

# C-01

## 题目分析

对字符串处理成数字后求平均值。

对整数和小数分开处理。

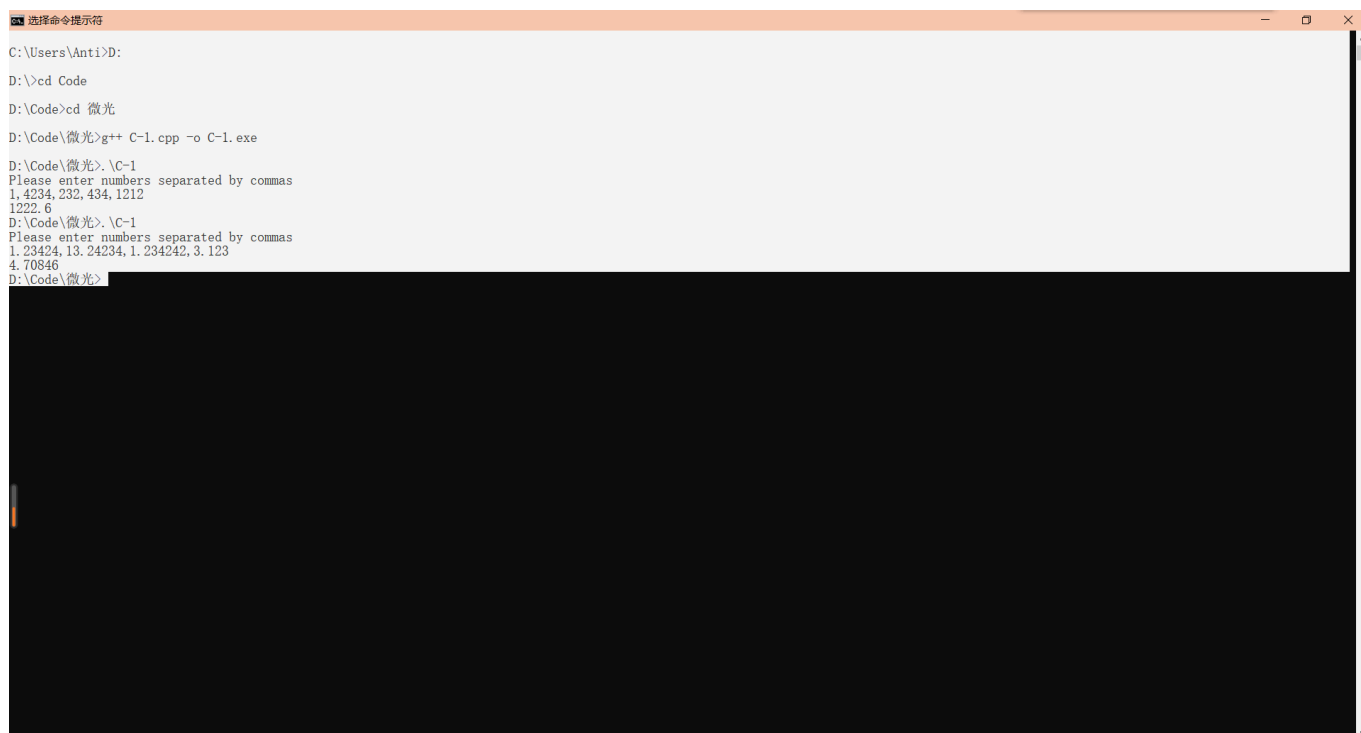
## 我的回答

1. 我认为是指针传递了数组第一个元素的地址，实参就是我输入的那个字符串。  
递进入函数后会被存在栈内存里，函数结束后内存被释放。原本被传递的变量不能被修改。指针传递了变量的内存位置，函数可以通过直接访问内存来修改值。

比如我在我的预处理函数`work`中使用了`int *cnt`我可以通过在函数中修改指针`cnt`对应地址中的值来使`solve`函数中对应变量也被修改，而值传递不能实现这个功能。其实我觉得这种小代码直接开全局变量方便一些

2. 。就是传递到对应参数啊啊。
3. `const`表示常量。保护数值不被修改。举个例子，如果`const int *cnt`。在尝试编译时会报错，显示这个只读。`const &`也可以引用。

## 运行截图



```
选择命令提示符
C:\Users\Anti>D:
D:\>cd Code
D:\Code>cd 微光
D:\Code\微光>g++ C-1.cpp -o C-1.exe
D:\Code\微光>. \C-1
Please enter numbers separated by commas
1, 4234, 232, 434, 1212
1222, 6
D:\Code\微光>. \C-1
Please enter numbers separated by commas
1. 23424, 13. 24234, 1. 234242, 3. 123
4. 70846
D:\Code\微光>
```

## 代码部分

ps 感觉没啥用，我是直接输入字符串^ ^

```
#include <bits/stdc++.h>
using namespace std;
```

```
template <typename T>
double Ave(const T *n, const int &cnt)
{
    double sum = 0;
    for (int i = 1; i <= cnt; i++)
    {
        sum += n[i];
    }
    return sum / (double)cnt;
}
//对字符预处理
void work_1(int *num, int *cnt, const char *s)
{
    int now = 0;
    int ct = 0;
    for (int i = 0; i < strlen(s); i++)
    {
        if (s[i] == ',')
        {
            num[++ct] = now;
            now = 0;
        }
        else
            now = now * 10 + (s[i] - '0');
    }
    num[++ct] = now; //注意最后一个数字后没有逗号
    *cnt = ct;
    return;
}
void work_2(double *num, int *cnt, const char *s)
{
    double now1, now2, xiaoShu = 1;
    int ct;
    bool flag = true; //true表示当前正在记录整数, false表示记录小数
    for (int i = 0; i < strlen(s); i++)
    {
        if (s[i] == ',')
        {
            num[++ct] = now1 + now2;
            now1 = now2 = 0;
            flag = true;
            xiaoShu = 1;
            continue;
        }
        if (s[i] == '.')
        {
            flag = false;
            continue;
        }
        if (flag)
            now1 = now1 * 10 + (s[i] - '0');
        else
        {

```

```
        xiaoShu /= 10;
        now2 = now2 + xiaoShu * (s[i] - '0');
    }
}
num[++ct] = now1 + now2;
*cnt = ct;
return;
}
double solve(const char *s)
{
    double ans;
    bool flag = true; //默认为整数
    for (int i = 0; i <= strlen(s); i++)
        if (s[i] == '.')
        {
            flag = false;
            break;
        }
    if (flag)
    {
        int num[10000];
        int cnt = 0;
        double sum = 0;
        work_1(num, &cnt, s);
        return Ave<int>(num, cnt);
    }
    else
    {
        int cnt = 0;
        double num[10000];
        work_2(num, &cnt, s);
        return Ave<double>(num, cnt);
    }
    return -1;
}
int main()
{
    char s[10086];
    memset(s, 0, sizeof(s));
    cout << "Please enter numbers separated by commas" << endl;
    gets(s);
    cout << solve(s);
    return 0;
}
```