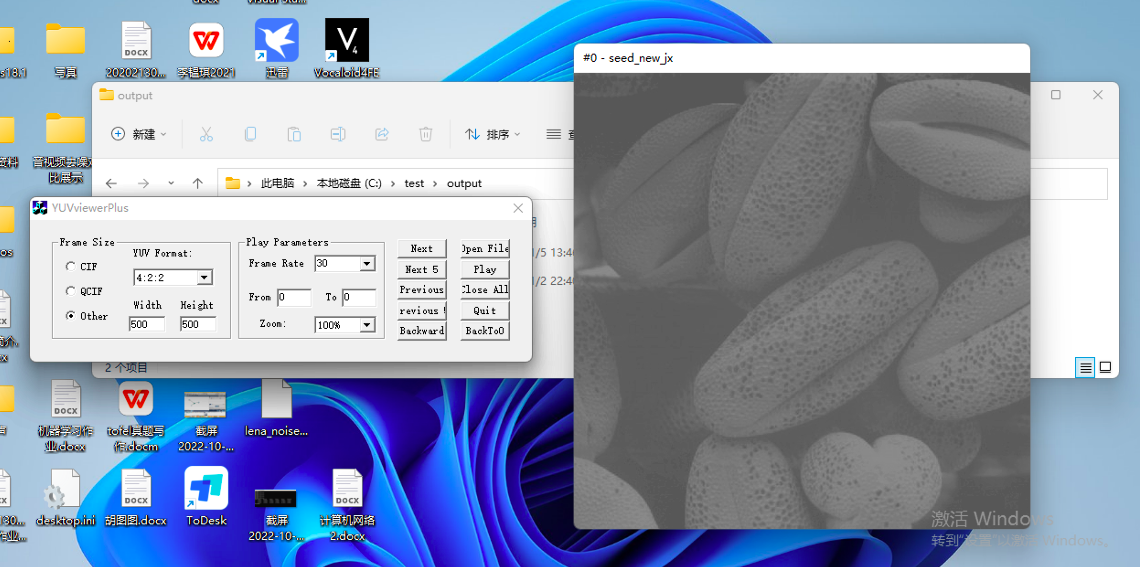
**fwrite(data, 1, len, fout);**

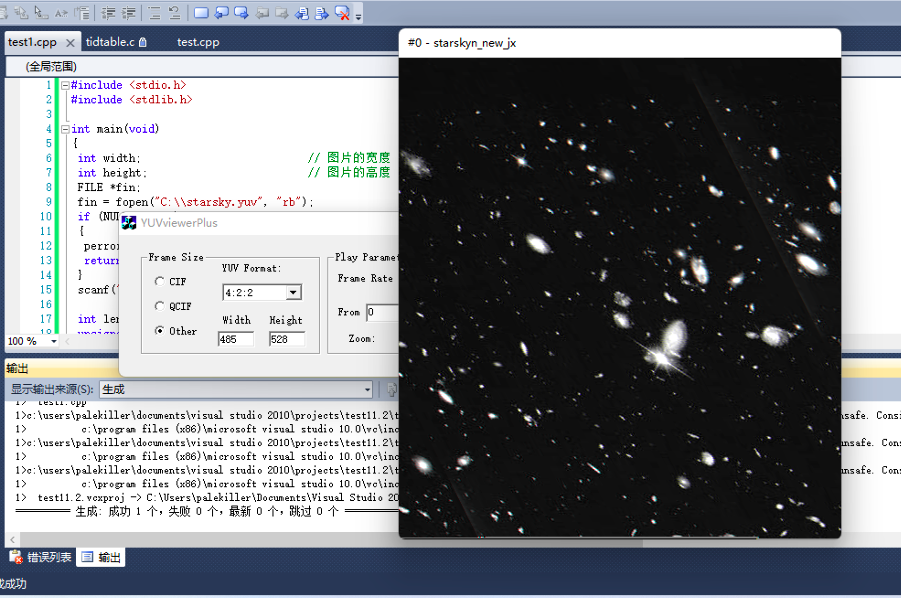
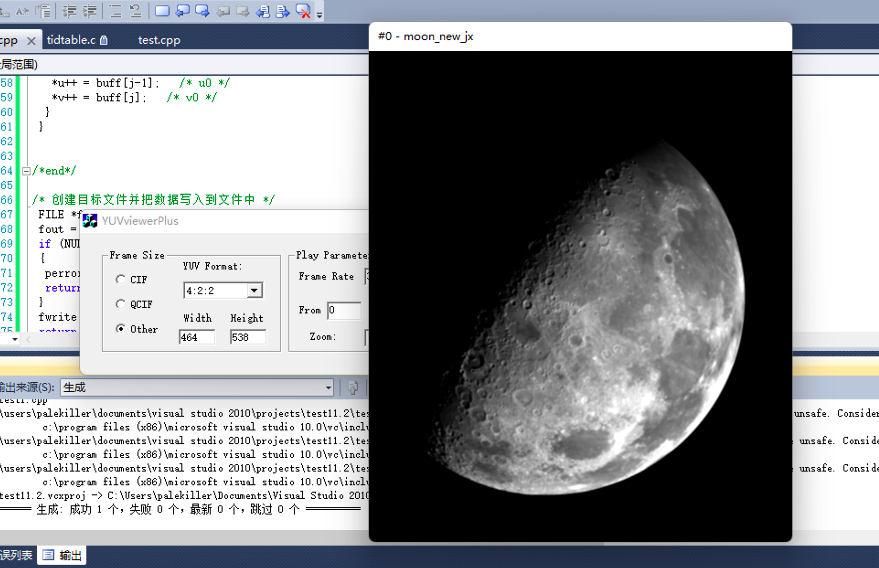
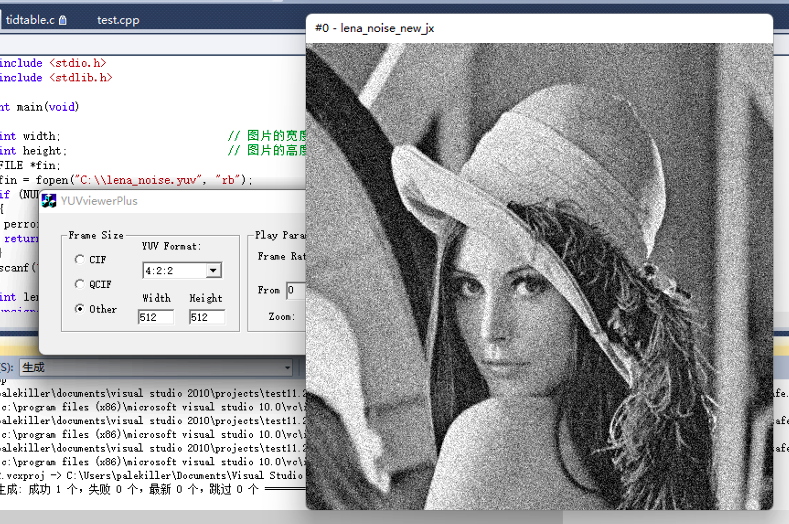
**fclose(fout);**

**return 0;**

**}**

1. 将图像做镜像（水平或垂直翻转），然后通过播放器进行播放。



****

**代码：（以seed.yuv为例，其他图片同理）**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(void)**

**{**

**int width; // 图片的宽度**

**int height; // 图片的高度**

**FILE \*fin;**

**fin = fopen("C:\\seed.yuv", "rb");**

**if (NULL == fin)**

**{**

**perror("open file is failed\n");**

**return -2;**

**}**

**scanf("%d,%d", &width, &height); // 图片宽高**

**int len=width\*height\*3 ; // 文件大小**

**unsigned char \*data = (unsigned char \*)malloc(len); // 转换后数据存放的位置**

**if (NULL == data)**

**{**

**fprintf(stderr, "malloc data failed\n");**

**fclose(fin);**

**return -3;**

**}**

**/\* 配置 y u v 数据位置 \*/**

**unsigned char \*y = data; // y数据存放的位置**

**unsigned char \*u = data+width\*height; // u数据存放的位置**

**unsigned char \*v = u+width\*height; // v数据存放的位置**

**unsigned char \*buff =(unsigned char \*)malloc(width\*3);//设置缓存区**

**if (NULL == buff)**

**{**

**fprintf(stderr, "malloc buff failed\n");**

**fclose(fin);**

**return -4;**

**}**

**/\*put your code here\*/**

**/\*提示：使用fread(buff, 1, ?, fin)\*/**

**int i,j;**

**//fread(buff, 1, height\*3, fin);**

**for (i = 0; i < height; i += 1)**

**{**

**fread(buff, 1, width, fin);**

**for (j = width-1; j >=0; j -= 1)**

**{**

**\*y++ = buff[j]; /\* y0 \*/**

**}**

**}**

**for (i = 0; i < height; i += 1)**

**{**

**fread(buff, 1, width\*3, fin);**

**for (j = width\*3-1; j>=0 ; j -= 3)**

**{**

**\*u++ = buff[j-1]; /\* u0 \*/**

**\*v++ = buff[j]; /\* v0 \*/**

**}**

**}**

**/\*end\*/**

**/\* 创建目标文件并把数据写入到文件中 \*/**

**FILE \*fout;**

**fout = fopen("C:\\test\\output\\seed\_new\_jx.yuv", "wb");**

**if (NULL == fout)**

**{**

**perror("open newfile is failed\n");**

**return -3;**

**}**

**fwrite(data ,1, len, fout);**

**return 0;**

**}**

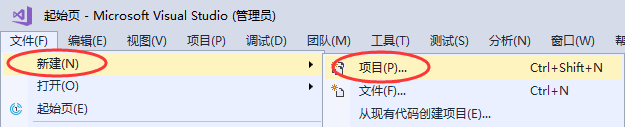
提示：可借助所给YUV播放器实现对视频图像的观看，注意界面参数设置。

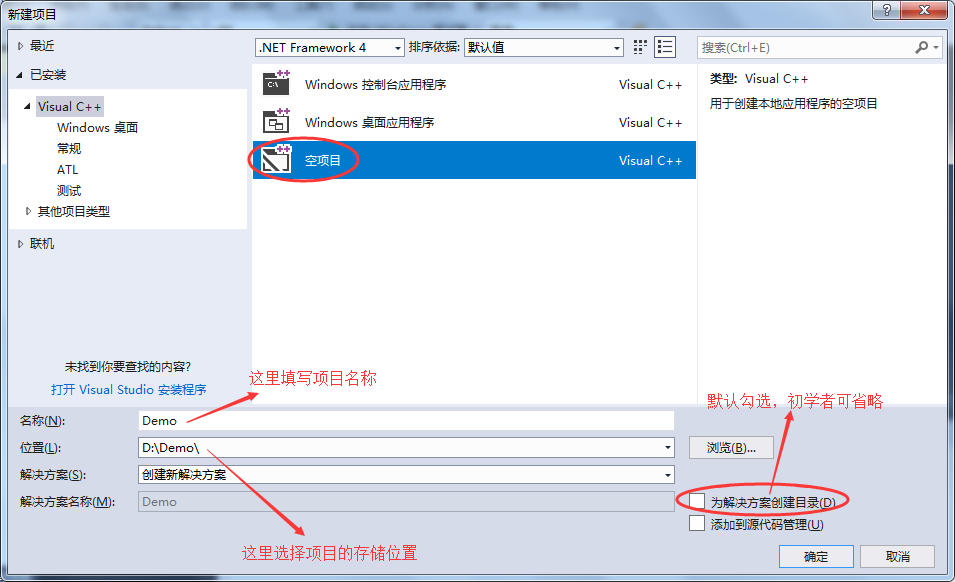
### 实验环境和搭建

本次实验建议在Visual Studio中进行程序的编写和调试。下面以Visual Studio 2017为例简单介绍项目的创建和源文件的创建。

1. **创建新项目**

打开 VS 2017，在菜单栏中依次选择 “文件 --> 新建 --> 项目”：



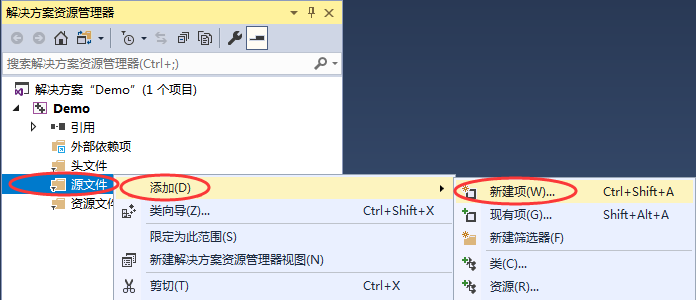


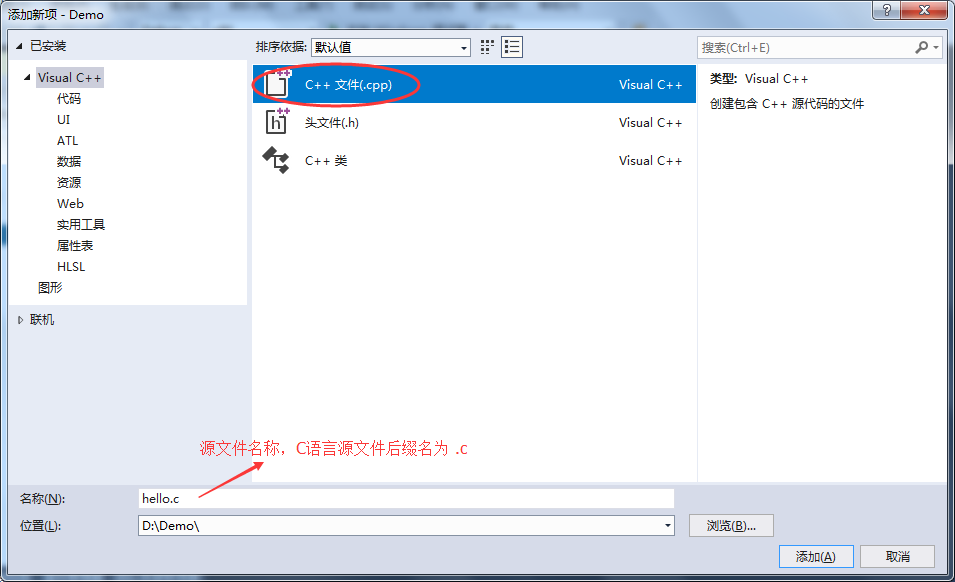
另外，项目名称和存储路径中最好不要包含中文。

点击 “确定” 按钮后，会直接进入项目可操作界面，我们将在这个界面完成所有的编程工作。

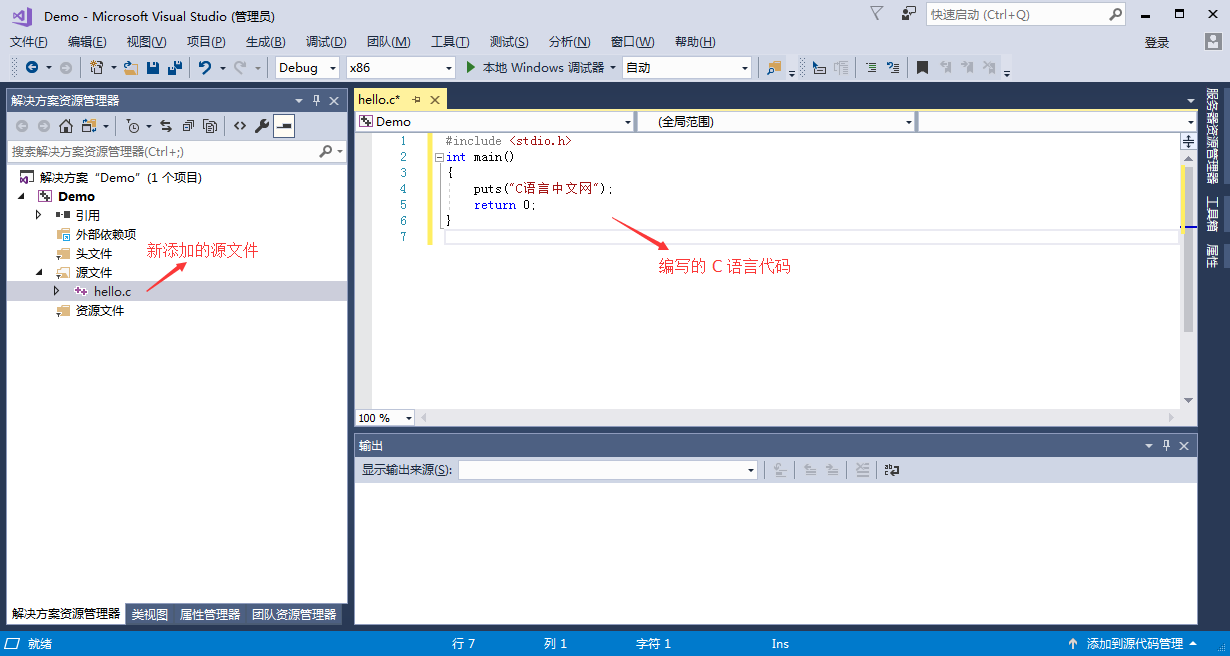
1. **添加源文件**

在 “源文件” 处右击鼠标，在弹出菜单中选择 “添加 --> 新建项” ，如下图所示：





点击 “添加” 按钮，就添加上了一个新的源文件。即可在编辑窗口中进行代码输入。



### 实验材料

四个YUV格式图像文件，采样格式均为4:4:4，其图像参数如下：

|  |  |  |
| --- | --- | --- |
| 文件名 | 宽Width | 高Height |
| seed.yuv | 500 | 500 |
| lena\_noise.yuv | 512 | 512 |
| moon.yuv | 464 | 538 |
| starsky.yuv | 485 | 528 |

可以用YUVviewerPlus看上述图像以及处理后的图像，注意正确选择宽、高和采样格式。（使用方法见附录）

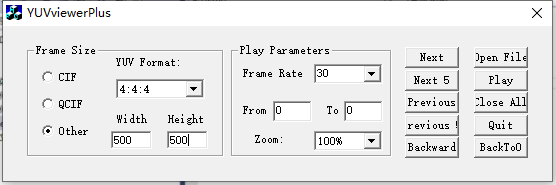
### 实验步骤

1. 创建新的项目和源文件。
2. 打开原图像文件。（涉及fopen函数的使用）
3. 定义图像宽高，存放数据的位置等变量。
4. 利用循环，对文件进行读操作，并对读取内容进行相应处理。（涉及fread函数的使用）
5. 创建新的文件并将处理好的数据写入。（涉及fwrite函数的使用）
6. 使用播放器查看新的文件是否达到预期效果。

### 附录：

YUVviewerPlus的使用方法

直接双击运行所给YUVviewerPlus.exe



在此面板上，如上进行对应的选择和参数设置，最后点击open file选择需要浏览的图像文件。