

Assignment_No.4

整数逆向输出

【问题描述】输入一个整数，将其逆向输出。要求定义并调用函数fun(n)，它的功能是返回n的逆向值。例如，fun(123)的返回值是321。main主函数调用子函数fun，并完成输入和输出功能。

【输入形式】输入一个整数。

【输出形式】将输入的数逆向输出。输出结束不换行。

【样例输入1】 123

【样例输出1】 321

【样例输入2】 -910

【样例输出2】 -19

解析

- 两种解决方案都是ok的
 - 使用字符串解决：不用计算位数直接使用各位数据
 - 使用整数进行计算：输出调用方便简洁，方便计算

示例代码1（字符串）

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <math.h>
5
6  int fun(int n){
7      char str[100];
8      sprintf(str, "%d", n);
9      n = 0;
10     if (str[0] == '-'){
11         for(int i = strlen(str) - 1; i > 0; i--){
12             n += (str[i] - '0') * pow(10, i-1);
13         }
14     }
15     else
16         for(int i = strlen(str) - 1; i >= 0; i--){
17             n += (str[i] - '0') * pow(10, i);
18         }
19     return n;
20 }
```

```

19  }
20
21  int main()
22  {
23      int n;
24      scanf("%d", &n);
25      printf("%d", fun(n));
26      return 0;
27  }

```

示例代码2(整数取余)

```

1  #include<stdio.h>
2  int fun(int num)
3  {
4      int n,s=0;
5      while(num!=0){
6          n = num%10;
7          s *= 10;
8          s += n;
9          num /= 10;
10     }
11     return s;
12 }
13 int main(){
14     int num;
15     scanf("%d",&num);
16     printf("%d",fun(num));
17     return 0;
18 }

```

整数合并

【问题描述】编写一个函数int comb(int a,int b)，将两个正整数a、b（取值范围为10~1000000）的十位数和个位数合并形成一个整数并返回。合并的方式是：将a的十位数和个位数依次放在结果的十位和千位上，将b的十位数和个位数依次放在结果的个位和百位上。例如，a=45，b=12，调用该函数后，返回5241。main主函数功能：输入两个整数，调用comb函数进行合并，并输出合并后的结果。

【输入形式】输入两个正整数。

【输出形式】输出合并后的正整数。输出结束不换行。

【样例输入1】 45 12

【样例输出1】 5241

【样例输入2】 10 223

【样例输出2】 312

【样例输入3】 123 4566

【样例输出3】 3626

解析

- 同上一题

示例代码1(字符串)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  char *comb(int a,int b){
6      char *p;
7      p=(char *)malloc(5*sizeof(char));
8      p[4]='\0';
9      p[0]=a%10+'0';
10     p[1]=b%10+'0';
11     p[2]=(a/10)%10+'0';
12     p[3]=(b/10)%10+'0';
13     if (p[0] == '0')
14         p++;
15     return p;
16 }
17
18 int main(){
19     int a,b;
20     scanf("%d%d",&a,&b);
21     printf("%s",comb(a,b));
22     return 0;
23 }
```

示例代码2(整数运算)

```
1  #include<stdio.h>
2
3  int comb(int a,int b){
4      return a%10*1000+a%100/10*10+b%100/10+b%10*100;
5  }
6
7  int main() {
8      int a,b;
```

```

9     int result;
10    scanf("%d%d",&a,&b);
11    result = comb(a,b);
12    printf("%d",result);
13    return 0;
14 }

```

绝对素数

【问题描述】

所谓“绝对素数”是指具有如下性质的素数：一个素数，将它的各位上的数逆序排列后形成的整数仍为素数，这样的数称为绝对素数。例如，11,79,389是素数，其各位上的数逆序排列后分别为11,97,983，仍为素数，因此这三个素数均为绝对素数。编写函数int absolute(int x)，判断x是否为绝对素数，如果x是，则返回1，否则返回0。main主函数功能：输入两个整数a和b，调用absolute函数，输出所有a和b之间（包括a和b）的绝对素数。

【输入形式】两个整数a和b

【输出形式】按从小到大的顺序依次输出，每行输出一个数值。输出结束有换行。

【样例输入】 80 120

【样例输出】

```

97
101
107
113

```

【样例说明】输入整数a=80，b=120，要求输出所有[80，120]之间的绝对素数。结果为：97,101,107,113，按升序分行输出。

示例代码

```

1  #include <stdio.h>
2  #include <math.h>
3
4  // 函数用来判断一个数是否为素数
5  int isPrime(int n) {
6      if (n <= 1)
7          return 0; // 0和1不是素数
8      if (n <= 3)
9          return 1; // 2和3是素数
10     if (n % 2 == 0 || n % 3 == 0)
11         return 0; // 能被2或3整除的不是素数
12     for (int i = 5; i * i <= n; i += 6) {
13         if (n % i == 0 || n % (i + 2) == 0)
14             return 0; // 能被i或i+2整除的不是素数

```

```

15     }
16     return 1; // 其他情况为素数
17 }
18
19 int absolute(int x) {
20     int original = x;
21     int reverse = 0;
22     while (x > 0) {
23         int digit = x % 10;
24         reverse = reverse * 10 + digit;
25         x /= 10;
26     }
27     return isPrime(original) && isPrime(reverse);
28 }
29
30 int main() {
31     int a, b;
32     scanf("%d%d", &a, &b);
33
34     for (int i = a; i <= b; i++) {
35         if (absolute(i))
36             printf("%d\n", i);
37     }
38     return 0;
39 }

```

求新数和倍数

【问题描述】输入正整数 n 和 $0 \sim 9$ 范围内的一个数 m ，判断 m 是否存在于 n 中（用函数实现），若不存在则输出“ m 不存在于 n 中”（ m 和 n 以具体输入的值代替）；若存在则删除 n 中的数字 m ，构成一个新数 k （高位为原高位，低位为原低位），并用原数 n 除以新数 k ，得到倍数（保留2位小数），然后依次输出新数 k 及其倍数，中间以一个逗号分隔。输出结束不
换行。

【样例输入1】 **12345 2**

【样例输出1】 **1345,9.18**

【样例说明1】 n 为12345， m 为2；删除2后的新数 k 为1345， n 是 k 的9.18倍。

【样例输入2】 **12045 0**

【样例输出2】 **1245,9.67**

【样例说明2】 n 为12045， m 为0；删除0后的新数 k 为1245， n 是 k 的9.67倍。

【样例输入3】 **12345 6**

【样例输出3】 **6不存在于12345中**

【样例说明3】 n 为12345， m 为6；输出“6不存在于12345中”。

示例代码1

```
1  #include <stdio.h>
2
3  // 函数用于判断数字m是否存在于数字n中
4  int isDigitInNumber(int n, int m) {
5      while (n > 0) {
6          if (n % 10 == m)
7              return 1; // 存在
8          n /= 10;
9      }
10     return 0; // 不存在
11 }
12
13 int main() {
14     int n, m;
15     scanf("%d%d", &n, &m);
16
17     if (isDigitInNumber(n, m)) {
18         // 删除数字m, 构成新数k
19         int k = 0, n_s = n;
20         int multiplier = 1;
21
22         while (n > 0) {
23             int digit = n % 10;
24             if (digit != m) {
25                 k += digit * multiplier;
26                 multiplier *= 10;
27             }
28             n /= 10;
29         }
30
31         float result = (float)n_s / (float)k;
32         printf("%d,%.2f\n", k, result);
33     } else
34         printf("%d不存在于%d中\n", m, n);
35
36     return 0;
37 }
```

合并字符串

【问题描述】编写一个函数void str_bin(char str1[], char str2[]), 其中str1、str2是两个有序字符串（字符已按ASCII码值从小到大排序），将str2合并到str1中，要求合并后的字符串仍是有序的，并且允许字符重复。main主函数功能：输入两个有序字符串，调用str_bin子函数完成字符串合并，并输出合并后的结果。

【输入形式】输入两个有序字符串（不超过100个字符）。

【输出形式】输出合并后的有序字符串。

【样例输入】

aceg

bdfh

【样例输出】 abcdefgh

【样例说明】输入两个有序字符串 "aceg" 和 "bdfh"，输出合并后的有序字符串 "abcdefgh"。输出结束不换行。

示例代码

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int strcom(const void *a, const void *b) {
6      return (*(char *)a - *(char *)b);
7  }
8
9  char *str_bin(const char* str1, const char* str2){
10     int len = strlen(str1) + strlen(str2);
11     char *p;
12     p = (char *)malloc(len);
13     memset(p, 0, len);
14     // 将str1和str2合并后放入p
15     strcat(p, str1);
16     strcat(p, str2);
17     // 将p中元素按照ascii码排序
18     qsort(p, len, sizeof(char), strcom);
19     return p;
20 }
21
22 int main(){
23     char str1[100], str2[100];
24     scanf("%s%s", str1, str2);
25     printf("%s", str_bin(str1, str2));
26     return 0;
27 }
```

组数

【问题描述】输入一行字符串（设字符数不大于80），提取该字符串中的数字字符并组成一个整数，输出该整数及其两倍的值。要求在主函数中输入字符串，并输出结果。在子函数中提取该字符串中的数字字符并组成一个整数。

【输入形式】输入一行字符串。

【输出形式】提取该字符串中的数字字符并组成一个整数，输出该整数及其两倍的值。

【样例输入】 **Enter a string:ab34df6**

【样例输出】 **digit=346,692**

【样例说明】输入和输出提示符中冒号、逗号、等号均为英文符号，两边均没有空格。英文字母区分大小写。必须严格按样例输入打印。

输出结束不换行。

示例代码

```
1  #include <stdio.h>
2
3  int TurntoInt(const char *str) {
4      int result = 0;
5      for (int i = 0; str[i] != '\0'; i++) {
6          if (str[i] >= '0' && str[i] <= '9')
7              result = result * 10 + (str[i] - '0');
8      }
9
10     return result;
11 }
12
13 int main() {
14     char str[81];
15     printf("Enter a string:");
16     scanf("%s", str);
17     int num = TurntoInt(str);
18     int doubledValue = num * 2;
19     printf("digit=%d,%d\n", num, doubledValue);
20     return 0;
21 }
22
```