

程序文件打开后注释乱码的解决方案

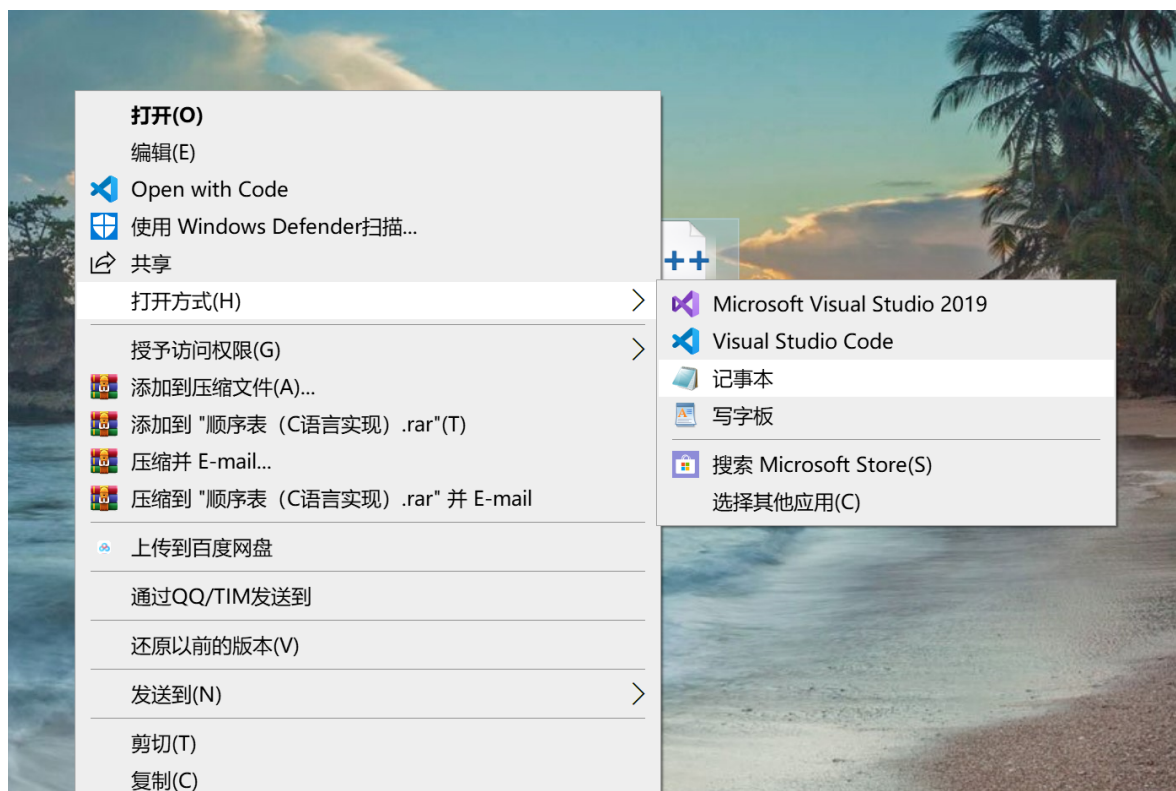
由于程序代码使用Visual Studio Code编写，默认为UTF-8格式，而有些编译器对中文的默认编码为GBK格式，因此打开程序文件后可能会出现中文注释乱码的情况。如下图所示：

```
顺序表 (C语言实现) .cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // 渝氢筵楞哄簪琛 | 栢鐳勳強鏄稿另硬格磬鍍芥啞澹版霖
5
6  #define OK 1
7  #define OVERFLOW -1
8  #define LIST_INIT_SIZE 2 // 绡撿€u " 濠樹促紬洪梔錄漬 錄噤厖淪归嘶
9  #define LISTINCREMENT 10 // 绡撿€u " 濠樹促紬洪梔錄噤厖遭為嘶
10 #define TRUE 1
11 #define FALSE 0
12 #define NOTEXIST 0
13 #define ERROR 0
14
15 typedef int Status;
16 typedef int ElemType;
17
18 struct Slist { // 楞哄簪琛 | 被鏗?
19     ElemType *elem; // 濠樹促紬洪梔錯哄滑
20     int length; // 褰嶳墀闈罔垮害
21     int listsize; // 褰嶳墀錄噤厖鐳勳項塗厶 閱?
22 };
```

如果出现上述情况，按以下步骤解决：

1. 将文件用记事本打开





2. 打开后可以看到程序文件现在为UTF-8格式（如下图右下角所示）

```
顺序表（C语言实现）.cpp - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#include <stdio.h>
#include <stdlib.h>

// 定义顺序表结构及相关算法函数声明

#define OK 1
#define OVERFLOW -1
#define LIST_INIT_SIZE 2 // 线性表存储空间初始分配容量
#define LISTINCREMENT 10 // 线性表存储空间分配增量
#define TRUE 1
#define FALSE 0
#define NOTEXIST 0
#define ERROR 0

typedef int Status;
typedef int ElemType;

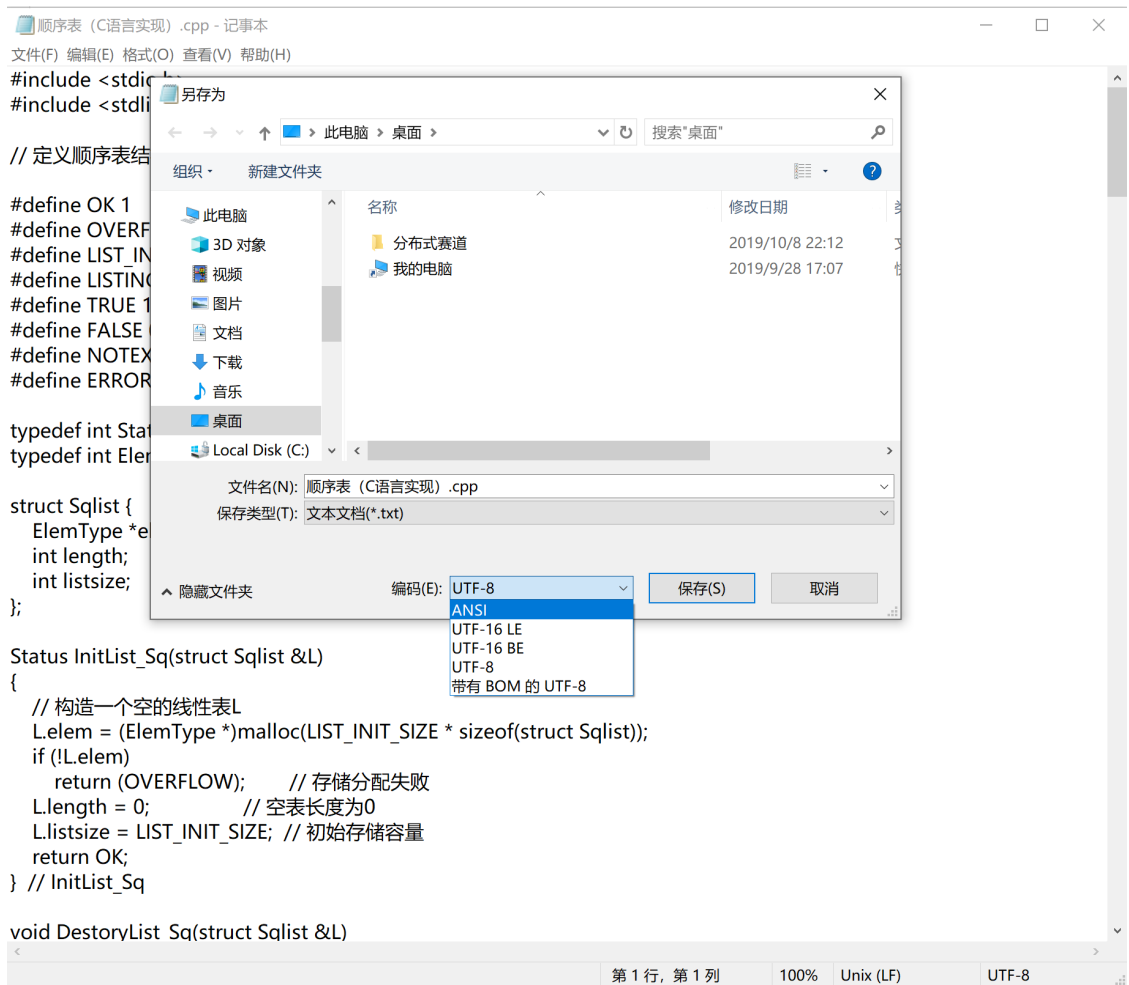
struct Sqli { //顺序表类型
    ElemType *elem; // 存储空间基址
    int length; // 当前长度
    int listsize; // 当前分配的存储容量
};

Status InitList_Sq(struct Sqli &L)
{
    // 构造一个空的线性表L
    L.elem = (ElemType *)malloc(LIST_INIT_SIZE * sizeof(struct Sqli));
    if (!L.elem)
        return (OVERFLOW); // 存储分配失败
    L.length = 0; // 空表长度为0
    L.listsize = LIST_INIT_SIZE; // 初始存储容量
    return OK;
} // InitList_Sq

void DestoryList Sq(struct Sqli &L)
```

第 1 行, 第 1 列 100% Unix (LF) UTF-8

3. 点击文件，然后选择另存为，在另存为界面将编码修改为ANSI，再点击保存。（可以不重命名，直接覆盖掉原来的文件）



4. 再次打开程序文件时就不会有中文注释乱码的情况了。

```
顺序表 (C语言实现).cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // 定义顺序表结构及相关算法函数声明
5
6  #define OK 1
7  #define OVERFLOW -1
8  #define LIST_INIT_SIZE 2 // 线性表存储空间初始分配容量
9  #define LISTINCREMENT 10 // 线性表存储空间分配增量
10 #define TRUE 1
11 #define FALSE 0
12 #define NOTEXIST 0
13 #define ERROR 0
14
15 typedef int Status;
16 typedef int ElemType;
17
18 struct Sqlist { // 顺序表类型
19     ElemType *elem; // 存储空间基址
20     int length; // 当前长度
21     int listsize; // 当前分配的存储容量
22 };
23
```