

【注意:】

- 1、每题均有知识点使用限制, 认真阅读, 注意合规
- 2、不考虑输入错误
- 3、首行及格式缩进要求同之前作业 (首行-10%, 缩进-10%)
- 4、要求 Dev 下 “0 errors, 0 warnings”
- 5、不要忘记首行信息

【本作业特别要求:】

- 1、所有题目都必须用指针 (即使出现数组, 也必须是指针法访问)
- 2、一维数组, 除定义变量外, 函数的形参及程序的执行语句不允许出现 [] 形式
- 3、指针在一维数组的元素间移动时, 要体现出指针变量自身的变化
 - 整个函数, 不能直接用 *(数组名+int)/*(指针+int) 来规避不允许出现 [] 形式的要求, 至少要有指针的++/--
 - 参考 060001 课件, 要求为类似 P. 61 的左侧形式, 如果写成右上或右下则得分为 0!!!

§ 6. 指针基础



6.4. 字符串与指针

6.4.4. 下标法与指针法处理字符串

例: 将str1的内容复制给str2

```
int main()
{ char str1[]="I Love CHINA!", str2[20];
  char *p1, *p2;
  p1=str1;
  p2=str2;
  for (; *p1!='\0'; p1++, p2++)
    *p2=*p1;
  *p2='\0';
  p1=str1;
  p2=str2;
  cout << "str1:" << p1 << endl
        << "str2:" << p2 << endl;
  return 0;
}
```



例: 将str1的内容复制给str2

```
int main()
{ char str1[]="I Love CHINA!", str2[20];
  int i;
  for (i=0; str1[i]!='\0'; i++)
    str2[i] = str1[i];
  str2[i]='\0';
  cout << str1 << endl;
  cout << str2 << endl;
  return 0;
}
```



数组法实现

例: 将str1的内容复制给str2

```
int main()
{ char str1[]="I Love CHINA!", str2[20];
  int i;
  for (i=0; *(str1+i]!='\0'; i++)
    *(str2+i) = *(str1+i);
  *(str2+i]='\0';
  cout << str1 << endl;
  cout << str2 << endl;
  return 0;
}
```



等价指针法

1、一维数组与指针

题目描述：键盘输入若干正整数到一个数组中（个数不定，以-1 结束，超过 2000 个数据则只取前 2000 个），将其调整为奇数在前，偶数在后。

输入输出格式要求：多行

Line1：输入提示：“请输入数组各元素(正整数),以-1 结束(不含-1 最多 2000 个数)”

Line2：键盘输入的若干数字（-1 结束，**不考虑输入错误**，允许加回车的多行形式）

Line3：输出提示：“数组的内容为”

Line4-x：若干行输出，每行 15 个元素

Linex+1：输出提示：“调整后的数组内容为”

Linex+2：若干行输出，每行 10 个元素

```
Microsoft Visual Studio 调试控制台
请输入数组各元素(正整数),以-1结束(不含-1最多2000个数)
3 103 123 97 101 87 72 88 74 61 65 2 30 17 19 16 9 9 5 43 2 9 -1
数组的内容为
3 103 123 97 101 87 72 88 74 61 65 2 30 17 19
16 9 9 5 43 2 9
调整后数组的内容为
3 103 123 97 101 87 9 43 5 61
65 9 9 17 19 16 30 2 74 88
2 72
```

① **不要忘记首行信息，检查是否正确的方法见⑥，不是 txt_compare**

② 本题允许使用的知识点：一维数组与指针（含之前的程序基本结构、函数、数组等知识）

③ 给出 w14-c3-s1-demo.exe 供参考，如果题目要求、截图、demo.exe 不一致，以 demo 为准

④ 给出基准程序 w14-c3-s1.cpp，按基准程序的限制要求完成即可（**违规 0 分**）

⑤ 生成测试数据的方法：w14-c3-s1-demo.exe --makedat 数量[1..2100]，会生成 data.txt（注意：makedat 前是两个减号）

⑥ 检查是否正确的方法（**不是 txt_compare**）：

w14-c3-s1.exe(你的) <data.txt >myout.txt（从 data.txt 中读，输出写入 myout.txt 中）
w14-c3-s1-demo.exe --check myout.txt（两个减号，myout.txt 为输出重定向文件）

```
D:\VS-Debug>w14-c3-s1-demo.exe --makedat 1698
已生成文件data.txt，共1698个有效数字.

D:\VS-Debug>w14-c3-s1-demo.exe < data.txt > myout.txt

D:\VS-Debug>w14-c3-s1-demo.exe --check myout.txt
检查原始数组：通过
检查调整后数组：通过
检查结束
```

生成数据和检查都用demo
红框内替换为你的exe

注：必测有效数据超过 2000 的情况（即--makedat 2001~2100 等情况下要正确）

2、一维数组与指针

题目描述：键盘输入 n 个整数，放入一个数组中，再输入 m 值，表示将每个数依次后移 m 个位置，后面的数则回绕到前面（ m 可能远远大于 n 的数量）

输出格式要求：多行

Line1: 输入提示：“请输入整数的数量 $n[10-20]$ ”

Line2: 键盘输入的 n 值（不考虑输入错误）

Line3: 输入提示：“请输入 xx 个整数”（ xx 为 Line2 的键盘输入）

Line4: 键盘输入的 n 个数（允许用加回车的多行形式输入，不考虑输入错误）

Line5: 输出提示：“输入的整数序列为”

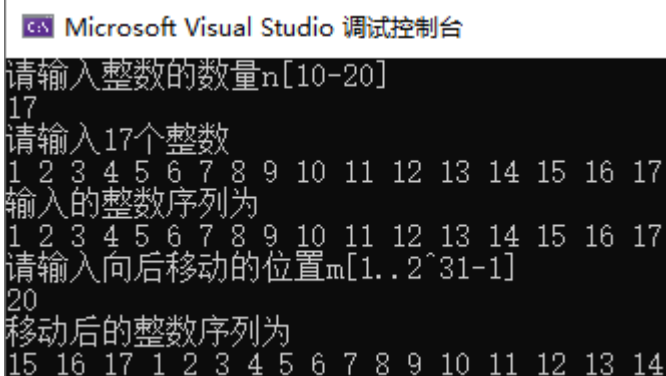
Line6: 输入数据的原始序列，空格分隔（最后有无空格均可）

Line7: 输入提示：“请输入向后移动的位置 $m[1..2^{31}-1]$ ”

Line8: 键盘输入的 m 值（不考虑输入错误，可能远远大于 n 值）

Line9: 输出提示：“移动后的整数序列为”

Line10: 调整后的序列，空格分隔（最后有无空格均可）



```
Microsoft Visual Studio 调试控制台
请输入整数的数量n[10-20]
17
请输入17个整数
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
输入的整数序列为
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
请输入向后移动的位置m[1..2^31-1]
20
移动后的整数序列为
15 16 17 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

- ① 不要忘记首行信息，检查是否正确的方法为 `txt_compare --trim right` 与 demo 做到一致
- ② 本题允许使用的知识点：一维字符数组与指针（含之前的程序基本结构、函数、数组等知识）
- ③ 给出 `w14-c3-s2-demo.exe` 供参考，如果题目要求、截图、`demo.exe` 不一致，以 `demo` 为准
- ④ 给出 `w14-c3-s2.c` 基准程序，按基准程序的限制要求完成即可（违规 0 分）

【编译器要求：】

| 源程序文件名 | 题目说明 | 编译器VS | 编译器Dev |
|---------------|-------------|-------|--------|
| w14-c3-s1.cpp | 数组元素调整 | / | Y |
| w14-c3-s2.c | 数组循环移动(C方式) | / | Y |

【提交要求：】

- 1、5月31日 17:04:59 前网上提交本次作业
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明

【重要提醒：】

实验课作业考察快速完成度，提交开放后，**先将**已完成的题目提交，**不要**最后集中提交，避免各种问题导致本次分数全部丢失。