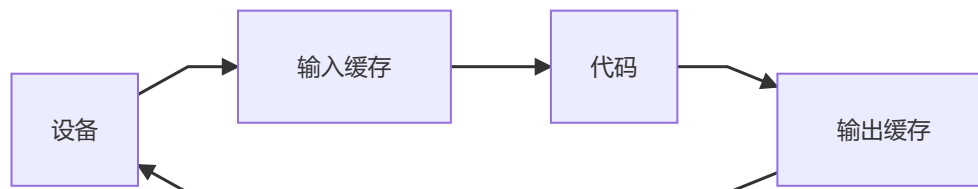


基础作业

1 先放入标准输入缓冲区的内容是先被读取还是后被读取？

- 先放入的先读取 直到缓冲区溢出或者没有数据为止



2 scanf的%d和%c对于标准输入缓冲区当中的空白有什么区别？

- %c会读取缓冲区内所有的字符数据数据包括：空白字符 && 换行符。
- %d读入可选有符号\无符号的十进制整数。不会读入空白字符。

3 scanf函数的返回值是什么类型？它的具体含义是什么？

- 返回值为int类型，表示成功读入的数据项数。如果出错则会返回EOF。

4 putchar和getchar等价于什么形式的scanf和printf？

- putchar：相当于只输出一个字符的printf。

如printf("%c",'A') 相当于 putchar('A')

- getchar：相当于只读入一个字符的scanf，但是在连续读入的时候会读取缓冲区里的空白字符和换行符。

5 如何使用关系表达式判断某一个年是不是闰年？（不能使用括号表达式）

- `year%4 == 0 && year %100 != 0 || year % 400 ==0`

6 整数左移一位相当于做了什么运算？右移一位呢？（注意区分正负、奇偶）

- 对于左移而言：相当于进行了乘以2的运算
- 对于右移而言：正数补0，负数补1。相当于正数做了除2的运算，负数做了减一再除2的运算。

7 无符号整型和有符号数的右移有区别吗？//看完再议

- 对于有符号整数而言（有符号整数有整数之分）
 - 高位补0，低位丢弃
 - 正数：除二
 - 负数：减一之后再除二
- 无符号右移时，高位补最高有效位的值，低位丢弃。

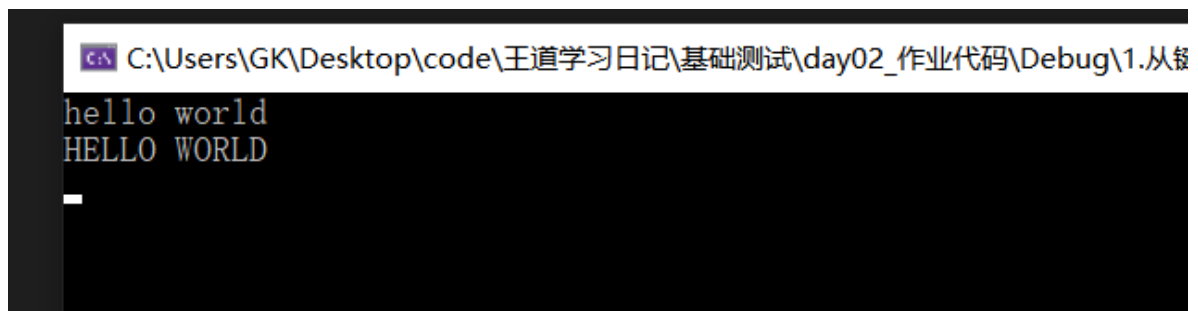
提高作业

1.从键盘上输入字符，将小写字母转换成大写字母。输入“ctl + z”结束

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>      //不知道为啥按回车就直接结束了
3  #include<cstdio>        //没给我输入ctrl + z的机会
4  using namespace std;   //可能这就是vs2019吧
5  int main() {
6      char c;
7      int ret;
8      while (scanf_s("%c",&c)!=EOF){
9          if (c != '\n') {
10             printf("%c", c - 32);
11         }
12         else {
13             printf("\n");
14         }
15     }
16 }

```



2 从键盘上输入字符

a 分别统计一下其中字母，数字，其他字符的个数

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<cstdio>
4  using namespace std;
5  int main() {
6      char c;
7      int i;
8      int alp = 0, num = 0, oth = 0;
9      while (scanf_s("%c", &c) != EOF) {
10         if (c != '\n') {
11             i = (int)c; //将输入的字符强制转换成整数
12             if (i >= 65 && i <= 90) { //大写字母
13                 alp++;
14             }
15             else if (i >= 97 && i <= 122) { //小写字母
16                 alp++;
17             }
18             else if (i >= 48 && i <= 57) { //数字
19                 num++;
20             }
21             else { //其他
22                 oth++;
23             }
24         }
25         else {
26             printf("\n");
27         }
28     }
29 }

```

```

28     }
29     cout << "alp = " << alp << endl;
30     cout << "num = " << num << endl;
31     cout << "oth = " << oth << endl;
32 }

```

b 将统计的字母，数字，其他字符的个数以柱状图的形式打印（按照数量关系排序）

```

1 //实在想不出来排序之后应该如何打印下面对应的字符

```

3 进制转换

- a 将10进制数转换成2进制数（使用字符数组存储）

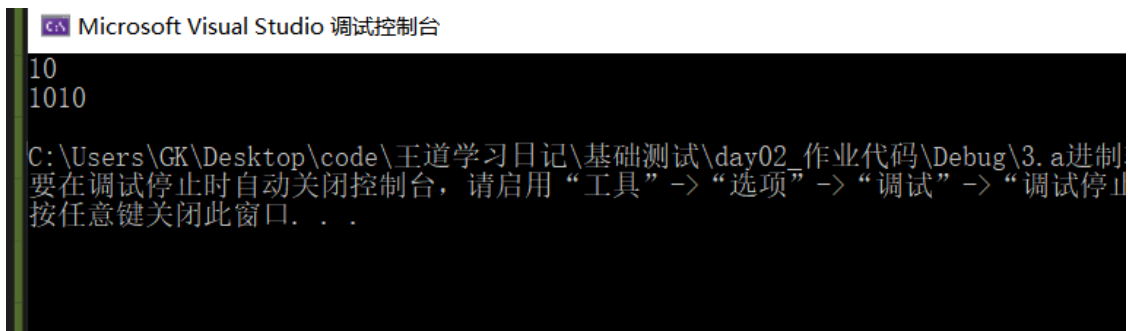
```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<cstdio>
4  #include<vector>
5  using namespace std;
6  int main() {
7      int num, count = 0;
8      int a[1024];
9      scanf("%d", &num);
10     while (num>0){
11         a[count] = (num % 2); //除二取余
12         num /= 2;
13         count++;
14     }
15     for (count; count >= 0; count--) {
16         printf("%d", a[count]);
17     }
18     cout << endl;
19     return 0;
20 }

```

- b 将2进制数（使用字符数组存储）转换成10进制数（使用整型存储）

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<cstdio>
4  #include<vector>
5  using namespace std;
6  int main() {
7      int num = 0, count = 0;
8      int c;
9      int a[100];
10     scanf("%d", &num);
11     while (num > 0) {
12         c = (num % 2);
13         a[count] = c; //除二取余
14         num /= 2;
15         count++;
16     }
17     for (count--; count >= 0; count--) { //倒序输出
18         cout << a[count];
19     }
20     cout << endl;
21     return 0;
22 }
```



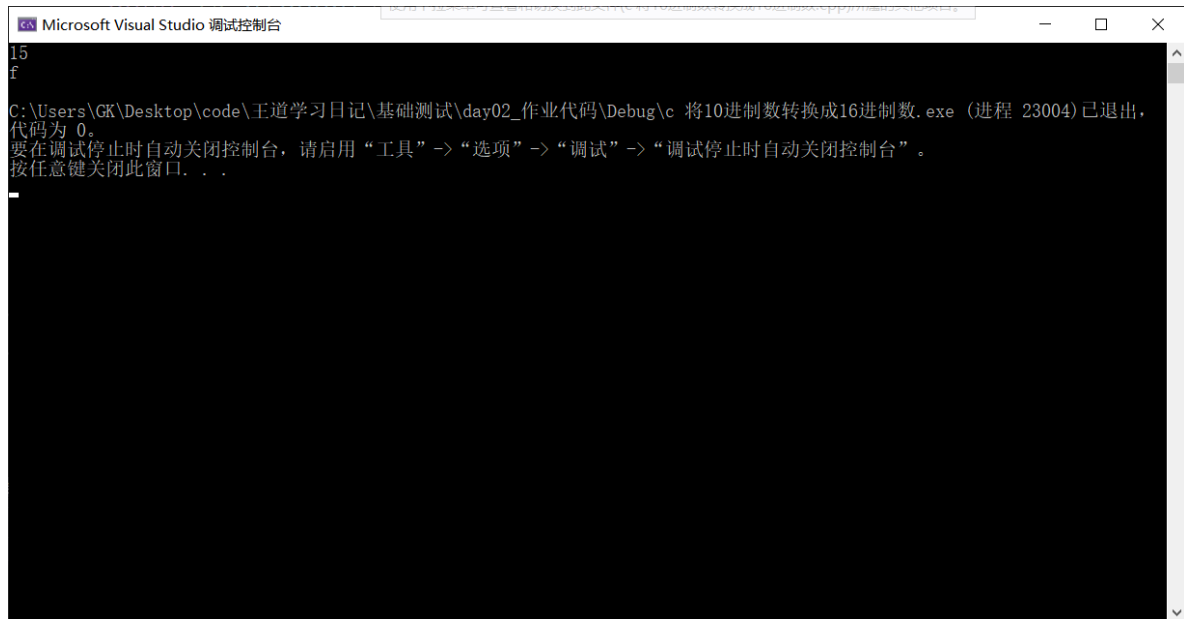
- c 将10进制数转换成16进制数（使用字符数组存储）

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<cstdio>
4  #include<vector>
5  using namespace std;
6  int main() {
7      int num = 0, count = 0;
8      int c;
9      char a[100];
10     scanf_s("%d", &num);
11     while (num > 0) { //除以16取余
12         c = (num % 16);
13         if(c > 9) {
14             a[count] = c + 'A' - 10;
15         }
16         else {
17             a[count] = c + '0';
18         }
19         num /= 16;
20         count++;
21     }
```

```

21     }
22     for (count--; count >= 0; count--) { //倒序输出
23         cout << a[count];
24     }
25     cout << endl;
26     return 0;
27 }

```



d 将16进制数（使用字符数组存储）转换成10进制数（使用整型存储）

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<cstdio>
4  #include<cmath>
5  using namespace std;
6  int main() {
7      int c;
8      long int num = 0;
9      char a[100] ;
10     scanf("%s", &a);
11     int l = strlen(a);
12     for (int i = 0; i < l; i++) { //遍历数组
13         if (a[i] > '0' && a[i] <= '9') {
14             c = a[i] - '0';
15         }
16         else if(a[i] >= 'a' && a[i] <= 'f'){
17             c = a[i] - 'a' + 10;
18         }
19         else {
20             c = a[i] - 'A' + 10;
21         }
22         num += pow(16, c);
23     }
24     cout << num << endl;
25     return 0;
26 }

```

4.统计一个整型数字的二进制形式中有多少个1（注意正负数都要计算）

```
1  #include<iostream>      //开始是想负数二进制应该怎么判断
2  #include<cstdio>        //想了半天没想出结果
3  using namespace std;    //还是直接写了个转换
4  void Count(int i);
5  int main() {
6      int i;
7      while (scanf_s("%d", &i) != EOF) {
8          Count(i);
9      }
10 }
11 void Count(int i) {
12     int count = 0;
13     while (i) {          //按位取 由于是二进制 所以各个位数的和就是1的个数
14         count += i & 0x01;
15         i >>= 1;
16     }
17     printf("%d", count);
18 }
```

5.输入年月日，输出该日期的下一天的年月日

```
1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  int date_designate(int year, int month, int day); //判断除开二月的所有
5  int main() {
6      int year, month, day;
7      cin >> year >> month >> day;
8      if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) { //优先判断闰年
9          if (month == 2 && day == 29) { //仅有二月份是不一样的 其他年份和别的一样
10             month++;
11             day = 1;
12             cout << year << "年" << month << "月" << day << "日";
13         }
14         else { //闰年的其他月份和非闰年一样
15             date_designate(year, month, day);
16         }
17     }
18     else { //非闰年
19         if (month == 2 && day == 28) { //函数中未写判断二月 单独判断
20             month++;
21             day = 1;
22             cout << year << "年" << month << "月" << day << "日";
23         }
24         else {
25             date_designate(year, month, day);
26         }
27     }
28 }
29 int date_designate(int year, int month, int day) {
30     if (month == 12 && day == 31) { //需要年++的情况
31         year++;
32     }
```

```

32     month = 1;
33     day = 1;
34 }
35 else if (month == 1 || 3 || 5 || 7 || 8 || 10 && day == 31) { //大月中需要
月++的情况
36     month++;
37     day = 1;
38 }
39 else if (month == 4 || 6 || 9 || 11 && day == 30) { //小月中需要月++的情况
40     month++;
41     day = 1;
42 }
43 else { //剩下的所有情况
44     day++;
45 }
46 cout << year << "年" << month << "月" << day << "日"; //最后输出年月日
47 return 0;
48 }

```

Microsoft Visual Studio 调试控制台

```

2020 2 29
2020年3月1日
C:\Users\GK\Desktop\code\王道学习日记\基础测试\day01
992) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项
按任意键关闭此窗口. . .

```