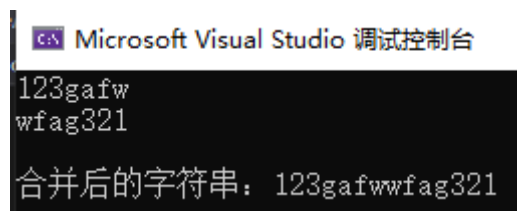


18. 由用户从键盘输入两个字符串；然后把两个字符串合并在一起；最后显示输出合并后的字符串

```
#include <stdio.h>
char a[200], b[100];
int main() {
    gets_s(a);
    gets_s(b);
    __asm {
        LEA EDI, a
        LEA ESI, b
        MOV ECX, -1
        XOR AL, AL
        REPNE SCASB // 到a的末尾+1
        DEC EDI // 到a的末尾
        XOR ECX, ECX
    CONCAT:
        MOVSB
        CMP BYTE PTR[ESI], AL
        LOOPNE CONCAT
        MOV [EDI], 0
    }
    printf("\n合并后的字符串: %s\n", a);
    return 0;
}
```

结果:



```
Microsoft Visual Studio 调试控制台
123gafw
wfag321
合并后的字符串: 123gafwwfag321
```

21. 由用户从键盘输入两个字符串 str1 和 str2；查找确定str2 在 str1中出现的起始位置（如果未出现，则设起始位置为 -1）；显示输出起始位置值

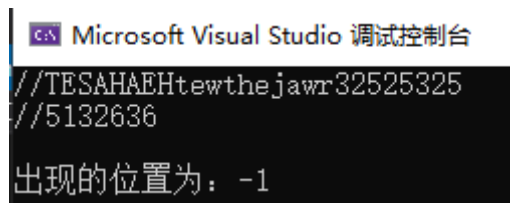
```
#include <stdio.h>
char a[200], b[100];
int main() {
    gets_s(a);
    gets_s(b);
    int i;
    __asm {
        LEA ESI, a // a 是源
        LEA EDI, b // b 是目的
        MOV EBX, ESI
        MOV ECX, -1
        XOR AL, AL
        REPNE SCASB
        NOT ECX
    }
```

```

    DEC ECX // len(b)
    JE NOTFOUND
    MOV EDX, ECX;
FORI:
    MOV ESI, EBX
    MOV ECX, EDX;
    LEA EDI, b
FORJ:
    MOV AL, BYTE PTR[ESI] // 取a的字符
    CMP BYTE PTR[EDI],AL // 和b的字符相比
    JNE NEXTI; //不等于则判断a是否已经结束
NEXTJ:
    INC EDI
    INC ESI
    LOOP FORJ // 等于则继续a和b比较
    JMP OVER; // b已经结束了
NEXTI:
    INC EBX
    CMP BYTE PTR[EBX], 0
    JNE FORI
    JMP NOTFOUND
OVER:
    MOV EAX, ESI
    LEA ESI, a
    SUB EAX, ESI
    DEC EDX
    SUB EAX, EDX
    JMP GETRESULT
NOTFOUND:
    MOV EAX, -1
GETRESULT:
    MOV i, EAX;
}
//TESAHAEHtewthejawr32525325
//5132636
printf("\n出现的位置为: %d\n", i);
return 0;
}

```

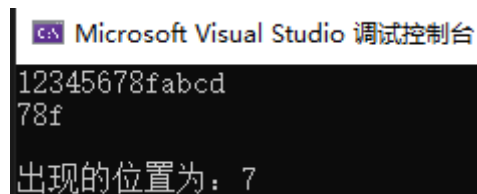
结果:



```

Microsoft Visual Studio 调试控制台
//TESAHAEHtewthejawr32525325
//5132636
出现的位置为: -1

```



```

Microsoft Visual Studio 调试控制台
12345678fabcd
78f
出现的位置为: 7

```

24. 由用户从键盘分别输入两个十六进制数值; 然后求他们的和与差; 最后才用十六进制数的格式分别输出结果。假设只能采用字符串格式实现输入和输出, 并且应充分采用字符串操作指令。

设计子程序，把一个由十六进制数字字符构成的字符串转换成对应的数值，注意字符串可能有前导空格等。设计子程序，判断一个字符是否是十六进制数的字符。

```
#include <stdio.h>
char a[100], b[100];
char sum[10], difference[10];
char hexTable[] = "0123456789abcdefABCDEF";
char check;
int main() {
    printf("请输入两个十六进制数值a 和 b\n");
    gets_s(a);
    gets_s(b);

    __asm {
        LEA EDI, b
        PUSH EDI
        CALL STRTOHEX
        POP EDI
        PUSH EAX

        LEA EDI, a
        PUSH EDI
        CALL STRTOHEX
        POP EDI
        PUSH EAX

        MOV ECX, [ESP] // a
        ADD ECX, [ESP + 4] // b
        LEA EDX, sum
        CALL HEXTOSTR

        MOV ECX, [ESP]
        SUB ECX, [ESP + 4]
        LEA EDX, difference
        CALL HEXTOSTR
        POP EAX
        POP EAX
    }

    printf("\na + b = 0x%s\n", sum);
    printf("a - b = 0x%s\n", difference);
    printf("\n请输入需要检查的字符\n");
    scanf_s("%c", &check, 1);
    bool flag;
    __asm {
        MOV AL, check;
        CALL CHECKHEX;
        MOV flag, AL;
    }
    if(flag) printf("是十六进制字符\n");
    else printf("不是十六进制字符\n");
    return 0;
    __asm { // 16进制字符串转数值
        STRTOHEX:
        PUSH EBP
        MOV EBP, ESP
```

```

MOV EDI, [ESP + 8]; // 得到字符串
MOV ECX, -1;
MOV AL, 20H;
REPE SCASB;
DEC EDI; // 去空格

MOV ECX, 16;
XOR EAX, EAX
PUSH EBX
PUSH EDI
XOR EBX, EBX
ALoop:
    MOV BL, BYTE PTR[EDI]
    SUB BL, '0'
    CMP BL, 9
    JBE CALCULATE
    SUB BL, 17
    CMP BL, 5
    JBE ADD_NUM
    SUB BL, 32
    CMP BL, 5
    JA CALCULATE
ADD_NUM:
    ADD BL, 10
CALCULATE:
    MUL ECX
    ADD EAX, EBX
    INC EDI
    CMP BYTE PTR[EDI], 0
    JNE ALoop
POP EDI
POP EBX
POP EBP
RET
}

```

```

__asm {
HEXTOSTR:
    PUSH EBP
    MOV EBP, ESP
    PUSH -1
    BLoop :
        MOV EAX, ECX
        AND AL, 0x0f
        ADD AL, '0'
        CMP AL, '9'
        JLE BNext
        ADD AL, 7
    BNext:
        PUSH EAX
        SHR ECX, 4
        TEST ECX, ECX
        JE BDeal
        JMP BLoop
    BDeal:
        POP EAX
        CMP EAX, -1
        JE BOVER
}

```

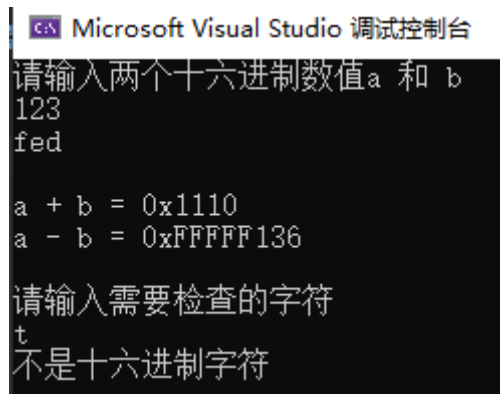
```

        MOV BYTE PTR[EDX], AL
        INC EDX
        JMP BDEAL
BOVER:
        POP EBP
        RET
    }

__asm {
CHECKHEX:
    PUSH EBP;
    MOV EBP, ESP;
    PUSH EDI;
    MOV ECX, 22;
    LEA EDI, hexTable;
NEXT:
    SCASB;
    LOOPNE NEXT;
    JNE NOTFOUND;
FOUND:
    MOV AL, 1;
    JMP SHORT OVER;
NOTFOUND:
    MOV AL, 0;
OVER:
    POP EDI;
    POP EBP;
    RET
}
}

```

结果:



```

Microsoft Visual Studio 调试控制台
请输入两个十六进制数值a 和 b
123
fed

a + b = 0x1110
a - b = 0xFFFF136

请输入需要检查的字符
t
不是十六进制字符

```

```
Microsoft Visual Studio 调试控制台
请输入两个十六进制数值a 和 b
4fa
6346f

a + b = 0x63969
a - b = 0xFFFF9D08B

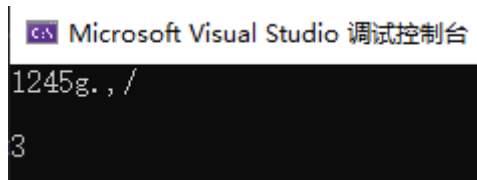
请输入需要检查的字符
A
是十六进制字符
```

27. 由用户输入一个字符串，统计其中标点符号的个数，显示输出统计结果。

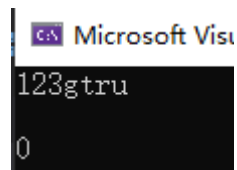
```
#include <stdio.h>
char a[100];
char interpunction[] = ".?!,:;\\"'\"'[]~<>/";
int main() {
    gets_s(a);
    int sum;
    __asm {
        LEA ESI, a // a 是源
        XOR EBX, EBX; // 计数
        CMP [ESI], 0;
        JE OVER;
        LEA EDI, interpunction; // b 是目的
    FORI:
        MOV ECX, 17; // len(interpunction)
        LEA EDI, interpunction
    FORJ:
        MOV AL, BYTE PTR[ESI] // 取a的字符
        CMP BYTE PTR[EDI], AL // 和interpunction的字符相比
        JE DEAL; //等于则判断a是否已经结束
    NEXTJ:
        INC EDI;
        LOOP FORJ; // 等于则继续a和b比较
    DEAL:
        CMP ECX, 0;
        JE NEXTI;
        INC EBX;
    NEXTI:
        INC ESI;
        CMP BYTE PTR[ESI], 0;
        JNE FORI;
    OVER:
        MOV sum, EBX;
    }

    printf("\n%d\n", sum);
    return 0;
}
```

结果：



```
Microsoft Visual Studio 调试控制台
1245g./
3
```



```
Microsoft Visual Studio 调试控制台
123gtru
0
```

28. 由用户输入一个字符串（假设非空），将其视为由若干个单字节数据构成的数组；分别统计正数和偶数的个数；显示输出统计结果。

```
#include <stdio.h>
char a[100];
int main() {
    gets_s(a);
    int positive, even;
    __asm {
        XOR AL, AL; // 结束标志
        XOR ESI, ESI; // 正数
        XOR ECX, ECX;
        XOR EDX, EDX; // 偶数
        LEA EDI, a;
        CMP [EDI], 0;
        JE OVER;
    ALOOP:
        BT BYTE PTR[EDI], 0; // 判断奇数
        ADC EDX, 0;
        BT BYTE PTR[EDI], 7; // 判断正数
        ADC ESI, 0;
        SCASB
        LOOPNE ALOOP;
    OVER:
        NOT ECX; // 字符串长度
        MOV EAX, ECX;
        SUB EAX, ESI; // 正数个数 = 字符串长度 - 负数个数
        MOV positive, EAX;
        MOV EAX, ECX;
        SUB EAX, EDX;
        MOV even, EAX; // 偶数个数 = 字符串长度 - 奇数个数
    }

    printf("\n正数个数: %d\n", positive);
    printf("偶数个数: %d\n", even);
    return 0;
}
```

结果：

```
C# 选择Microsoft Visual Studio 调试控制台  
this is a test  
正数个数: 14  
偶数个数: 7
```