MAZ题writeup

原创

黑白佩 于 2019-03-16 18:25:52 发布 157 收藏

分类专栏： writeup

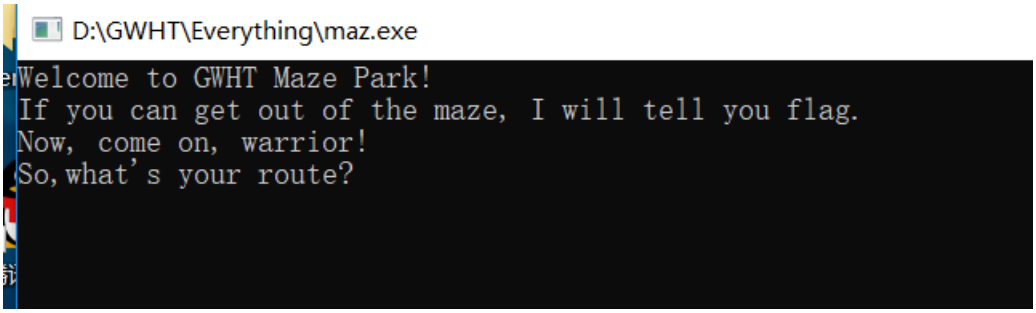
本文链接：https://blog.csdn.net/weixin_44075052/article/details/88602871

一、题目备份

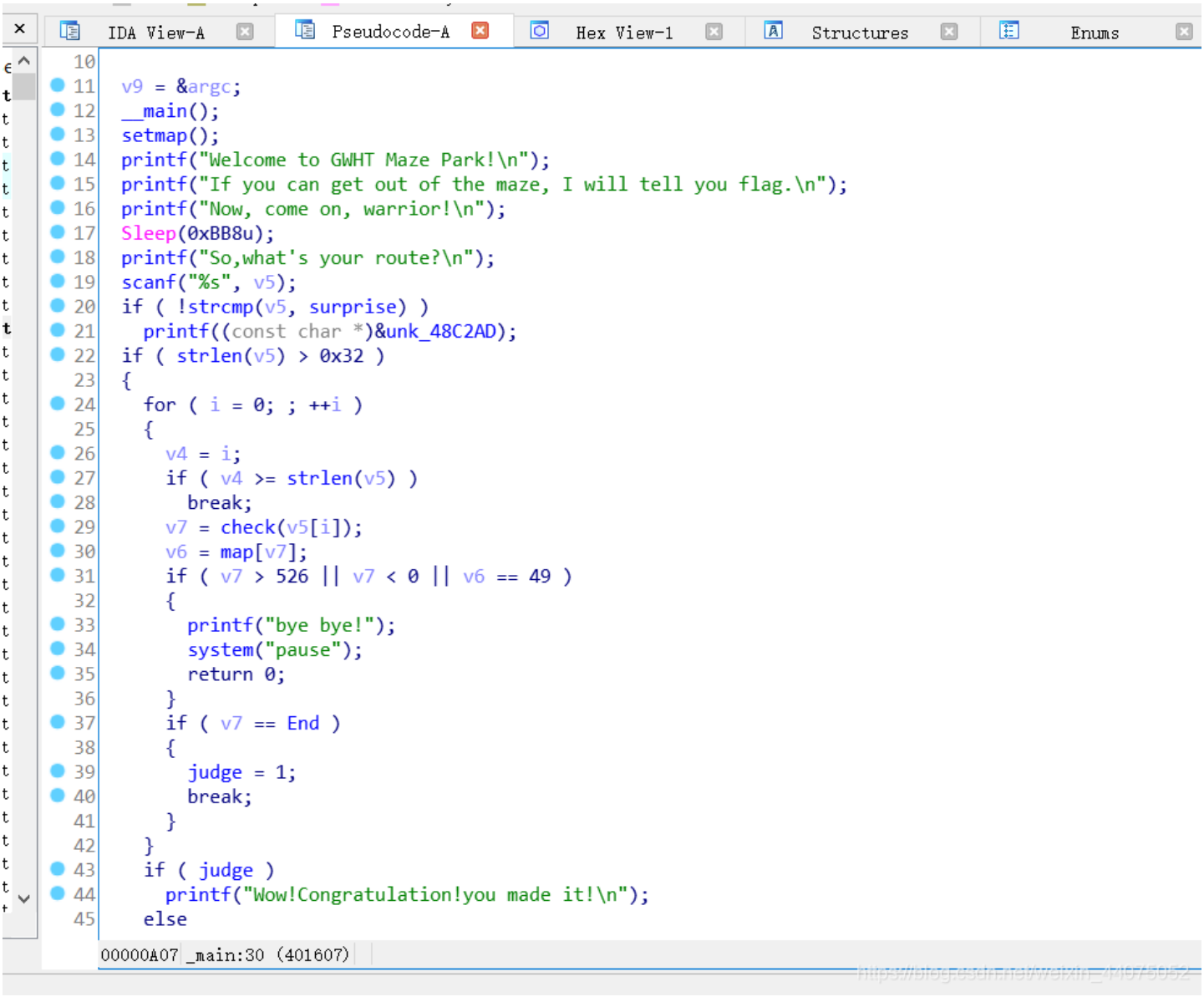
maz 提取码:9v16

二、题目探索

下载下来后发现是exe文件，可以直接打开，于是打开后如下



信息不多，于是把它放入32IDA后可以看到

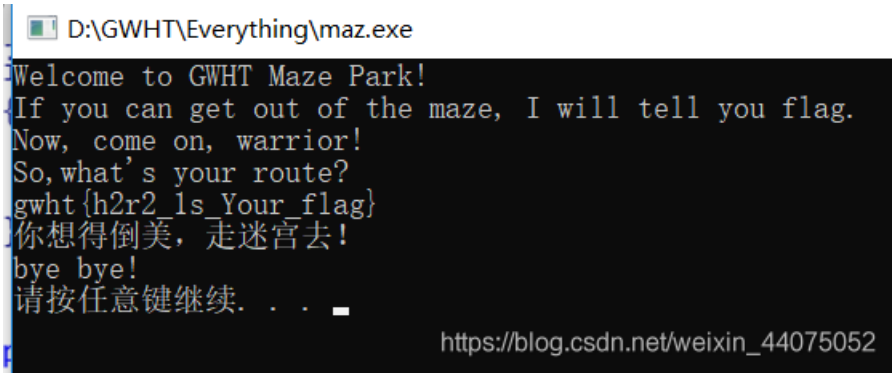



```
10
11  v9 = &argc;
12  __main();
13  setmap();
14  printf("Welcome to GWHT Maze Park!\n");
15  printf("If you can get out of the maze, I will tell you flag.\n");
16  printf("Now, come on, warrior!\n");
17  Sleep(0xBB8u);
18  printf("So,what's your route?\n");
19  scanf("%s", v5);
20  if ( !strcmp(v5, surprise) )
21    printf((const char *)&unk_48C2AD);
22  if ( strlen(v5) > 0x32 )
23  {
24    for ( i = 0; ; ++i )
25    {
26      v4 = i;
27      if ( v4 >= strlen(v5) )
28        break;
29      v7 = check(v5[i]);
30      v6 = map[v7];
31      if ( v7 > 526 || v7 < 0 || v6 == 49 )
32      {
33        printf("bye bye!");
34        system("pause");
35        return 0;
36      }
37      if ( v7 == End )
38      {
39        judge = 1;
40        break;
41      }
42    }
43    if ( judge )
44      printf("Wow!Congratulation!you made it!\n");
45    else
```

好奇的我看到了surprise的对比，IDA显示surprise也是一个字符型数组（char[]），于是我便去试了一下

```
.data:0048500C                 public _surprise
.data:0048500C ; char surprise[]
.data:0048500C _surprise       db 'gwht{h2r2_1s_Your_flag}',0
.data:0048500C                                         ; DATA XREF: _main
```

结果却让我**大吃一惊**==



认真揣摩代码，发现在这里有一个setmap函数

```
  8  size_t i; // [esp+CEh] [ebp-Ch]
  9  int *v9; // [esp+D2h] [ebp-8h]
 10
 11  v9 = &argc;
 12  __main();
 13  setmap();
 14  printf("Welcome to GWHT Maze Park!\n");
 15  printf("If you can get out of the maze, I will tell you flag.\n");
 16  printf("Now, come on, warrior!\n");
 17  Sleep(0xBB8u);
 18  printf("So,what's your route?\n");
 19  scanf("%s", v5);
 20  if ( !strcmp(v5, surprise) )
```



```
 1 void *setmap(void)
 2 {
 3   return memcpy(
 4            map,
 5            "11111111111111111111111111111111111111111111111111111111111111111111111111**********s11111111111111111*1111111"
 6            "11111111111111111111*1111111111111111111111111111*1111111111111111111111111111*11111111111111111111"
 7            "1111111*11111111111111111111111111111*11111111111111111111111111111*1e*********1111111111111111*11111111"
 8            "111*1111111111111111*1111111111*1111111111111111*1111111111*1111111111111111*1111111111*111111111111"
 9            "111111************1111111111111111111111111111111111111111111111111111111111111111111111111gwht{h2r2_1s_Your_flag}",
10            0x20Fu);
11 }
```


既然是走迷宫，这里应该是类似布置迷宫地图的函数，而且0x20F转化为十进制是527，也就是memcpy刚好只拷贝到gwht前面的字符到map里面，我们继续往下看，能找到一个循环，仔细一看会发现它其实类似于一种认证环节

```
if ( strlen(v5) > 0x32 )
{
  for ( i = 0; ; ++i )
  {
    v4 = i;
    if ( v4 >= strlen(v5) )
      break;
    v7 = check(v5[i]);
    v6 = map[v7];
    if ( v7 > 526 || v7 < 0 || v6 == 49 )
    {
      printf("bye bye!");
      system("pause");
      return 0;
    }
    if ( v7 == End )
    {
```

```
      judge = 1;
      break;
    }
  }
  if ( judge )
    printf("Wow!Congratulation!you made it!\n");
  else
    printf("bye bye!");
  system("pause");
  result = 0;
}
```


v5是我们输入的数据，输入下来会进到一个check函数，出来后的数值相当于踩到map上，进行一次IF检测。**鼠标轻碰IF里面的49显示其实是一个char数值 对应的是 '1' 意思是踩到的不能是地图上的 '1' 也不能超过这个地图范围内，否则就byebye，**而只有当出来的值等于End的时候就会出现Congratulation的提示，而End的值我们也可以点进去查看

```
.data:00485008                  public _End
.data:00485008 _End             dd 121h
.data:0048500C                  public surpri
```

121十六进制转化为十进制就是289
我们进去check函数里看一下，发现操纵的其实是一个start的值
start的值点进去也是有初始值的

```
22  if ( strlen(v5) > 0x32 )
23  {
24    for ( i = 0; ; ++i )
25    {
26      v4 = i;
27      if ( v4 >= strlen(v5) )
28        break;
29      v7 = check(v5[i]);
30      v6 = map[v7];
31      if ( v7 > 526 || v7 < 0 || v6 == 49 )
32      {
33        printf("bye bye!");
34        system("pause");
35        return 0;
36      }
37      if ( v7 == End )
38      {
39        judge = 1;
40        break;
41      }
42    }
```


IDA View-A | Pseudocode-A

```
1  int __cdecl check(char a1)
2  {
3    char v2; // [esp+4h] [ebp-4h]
4
5    v2 = change(a1);
6    if ( v2 == 105 )
7      start -= 31;
8    if ( v2 == 121 )
9      start += 31;
10   if ( v2 == 118 )
11     --start;
```

```
12  if ( v2 == 122 )
13    ++start;
14  return start;
15}
```



```
.data:00485004                  public _start
.data:00485004 _start           dd 52h
.data:00485004
```

52十六进制转化为十进制就是82，start开始就是82

IDA View-A | Pseudocode-A

```
1int __cdecl check(char a1)
2{
3  char v2; // [esp+4h] [ebp-4h]
4
5  v2 = change(a1);
6  if ( v2 == 105 )
7    start -= 31;
8  if ( v2 == 121 )
9    start += 31;
10 if ( v2 == 118 )
11   --start;
12 if ( v2 == 122 )
13   ++start;
14 return start;
15}
```



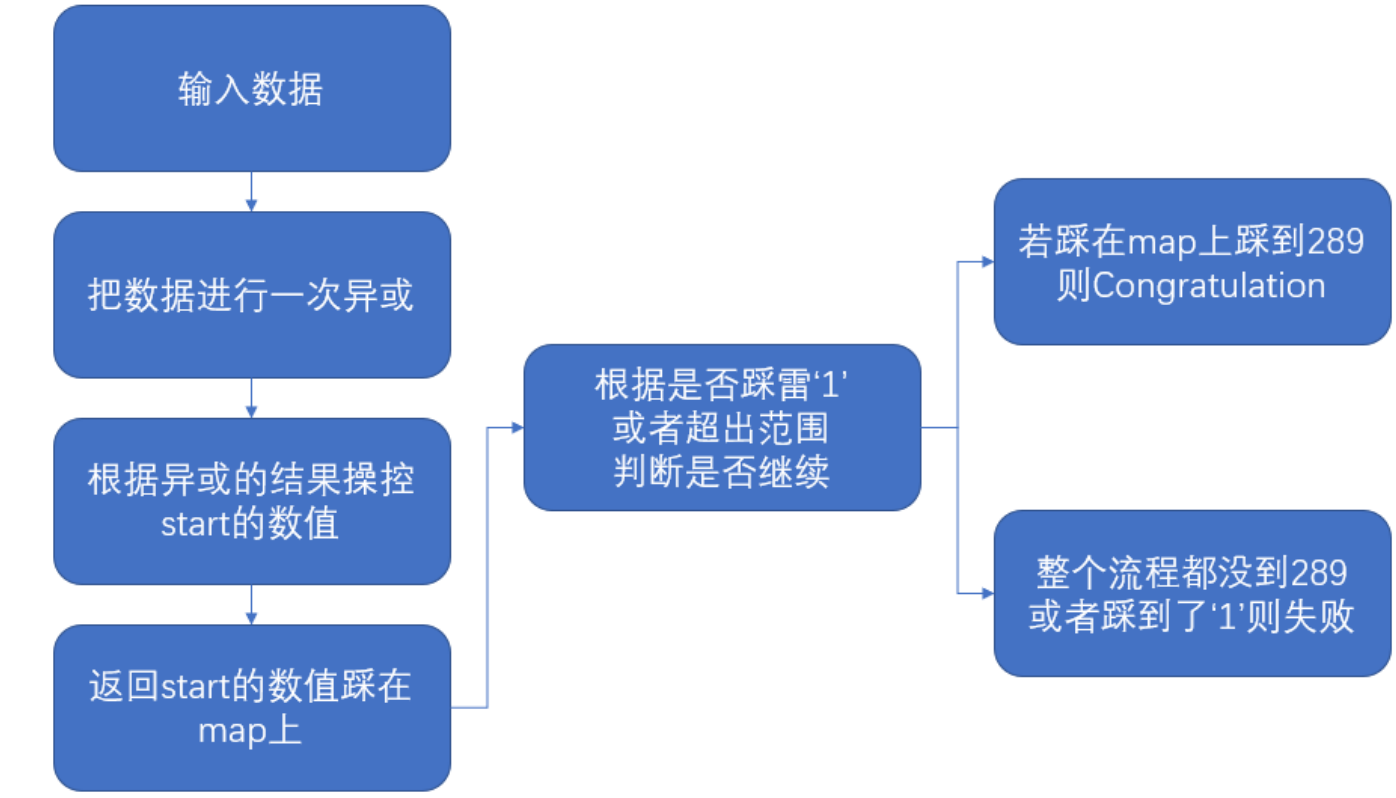
```
1int __cdecl check(char a1)
2{
3  char v2; // [esp+4h] [ebp-4h]
4
5  v2 = change(a1);
6  if ( v2 == 105 )
7    start -= 31;
8  if ( v2 == 121 )
9    start += 31;
10 if ( v2 == 118 )
11   --start;
12 if ( v2 == 122 )
13   ++start;
14 return start;
15}
```


然而在check函数里面还有一个change函数，再进去后发现是把输入的数值，进行一次异或

IDA View-A | Pseudocode-A | Hex

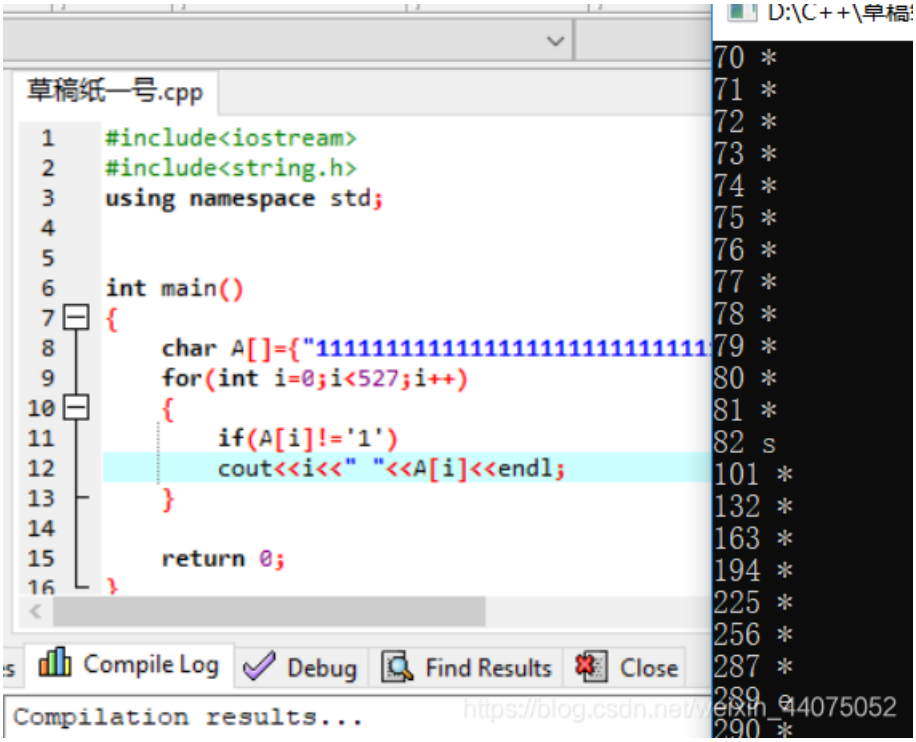
```
1int __cdecl change(char a1)
2{
3  return ((unsigned __int8)a1 ^ 6) + 8;
4}
```

到这里我们整理一下这道题的思绪，差不多可以得出这样的思维导图



三、解决思路

我想先找到地图上有哪些位置是可以踩的，于是写了一个代码稍微还原了一些地图



在这里可以看到在82上有个s(start)，在289上有个e(end)，其实寓意着一个是迷宫的出发点，一个是迷宫的结束点，只要我们能使start数最后以289出来就行

然后继续探究操作start的check函数，把后面的条件里的数按R转换为char型
将他条件里的数值按照他的异或反着算回去

IDA View-A | Pseudocode-A

```
1  int __cdecl check(char a1)
2  {
3    char v2; // [esp+4h] [ebp-4h]
4
5    v2 = change(a1);
6    if ( v2 == 'i' )
7      start -= 31;
8    if ( v2 == 'y' )
9      start += 31;
10   if ( v2 == 'v' )
11     --start;
12   if ( v2 == 'z' )
13     ++start;
14   return start;
15 }
```

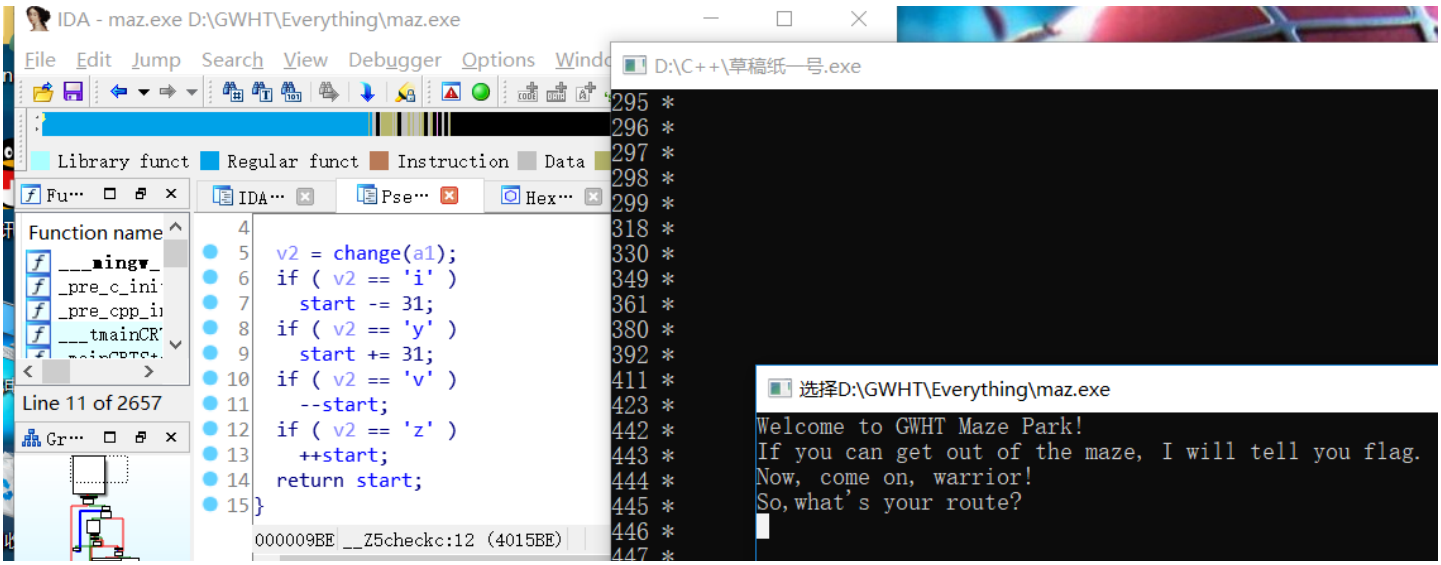

草稿纸一号.cpp

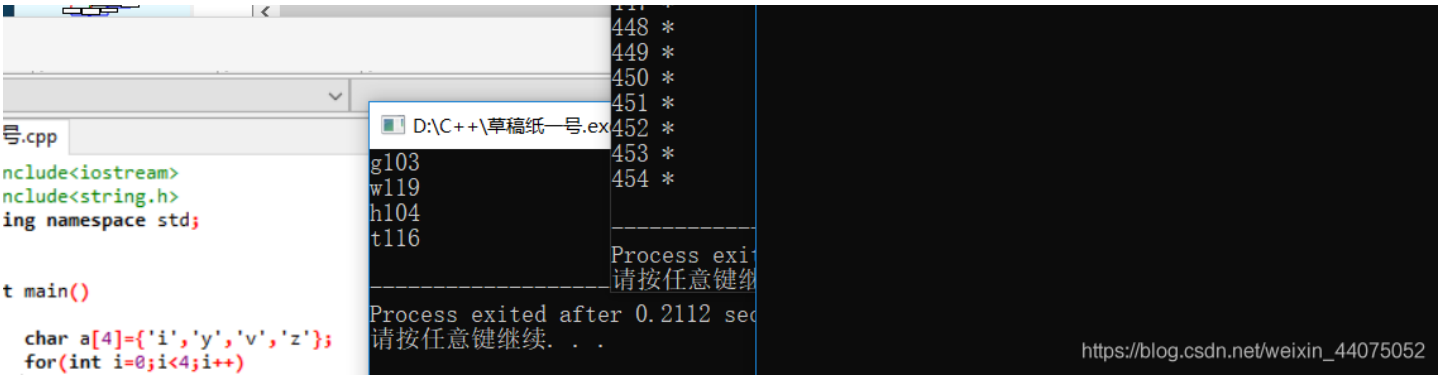
```
1  #include<iostream>
2  #include<string.h>
3  using namespace std;
4
5
6  int main()
7  {
8      char a[4]={'i','y','v','z'};
9      for(int i=0;i<4;i++)
10         a[i]=(a[i]-8)^6;
11     for(int i=0;i<4;i++)
12     {
13         cout<<a[i];
14         cout<<(int)a[i]<<endl;
15     }
16     return 0;
```

D:\C++\草稿纸一号.exe

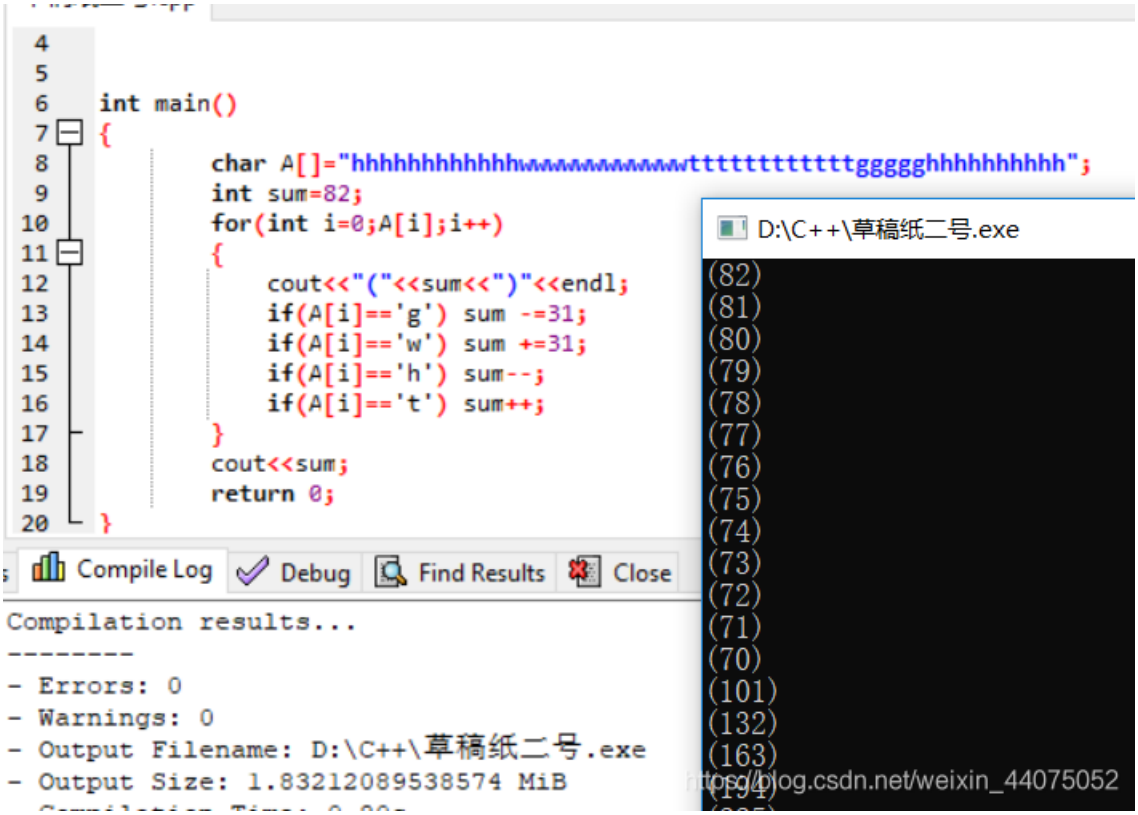
```
g103
w119
h104
t116

--------------------------------
Process exited after 0.2112 seconds with return value 0
请按任意键继续. . .
```


可以得到输入 **g–将start减31， w–将start加31， h–将start减1， t–将start加1**
顺着地图和操作数把这个迷宫慢慢走完= =





当然看一步走一步是非常难受的，而且很容易看错，所以答案并不是我一次过的，我甚至还打了个代码让我自己核对着地图看看我走的对不对



最后终于得到答案