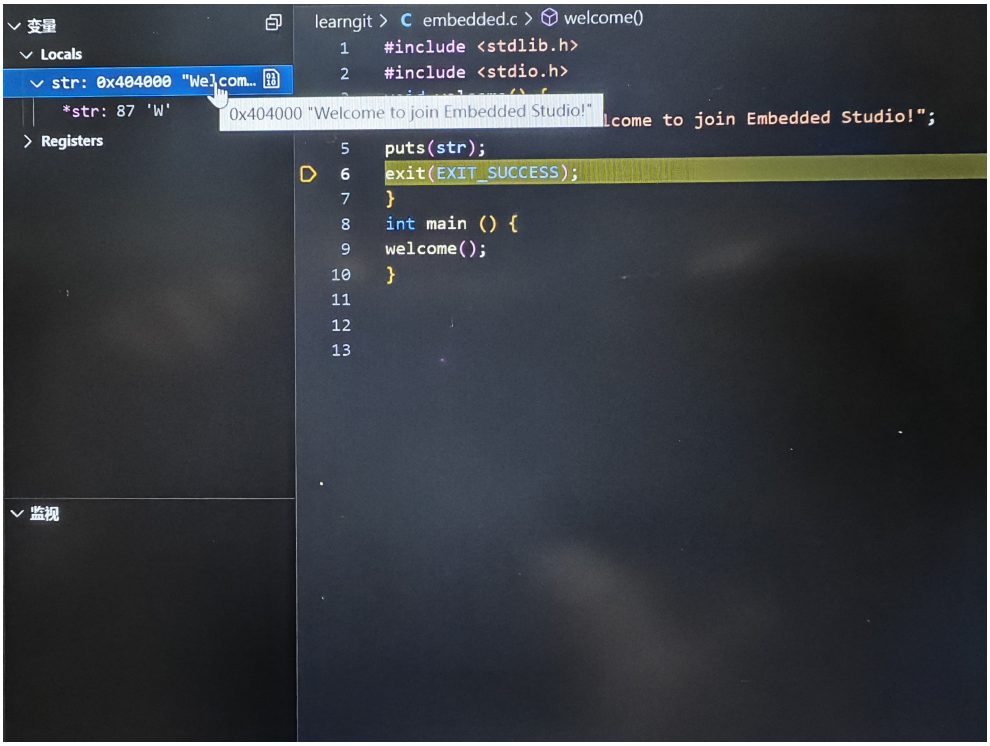
嵌入式一轮C语言基础

工欲善其事，必先利其器

Q1:

```
PS D:\learngit> gcc embedded.c -o embedded.exe
PS D:\learngit> .\embedded
Welcome to join Embedded Studio!
PS D:\learngit>
```

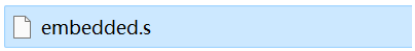
Q2:



Q3:1.gcc -E source_file.c -E，只执行到预编译。直接输出预编译结果。

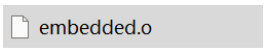
2.gcc -S source*file.c -S，只执行源代码到汇编代码的转换，默认生成名为source*file.s的汇编代码。

```
PS D:\learngit> gcc -S embedded.c
PS D:\learngit>
```



3.-c，执行完编译和汇编，默认生成名为source_file.o的目标文件。

```
PS D:\learngit> gcc -c embedded.c
PS D:\learngit>
```



4.gcc (-E/S/c）source_file.c -o

output_filename -o，指定输出文件名，可任意配合以上三种标签使用。-o可省略，无标签情况默认生成名为a.out的可执行文件。

```
PS D:\learngit> gcc embedded.c -o embedded.exe
PS D:\learngit>
```

```
PS D:\learngit> gcc embedded.c
PS D:\learngit> .\a
Welcome to join Embedded Studio!
PS D:\learngit>
```

配置开发环境过程

1.下载并安装VS Code

2.安装C/C++语言插件，安装highlight-words插件

3.下载MinGW-w64，编译器

4.配置环境变量，添加ming64的bin目录路径，并在cmd中输入where gcc查看gcc路径，输入gcc--version查看版本，以确认环境变量生效

5.了解了如何在vscode中打开文件夹，新建c文件夹，编译运行程序

6.为了实现调试，在vscode中配置了c*cpp*properties.json launch.json tasks.json三个文件

7.为了在vscode中写markdown，安装了Markdown All in One插件，安装Markdown Preview Enhanced插件来实时预览，安装Paste Image来方便在插入图片

分支与循环

重拾小学数学

解题思路：根据题意，可分为能组成三角形和不能组成三角形两种大情况；其中能组成三角形又可分为钝角三角形，直角三角形，锐角三角形三种情况；其中钝角三角形和锐角三角形可能是等腰三角形（由于三边长均为正整数直角三角形不可能是等腰三角形），锐角三角形可能是等边三角形。可见有多种情况，所以我选择用if-else语句。首先要输入三角形三边长，所以定义三个变量a，b，c；又因为后续三角形判定过程中需要用到边长的平方，所以再定义m,n，q分别为a*a，b*b*，*c*c。接下来先判定三条边能否组成三角形，利用三角形两边之和大于第三边，两边之差小于第三边，用if-else写出如下代码：

```
if(a+b>c&&a-b<c)
```

```
else{
    printf("Not triangle\n");
```

然后在if后面嵌套if-else，用勾股定理来和余弦定理来判定三角形是直角锐角或钝角三角形：

```
if(a+b>c&&a-b<c){
    if(m+n==q||m+q==n||n+q==m){
        printf("Right triangle\n");
    }else if(m+n>q&&m+q>n&&n+q>m){
        printf("Acute triangle\n");
    }else {
        printf("Obtuse triangle\n");
    }
```

三角形是等腰三角形或等边三角形的同时也可能是其他三角形，比较特殊,而题目要求若三角形满足多个条件要分别输出，所以用两个if单独写出：

```
    if(a==b||a==c||b==c){
        printf("Isosceles triangle\n");
    }
    if(a==b&&b==c){
        printf("Equilateral triangle\n");
    }
}
```

最后形成完整的代码如下：

```
#include<stdio.h>
int main(){
    int a,b,c;
    printf("输入三角形三边长度,均是不大于10000的正整数:\n");
    scanf("%d %d %d",&a,&b,&c);
    int m=a*a;
    int n=b*b;
    int q=c*c;
    if(a+b>c&&a-b<c){
        if(m+n==q||m+q==n||n+q==m){
            printf("Right triangle\n");
        }else if(m+n>q&&m+q>n&&n+q>m){
            printf("Acute triangle\n");
        }else {
            printf("Obtuse triangle\n");
        }
        if(a==b||a==c||b==c){
            printf("Isosceles triangle\n");
        }
        if(a==b&&b==c){
            printf("Equilateral triangle\n");
        }
    }
    else{
        printf("Not triangle\n");
    }
    return 0;
}
```

运行结果：

1.

```
输入三角形三边长度,均是不大于10000的正整数:
1 2 3
Not triangle
```

2.

```
输入三角形三边长度,均是不大于10000的正整数:
3 4 5
Right triangle
```

3.

```
输入三角形三边长度,均是不大于10000的正整数:
12 14 15
Acute triangle
```

4.

```
输入三角形三边长度,均是不大于10000的正整数:
12 13 20
Obtuse triangle
```

5.

```
输入三角形三边长度,均是不大于10000的正整数:
6 10 6
Obtuse triangle
Isosceles triangle
```

6.

```
输入三角形三边长度,均是不大于10000的正整数:
4 4 4
Acute triangle
Isosceles triangle
Equilateral triangle
```

方阵？内卷！

解题思路：根据题意，输出方阵要用二维数组，而要达到“内卷”的效果，需要通过方阵中数的坐标控制输出，在循环中还要判定边界，最后遍历数组输出方阵。我首先定义i，j两个变量充当坐标并初始化为0，在定义n作为要输入的正整数，定义k作为方阵中的数，初始化为1。

```c
#include<stdio.h>
int a[50][50];
int main()
{

    int i=0,j=0;
    int n,k=1;
```

方阵中数的排列有四个方向，先从左到右，再从上到下，再从右到左，再从下到上。进入循环的条件是k<n*n。

```c
while (k<n * n){
```

在从左到右和从上到下循环时，分别让k和j（列数）；k和i（行数）自增。条件是j/i小于最大列数/行数，且这个数的下一个坐标没有数。

```c
while(j < n-1 && a[i][j +1]== 0){
a[i][j] = k++;j++;
}
while (i< n-1 && a[i + 1][j] == 0){
a[i][j] = k++;i++;
}
```

在从右到左和从下到上循环时，同样让k自增，而j和i自减。条件是j/i大于最小列数/行数且下一个坐标没有数。

```c
while (j> 0 && a[i][j- 1] == 0){
a[i][j] = k++;j--;
}
while (i > 0 && a[i - 1][j] == 0){
a[i][j] = k++;i--;
}
```

这样最后一个数是0，所以额外让最后一个数是n*n。

```
a[i][j]=n*n;
```

最后遍历数组，按每个数字占用三个字符的格式输出。

```
for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        printf("%3d",a[i][j]);
    }
printf("\n");}
```

完整代码：

```
#include<stdio.h>
int a[50][50];
int main()
{

    int i=0,j=0;
    int n,k=1;


    scanf("%d",&n);
   while (k<n * n){
while(j < n-1 && a[i][j +1]== 0){
a[i][j] = k++;j++;
}
while (i< n-1 && a[i + 1][j] == 0){
a[i][j] = k++;i++;
}
while (j> 0 && a[i][j- 1] == 0){
a[i][j] = k++;j--;
}
while (i > 0 && a[i - 1][j] == 0){
a[i][j] = k++;i--;
}

}
a[i][j]=n*n;


    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%3d",a[i][j]);
        }
    printf("\n");}
    return 0;
```

测试：1.

```
4
  1  2  3  4
 12 13 14  5
 11 16 15  6
 10  9  8  7
```

2.

```
6
  1  2  3  4  5  6
 20 21 22 23 24  7
 19 32 33 34 25  8
 18 31 36 35 26  9
 17 30 29 28 27 10
 16 15 14 13 12 11
```

3.

```
 8
  1  2  3  4  5  6  7  8
 28 29 30 31 32 33 34  9
 27 48 49 50 51 52 35 10
 26 47 60 61 62 53 36 11
 25 46 59 64 63 54 37 12
 24 45 58 57 56 55 38 13
 23 44 43 42 41 40 39 14
 22 21 20 19 18 17 16 15
```

4.

```
10
  1  2  3  4  5  6  7  8  9 10
 36 37 38 39 40 41 42 43 44 11
 35 64 65 66 67 68 69 70 45 12
 34 63 84 85 86 87 88 71 46 13
 33 62 83 96 97 98 89 72 47 14
 32 61 82 95100 99 90 73 48 15
 31 60 81 94 93 92 91 74 49 16
 30 59 80 79 78 77 76 75 50 17
 29 58 57 56 55 54 53 52 51 18
 28 27 26 25 24 23 22 21 20 19
```

5.

```
2
  1  2
  4  3
```

函数

你也是函数大师

restrict作用：用于告诉编译器，对象已经被指针所引用，不能通过除该指针外所有其他直接或间接的方式修改该对象的内容。

解释输出结果由来：n1是str字符串长度，经过for循环后，str字符串中是Embedded，strlen作用是计算字符串长度但不包括“\0”，Embedded长度是8，所以n1=8。

str1是字符串str，所以str1=Embedded。