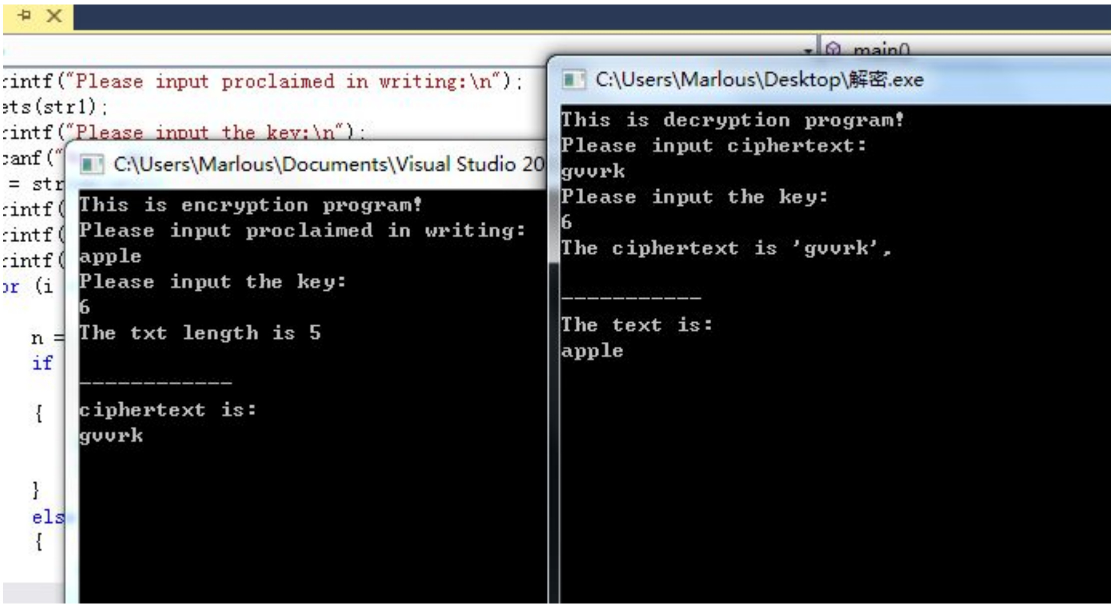
加密解密算法分析



```
:intf("Please input proclaimed in writing:\n");
ets(str1);
:intf("Please input the key:\n");
:anf("
 = str
:intf(
:intf(
:intf(
or (i

  n =
  if

  {


  }
  els
  {
```

C:\Users\Marlous\Documents\Visual Studio 20

```
This is encryption program!
Please input proclaimed in writing:
apple
Please input the key:
6
The txt length is 5

------------
ciphertext is:
gvvrk
```

C:\Users\Marlous\Desktop\解密.exe

```
This is decryption program!
Please input ciphertext:
gvvrk
Please input the key:
6
The ciphertext is 'gvvrk',

-----------
The text is:
apple
```

加密源代码：

```
#include<stdio.h>
#include<math.h>
#include<string.h>
#define N 500
int main()
{
        int i = 0, k, m, n, l;
        char str1[N], str2[N];
        printf("This is encryption program!\n");
        printf("Please input proclaimed in writing:\n");
        gets(str1);
        printf("Please input the key:\n");
        scanf("%d", &k);
        m = strlen(str1);
        printf("The txt length is %d\n", m);
        printf("\n------------\n");
        printf("ciphertext is:\n");
        for (i = 0; i<m; i++)
        {
                n = (int)str1[i];
                if
                        (str1[i] == ' ')
        {
            printf(" ");
            str2[i] = str1[i];
        }
        else if (n>96 && n < 123)
        {
            n = (n - 97 + k) % 26;
            if (n<0)
                n = 26 + n;
            l = (char)(n + 97);
            printf("%c", l);
            str2[i] = l;
        }
        else if (n>64 && n < 91)
        {
            n = (n - 65 + k) % 26;
            if (n < 0)
                n = 26 + n;
            l = (char)(n + 97);
            printf("%c", l);
            str2[i] = l;
        }
    }
    str2[i] = '\0';
    getch();
    return 0;
}
```

算法分析：

数组 str1 存放待加密的明文，str2 存放经变换的密文，k 存放密钥，n 为字符的 ASCII 码数值，m 为明文长度。

通过下标对数组内每个字符以密钥 k 的值进行变换。

空格以原样输出，
小写字母 n>96 && n < 123，
n = (n - 97 + k) % 26，
大写字母 n>64 && n < 91，
n = (n - 65 + k) % 26，
此时 n 变为 26 字母表中字母编号，加上 97 的偏移量变为 ASCII 码表中对应字符，值赋给变量 l,
输出。
解密为逆过程。

解密源代码：

```
#include<stdio.h>
#include<math.h>
#include<string.h>
#define N 500
int main()
{
    int i = 0, k, m, n, l;
    char    str1[N], str2[N];
    printf("This is decryption program!\n");
    printf("Please input ciphertext:\n");
    gets(str2);
    printf("Please input the key:\n");
    scanf("%d", &k);
    printf("The ciphertext is '%s',\n\n-----------\nThe text is:\n", str2);
    m = strlen(str2);
    for (i = 0; i<m; i++)
    {
        n = (int)str2[i];
        if (str2[i] == ' ')
        {
            printf(" ");
        }
        else if (n>96 && n<123)
        {
            n = (n - 97 - k) % 26;
            if (n<0)
                n = 26 + n;
            l = (char)(n + 97);
            printf("%c", l);
        }
        else if (n>64 && n<91)
        {
            n = (n - 65 - k) % 26;
            if (n<0)
                n = 26 + n;
            l = (char)(n + 97);
            printf("%c", l);
        }
    }
    str2[i] = '\0';
    getch();
    return 0;
}
```