目录

**[综合研究12 1](#_Toc16713)**

[问题1 1](#_Toc16297)

[问题2 3](#_Toc31905)

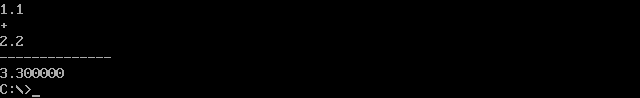
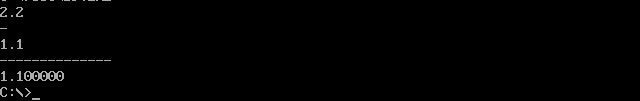
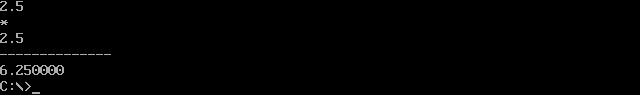
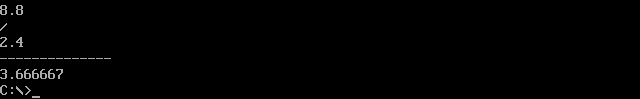
**[综合研究13 3](#_Toc27426)**

# 综合研究12

### 问题1

要将a.c程序扩展为可以处理浮点型数据的加减乘除操作。（研究atoi和itoa类似功能的函数，要写程序测试各个函数的功能）

#include <stdio.h>  
#include <stdlib.h>  
  
char \*codes = "+-\*/";  
double add(double a, double b) {  
 printf("%lf,%lf", a, b);  
 return a + b;  
}  
double sub(double a, double b) { return a - b; }  
double mul(double a, double b) { return a \* b; }  
double div\_d(double a, double b) {  
 if (b == 0) {  
 printf("error!");  
 return -1;  
 } else  
 return a / b;  
}  
double (\*func[4])(double, double) = {add, sub, mul, div\_d};  
main() {  
 char a[20];  
 char b[20];  
 char ch;  
 double d\_a = 0, d\_b = 0;  
  
 int n;  
  
 gets(a);  
 printf("%c\n", ch = getch());  
 gets(b);  
  
 d\_a = atof(a);  
 d\_b = atof(b);  
  
 for (n = 0; codes[n] && codes[n] != ch; n++)  
 ;  
 if (!codes[n]) {  
 printf("error!");  
 return;  
 }  
  
 printf("--------------\n");  
  
 printf("%lf", func[n](d\_a, d\_b));  
}

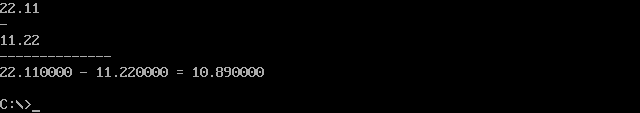
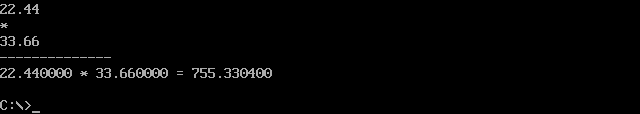
* 加法
* 
* 减法
* 
* 乘法
* 
* 除法
* 

### 问题2

考虑是否可以将综合研究12 a.c程序中的a和b数组定义为char \* 类型。

可以但前提是在使用前通过malloc来分配一块内存然后再继续使用，不然都的话数据可能会丢失。

# 综合研究13

* ar.h
* typedef struct {  
   double a;  
   double b;  
   char ch;  
  } pra;  
  typedef struct {  
   char op;  
   void (\*f)(double, double);  
  } exType;  
    
  void arithmetic(pra, exType \*, int);  
  void arithmetic(pra p, exType \*ex, int n) {  
   char ch = p.ch;  
   int a;  
    
   for (a = 0; ex[a].op != ch && a < n; a++)  
   ;  
    
   printf("--------------\n");  
   ex[a].f(p.a, p.b);  
  }
* arithmetic.c
* #include "ar.h"  
  #include <stdlib.h>  
    
  void add\_d(double a, double b);  
  void sub\_d(double a, double b);  
  void mul\_d(double a, double b);  
  void div\_d(double a, double b);  
    
  pra p;  
  exType ex[4] = {'+', add\_d, '-', sub\_d, '\*', mul\_d, '/', div\_d};  
    
  int main() {  
   char ca[100], cb[100];  
   char ch;  
    
   gets(ca);  
   printf("%c\n", ch = getch());  
   gets(cb);  
    
   p.a = atof(ca);  
   p.b = atof(cb);  
   p.ch = ch;  
    
   arithmetic(p, ex, 4);  
    
   return 0;  
  }  
    
  void add\_d(double a, double b) { printf("%lf + %lf = %lf\n", a, b, a + b); }  
  void sub\_d(double a, double b) { printf("%lf - %lf = %lf\n", a, b, a - b); }  
  void mul\_d(double a, double b) { printf("%lf \* %lf = %lf\n", a, b, a \* b); }  
  void div\_d(double a, double b) { printf("%lf / %lf = %lf\n", a, b, a / b); }
* 加法
* 
* 减法
* 
* 乘法
* 
* 除法
* 