

综合研究12

问题1

问题2

综合研究13

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）

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  char *codes = "+-*/";
5  double add(double a, double b) {
6      printf("%lf,%lf", a, b);
7      return a + b;
8  }
9  double sub(double a, double b) { return a - b; }
10 double mul(double a, double b) { return a * b; }
11 double div_d(double a, double b) {
12     if (b == 0) {
13         printf("error!");
14         return -1;
15     } else
16         return a / b;
17 }
18 double (*func[4])(double, double) = {add, sub, mul, div_d};
19 main() {
20     char a[20];
21     char b[20];
22     char ch;
23     double d_a = 0, d_b = 0;
24
25     int n;
26
27     gets(a);
28     printf("%c\n", ch = getch());
29     gets(b);
30
31     d_a = atof(a);
32     d_b = atof(b);
33
34     for (n = 0; codes[n] && codes[n] != ch; n++)
35         ;
36     if (!codes[n]) {
37         printf("error!");
```

```

38         return;
39     }
40
41     printf("—————\n");
42
43     printf("%lf", func[n](d_a, d_b));
44 }

```

- 加法

```

1.1
+
2.2
-----
3.300000
C:\>_

```

- 减法

```

2.2
-
1.1
-----
1.100000
C:\>_

```

- 乘法

```

2.5
*
2.5
-----
6.250000
C:\>_

```

- 除法

```

8.8
/
2.4
-----
3.666667
C:\>_

```

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

```

1     typedef struct {
2         double a;
3         double b;
4         char ch;
5     } pra;
6     typedef struct {
7         char op;
8         void (*f)(double, double);
9     } exType;
10

```

```

11 void arithmetic(pra, exType *, int);
12 void arithmetic(pra p, exType *ex, int n) {
13     char ch = p.ch;
14     int a;
15
16     for (a = 0; ex[a].op != ch && a < n; a++)
17         ;
18
19     printf("-----\n");
20     ex[a].f(p.a, p.b);
21 }

```

- arithmetic.c

```

1  #include "ar.h"
2  #include <stdlib.h>
3
4  void add_d(double a, double b);
5  void sub_d(double a, double b);
6  void mul_d(double a, double b);
7  void div_d(double a, double b);
8
9  pra p;
10 exType ex[4] = {'+', add_d, '-', sub_d, '*', mul_d, '/', div_d};
11
12 int main() {
13     char ca[100], cb[100];
14     char ch;
15
16     gets(ca);
17     printf("%c\n", ch = getch());
18     gets(cb);
19
20     p.a = atof(ca);
21     p.b = atof(cb);
22     p.ch = ch;
23
24     arithmetic(p, ex, 4);
25
26     return 0;
27 }
28
29 void add_d(double a, double b) { printf("%lf + %lf = %lf\n", a, b, a +
b); }
30 void sub_d(double a, double b) { printf("%lf - %lf = %lf\n", a, b, a -
b); }
31 void mul_d(double a, double b) { printf("%lf * %lf = %lf\n", a, b, a *
b); }
32 void div_d(double a, double b) { printf("%lf / %lf = %lf\n", a, b, a /
b); }

```

- 加法

```

11.11
+
22.22
-----
11.110000 + 22.220000 = 33.330000
C:\>_

```

- 减法

```
22.11
-
11.22
-----
22.110000 - 11.220000 = 10.890000

C:\>_
```

- 乘法

```
22.44
*
33.66
-----
22.440000 * 33.660000 = 755.330400

C:\>_
```

- 除法

```
22.88
/
1.55
-----
22.880000 / 1.550000 = 14.761290

C:\>
```