

清华大学本科生考试试题专用纸

考试课程 计算机语言与程序设计 (卷) 2020 年 11 月 15 日

姓名 请填写姓名 班级 请填写班级 学号 请填写学号

大题	题号	一	二	三	四	总分
	分数	30 分	30 分	30 分	10 分	
成绩						

试卷提交说明

1. 在考卷上填写你的姓名、学号；
2. 所有程序，都只能使用 C89/C99 标准包含的头文件；并且禁止使用任何全局变量；
3. 按要求编程，调试运行，将每道题的程序进行复制，将运行结果进行截图，粘贴在考卷对应的题后。答卷中的程序和截图仅做备用，以提交的.c 文件为准；
4. 试卷中示例运行结果的提示语、空行仅做参考，不参与评分，不必在编程中输出与示例完全一致的提示语。只有有实际作用的输入、输出、换行等数据参与评分。
5. 无特殊要求的函数，形参、输出均可自拟；
6. 交卷时，提交一个压缩文件包，包括：
 - a) 文件名“学号.zip”，如 2019123456.zip
 - b) ZIP 文件包内容
 - ◇ 每道题的 c 语言源文件，文件名为“学号_题号.c”，如 2019123456_1.c；
 - ◇ 考卷，文件名为“学号_考卷.docx”
 - ◇ 例如：2019123456.zip 应包括 2019123456_1.c、2019123456_2.c、2019123456_3.c、2019123456_4.c、2019123456_考卷.docx，共 5 个文件。

一、字符串处理 (30 分)

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型。本题的研究对象是一个包含 4 个个体的种群，每个个体可以用一个长度为 n 的“染色体”来表示，“染色体”是指由 AGCT 四种字符构成的字符串，例如“AAGTCCA”就是一条“染色体”。请按照要求编写程序，程序结构如下：

1. 函数 Input() (7 分)

该函数的功能是从键盘输入 1 个整数 n 作为染色体的长度；4 个字符串 str1, str2, str3, str4 作为 4 个染色体；1 个整数 g 作为交叉点。该函数还能将上述输入的内容传回主调函数。假设 str1~str4 的长度不超过 n，且 n 最大不超过 200。该函数具体的要求为：(1) str1~str4 只能包含 AGCT 四种字符 (2) g 不能超过 n。若某个字符串不符合要求 (1)，则提示重新输入它，若整数 g 不符合要求 (2)，则提示重新输入它。

2. 函数 Complement() (5 分)

通过参数接收主函数提供的字符串 str，整数 n，实现对字符串的补全操作，得到新字符串 str_comp，并能将它返回给主调函数。补全操作的要求是在 str 末尾加入一个或多个 str 的逆序串，直到它的长度大于等于 n，并截取前 n 个字符作为新字符串 str_comp。

禁止使用 strrev 库函数。

如字符串 str, n 分别是“AGTAATC”, 10, 则得到新字符串 str_comp 为“AGTAATCCTA”。
如字符串 str, n 分别是“AGGT”, 10, 则得到新字符串 str_comp 为“AGGTGGATG”。

3. 函数 PickTwo() (10 分)

通过参数接收主函数提供的字符串 str1_comp, str2_comp, str3_comp, str4_comp，实现对 4 个字符串的选择操作。选择操作的要求是，首先依次统计 4 个字符串中连续子串数量，作为该字符串的分数，接着将分数最高的前两个字符串 str_n1, str_n2 返回

给主调函数。如果两个字符串分数相同，则按照他们在 Input 函数中输入的次序排序。如 str1_comp~str4_comp 分别是“AGTAATCCTA”，“CGGGTACAGC”，“TATAGCCATT”，“AGGGGCTTTT”，则 str_n1,str_n2 为“AGTAATCCTA”，“TATAGCCATT”。

4. 函数 Evolve() (4 分)

通过参数接收主函数提供的字符串 str_n1,str_n2 和整数 g，实现对 str_n1,str_n2 的进化操作，并能将进化后的字符串 str_new_1,str_new_2 返回给主调函数。进化操作的要求是，将 str_n1,str_n2 在位置 g 处及之后的部分进行交叉，得到 str_new_1, str_new_2。首字符的位置为 0。

如 str_n1,str_n2,g 分别是“AGTTTAT”,“GTACCCA”，“3”则 str_new_1, str_new_2 为“AGTCCCA”,“GTATTAT”。

如 str_n1,str_n2,g 分别是“AGTTTAT”,“GTACCCA”，“0”则 str_new_1, str_new_2 为“GTACCCA”,“AGTTTAT”。

5. 主函数 main() (4 分)

- (1) 定义所有使用到的字符串及各变量。
- (2) 调用 Input()函数，完成输入操作。
- (3) 调用 Complement ()函数，依次完成 str1, str2, str3, str4 的补全；在主函数中依次输出 str1_comp, str2_comp, str3_comp, str4_comp。
- (4) 调用 PickTwo()函数，完成 str1_comp, str2_comp, str3_comp, str4_comp 的选择操作；在主函数依次输出 str_n1,str_n2。
- (5) 调用 Envolve()函数，完成 str_n1,str_n2 的进化操作；在主函数依次输出 str_new_1, str_new_2。

[程序示例]

【Input】

```
20
AGGTCCGGGGTAAA
TGGGAATCGAGCGA
TTTGGGGAAAAGG
ATATATGCGCGC
15
```

【Output】

```
AGGTCCGGGGTAAAAAATGG
TGGGAATCGAGCGAAGCGAG
TTTGGGGAAAAGGGGAAAAG
ATATATGCGCGCCGCGCGTA
AGGTCCGGGGTAAAAAATGG
TTTGGGGAAAAGGGGAAAAG
AGGTCCGGGGTAAAAAAAAG
TTTGGGGAAAAGGGGAATGG
```

【Run】

```

Please input n
20
Please input str1
AGGTCCGGGGTAAA
Please input str2
TGGGAATCGAGCGA
Please input str3
TTTGGGGAAAAGG
Please input str4
ATATATGCGCGC
Please input g
15
str1_comp:
AGGTCCGGGGTAAAAATGG
str2_comp:
TGGGAATCGAGCGAAGCGAG
str3_comp:
TTTGGGGAAAAGGGGAAAAG
str4_comp:
ATATATGCGCGCCGCGCGTA
str_n1:
AGGTCCGGGGTAAAAATGG
str_n2:
TTTGGGGAAAAGGGGAAAAG
str_new_1:
AGGTCCGGGGTAAAAAAAAG
str_new_2:
TTTGGGGAAAAGGGGAATGG
请按任意键继续. . .

```

请在此处粘贴程序和运行结果截图

批阅要点:

1.函数 Input() (7 分)

- 1) 对输入的 4 个字符串进行合法性判断 3 分
- 2) 对输入的整数 g 进行合法性判断 2 分
- 3) 传回输入的 4 个字符串和整数 g 2 分

2.函数 Complement () (5 分)

- 1) 实现字符串的逆序补全 3 分
- 2) 补全后的字符串长度等于 n 1 分
- 3) 传回 4 个补全的字符串 1 分

3.函数 PickTwo () (10 分)

- 1) 统计连续子串数量 4 分
- 2) 对字符串按分数合理排序 3 分
- 3) 找到正确的 str_n1 与 str_n2 2 分
- 4) 传回 str_n1 与 str_n2 1 分

4.函数 Evolve () (4 分)

- 1) 实现字符串的交换 1 分
- 2) 在整数 g=0 时正确交换字符串 1 分
- 3) 在整数 g=n 时正确交换字符串 1 分
- 4) 传回 str_new_1 与 str_new_2 1 分

5.主函数 main () (4 分)

- 1) 调用 Input 函数, 完成输入操作 1 分
- 2) 调用 Complement 函数, 输出 str1_comp, str2_comp, str3_comp, str4_comp 1 分

- 3) 调用 PickTwo 函数, 输出 str1_n1, str2_n2; 1 分
4) 调用 Evolve 函数, 输出 str_new_1, str_new_2; 1 分

二、数组处理 (30 分)

单分子定位显微术(SMLM)是一类超分辨率显微技术, 通常用于在三维空间 (x,y,z) 中对用特定波长激发的荧光分子进行精确的成像。假设用一个三维矩阵代表待成像的三维空间, 每次激发的荧光分子位置是不确定的, 且每个激发的荧光分子亮度彼此无关, 范围在 1~255 之间。没有荧光分子的位置亮度为 0。有如下定义: 对任意维度的矩阵, 若数值为 0 的元素数目远远多于非 0 元素的数目, 并且非 0 元素分布没有规律时, 则称该矩阵为稀疏矩阵; 与之相反, 若非 0 元素数目占大多数时, 则称该矩阵为稠密矩阵。则单分子定位的测量结果为典型的三维稀疏矩阵

请编制一个程序, 实现如下功能:

1. 函数 InputMatrix() (8 分)

该函数内, 用户从键盘输入 3 个大于 0 的整数 m、n、p, 用空格分开。根据输入的整数值动态生成一个三维整型矩阵 A (m*n*p), 代表进行单分子定位测量的三维空间。将上述输入的内容和生成的三维矩阵 A 传回主函数。假设用户输入的 m、n、p 是符合要求的, 该函数不用判断。

2. 函数 InputResult() (5 分)

该函数内, 用户从键盘输入多行, 每行包含 4 个非负整数 x、y、z、g, 要求 $0 \leq x < m$, $0 \leq y < n$, $0 \leq z < p$, $1 \leq g < 256$, 代表检测到在坐标 (x,y,z) 点处检测到亮度为 g 的荧光分子, 假设至少输入一行有效数据。不同行数据保证坐标不会重复。若输入的 x、y、z 值不符合上述的区间, 则代表输入完成。将含有全部有效数据信息的矩阵 A 传回主函数。

3. 函数 Sparseness() (6 分)

对一个稀疏矩阵, Hoyer 提出可以利用 L1 范数和 L2 范数之间的差异度来衡量其稀疏度 (sparseness)。它是一个 [0,1] 之间的数, 值越大, 说明矩阵越稀疏。对于一个非负矩阵 A, 计算其稀疏度, 可分为如下三步:

- (1) 计算 A 所包含的所有元素个数 t;
- (2) 计算 A 的 L1 范数 (所有元素之和):

$$\mathcal{L}_1(A) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{p-1} A(i,j,k)$$

- (3) 计算 A 的 L2 范数 (所有元素的平方之和的平方根):

$$\mathcal{L}_2(A) = \sqrt{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{p-1} (A(i,j,k))^2}$$

则稀疏度可表示为:

$$\text{sparseness}(A) = \frac{\sqrt{t} - \mathcal{L}_1(A)/\mathcal{L}_2(A)}{\sqrt{t} - 1}$$

该函数内, 计算三维矩阵 A 的稀疏度, 并将结果传回主函数。

4. 函数 Centroid() (6 分)

对于一个多维物体, 其质心指物质系统上被认为质量集中于此的一个假想点。以亮度作为权重, 三维矩阵的质心坐标计算公式为

$$x = \frac{\sum_{i=0}^{m-1} [(\sum_{j=0}^{n-1} \sum_{k=0}^{p-1} A(i,j,k)) * i]}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{p-1} A(i,j,k)}$$

$$y = \frac{\sum_{j=0}^{n-1} [(\sum_{i=0}^{m-1} \sum_{k=0}^{p-1} A(i,j,k)) * j]}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{p-1} A(i,j,k)}$$

$$z = \frac{\sum_{k=0}^{p-1} [(\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A(i, j, k)) * k]}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{k=0}^{p-1} A(i, j, k)}$$

其中 $A(i, j, k)$ 为矩阵 A 在 (i, j, k) 坐标处的值。该函数内，计算三维矩阵 A 的质心坐标，并将结果传回主函数。

5. 主函数 main() (5分)

- (1) 定义所使用到的变量。
- (2) 依次调用 InputMatrix()、InputResult()函数完成输入，从而完成对矩阵 A 的定义和赋值。
- (3) 调用 Sparseness()函数，完成矩阵 A 的稀疏度计算，并在主函数中输出稀疏度，结果四舍五入保留四位小数。
- (4) 调用 Centroid()函数，完成矩阵 A 的质心坐标计算，并在主函数中输出三维坐标，结果四舍五入保留两位小数。三个坐标值用空格隔开。

[程序示例]

【Input】

```
3 2 4
2 0 0 208
2 1 1 78
0 1 2 113
0 1 1 157
2 1 3 231
2 0 3 194
0 0 3 91
1 1 3 240
1 0 1 65
3 2 4 2
```

【Output】

```
0.5471
1.25 0.59 2.03
```

【Run】

```
3 2 4
2 0 0 208
2 1 1 78
0 1 2 113
0 1 1 157
2 1 3 231
2 0 3 194
0 0 3 91
1 1 3 240
1 0 1 65
3 2 4 2
0.5471
1.25 0.59 2.03
```

请在此处粘贴程序和运行结果截图

批阅要点：

1、 InputMatrix()函数 (8)

- a) 函数头及形参表列意义正确；1分
- b) 在函数中输入的 3 个大于 0 的整数 m、n、p 正确；1分
- c) 根据输入的 m、n、p 动态生成的矩阵正确（一层 1 分）；3分
- d) 必须能够正确传回输入的矩阵，即相关的多级指针赋值正确；3分

2、 InputResult ()函数 (5)

- a) 多行输入正确；1 分
 - b) 输入终止条件正确；1 分
 - c) 对 A 的全部元素均赋值正确；2 分
 - d) 参数传递正确；1 分
- 3、Sparseness ()函数（6）
- a) 函数头及形参表列意义正确；1 分
 - b) 范数数值类型、求值正确；3 分
 - c) 求解稀疏度正确；2 分
- 4、Centroid ()函数（6）
- a) 函数头及形参表列意义正确；1 分
 - b) 质心数值类型、求值正确；4 分
 - c) 返回值正确；1 分
- 5、main()函数（5）
- a) 在主函数中能够正确定义多级指针来代表矩阵，及各个参数在传递时的指针；2 分
 - b) 函数调用正确（与定义的函数均能对应）；1 分
 - c) 输出稀疏度，精度和数值均正确；1 分
 - d) 输出质心，精度和数值均正确；1 分

三、链表/结构体编程（30 分）

随着基因测序技术的发展，现在世界上的很多国家都建立起了自己的病毒基因库，从而为疾病的筛查治疗提供数据支持。在当下新型冠状病毒肆虐的背景下，建立全球共享的病毒基因库对检测病毒变异，控制疫情蔓延更是具有不可替代的作用。下面，请你根据要求，利用链表或者结构体数组（结构体指针数组亦可，下同）模拟建立一个简单的病毒基因数据库，并完成相关操作。

1. 定义节点类型 virus_gene（3 分）

每个节点主要包含以下信息：

- （1）病毒编号 number: 字符串，包含 4 位数值字符，如“1001”，不同病毒编号唯一；
- （2）病毒名称 name: 字符串，不含空格，长度不超过 20，如“covid”，不同病毒名称唯一；
- （3）特征序列 sequence: 字符串，由“A,T,G,C”四种字符组成，长度不超过 100，如“ATCGGCATC”。

2. 函数 Create()（5 分）

功能：

- （1）从键盘依次输入若干个病毒的信息（包含病毒编号、病毒名称、特征序列）。按照键盘输入的顺序，构造一个链表/结构体数组。
- （2）若输入“STOP”，则表示信息输入结束。
- （3）回传链表/结构体数组给主调函数。

说明：输入的病毒数目不超过 20；假设输入的信息均满足要求，不用判断非法输入

3. 函数 PrintInfo()（3 分）

功能：该函数接收一个链表/结构体数组作为参数，然后将其内容输出到屏幕上。

4. 函数 InsertInfo()（5 分）

功能：

- （1）该函数接收两个链表/结构体数组*list1 和*list2 作为参数，并从键盘读入一个整数 n 作为输入。
- （2）然后把第二个链表/结构体数组*list2 插入合并到第一个链表/结构体数组*list1 的第 n 个节点之后得到新链表/结构体数组，最后回传给主调函数。

说明：节点序号从 1 开始计数，即 n=1 时，是将*list2 插入到*list1 的首节点之后。特殊的，当 n=0 时，代表将*list2 整个插入到*list1 之前。**注意，需要判断 n 是否越界**

($n < 0$ 或者大于待插入链表/结构体数组长度均为越界), 越界则提示重新输入。

5. 函数 SortInfo() (4 分)

功能: 该函数接收一个链表/结构体数组作为参数, 然后将该链表/结构体数组按病毒编号的升序进行排序, 并回传新链表/结构体数组给主调函数。

6. 函数 SeqLoc() (6 分)

功能:

- (1) 该函数接收一个链表/结构体数组 *list 作为参数, 然后循环地从键盘中读入一个病毒名称 virus_name 和一个特征子序列 str (不超过 10 个字符) 作为输入。
- (2) 得到输入后, 该函数在链表/结构体数组中搜索该病毒名称, 若搜索到该病毒, 则在该病毒的特征序列中定位并输出所给的特征子序列 str 出现的**所有位置 (位置从 0 开始计数, 输出多个位置时用空格隔开)**; 如果没有该病毒名称则输出: NoVirus; 如果该病毒中不存在该特征子序列则输出: NotFound。
- (3) 当输入 Ctrl-Z, 回车 时截止。

说明:

- (1) 允许匹配到的子串之间有重叠。示例: 病毒的特征序列为“AT**CGG**AT**CGG**CTCG”, 待定位的特征子序列为 “CGG”, 则要输出的定位到的位置为“2 7”; 病毒的特征序列为 “AT**CGGG**CGGCTCG”, 待定位的特征子序列为 “GG”, 则要输出的定位到的位置为“3 4 5”;
- (2) **禁止使用 strstr() 函数。**

7. 主函数 main() (4 分)

- (1) 两次调用 Create(), 生成两个链表/结构体数组 *list1 和 *list2。
- (2) 调用 InsertInfo (), 从键盘读入插入位置 n, 并将 *list2 插入到 *list1 的第 n 个节点之后; 然后调用 PrintInfo() 显示插入合并后的链表/结构体数组。
- (3) 以合并后的链表/结构体数组作为实参, 调用 SortInfo(), 对合并后的链表/结构体数组进行排序, 然后调用 PrintInfo() 显示排序后的链表/结构体数组。
- (4) 以排序后的链表/结构体数组作为实参, 调用 SeqLoc(), 键盘多次输入待查询的病毒名称和需要定位的特征子序列, 输出基因序列的定位结果。
- (5) 若有动态分配的内存空间, 需要手动进行释放。

[程序示例]

【Input】

```
1232 covid1 AGTCGACGAGCTCGGACT
1212 hivirus TCTACTCATTGACCACTGA
1244 mkvir AGGCGCATCGCGCTTCC
1256 witvir ATCGGTCAATTAACG
1135 aidsvir CGGTATCGACGGATCCAGT
STOP
1776 fatrvir TGCATCGACCGTACGCG
1555 sfavir TGCATGCTAGCT
1237 jpkvir TGGACCTACTGCC
STOP
6
3
fatrvir CG
mkvir GCG
covid1 TCCG
lovvir TCCA
```

【Output】

Out of valid range, please input again:

After Inserting:

```
1232 covid1 AGTCGACGAGCTCGGACT
```

1212 hivirus TCTACTCATTTCGACCACTGA
1244 mkvir AGGCGCATCGCGCTTCC
1776 fatrvir TGCATCGACCGTACGCG
1555 sfavir TGCATGCTAGCT
1237 jpkvir TGGACCTACTGCC
1256 witvir ATCGGTCAATTAACG
1135 aidsvir CGGTATCGACGGATCCAGT

After sorting:

1135 aidsvir CGGTATCGACGGATCCAGT
1212 hivirus TCTACTCATTTCGACCACTGA
1232 covid1 AGTCGACGAGCTCGGACT
1237 jpkvir TGGACCTACTGCC
1244 mkvir AGGCGCATCGCGCTTCC
1256 witvir ATCGGTCAATTAACG
1555 sfavir TGCATGCTAGCT
1776 fatrvir TGCATCGACCGTACGCG
loc: 5 9 13 15
loc: 2 9
NotFound
NoVirus

【Run】


```

Please input the first group of viruses' information (ending with 'STOP'):
1232 covidl AGTCGACGAGCTCGGACT
1212 hivirus TCTACTCATTCGACCACTGA
1244 mkvir AGGCGCATCGCGCTTCC
1256 witvir ATCGGTCAATTAACG
1135 aidsvir CGGTATCGACGGATCCAGT
STOP

Please input the second group of viruses' information (ending with 'STOP'):
1776 fatrvir TGCATCGACCGTACGCG
1555 sfavir TGCATGCTAGCT
1237 jpkvir TGGACCTACTGCC
STOP

Please input the place for insert:
6
Out of valid range, please input again:
3

After Inserting:
1232 covidl AGTCGACGAGCTCGGACT
1212 hivirus TCTACTCATTCGACCACTGA
1244 mkvir AGGCGCATCGCGCTTCC
1776 fatrvir TGCATCGACCGTACGCG
1555 sfavir TGCATGCTAGCT
1237 jpkvir TGGACCTACTGCC
1256 witvir ATCGGTCAATTAACG
1135 aidsvir CGGTATCGACGGATCCAGT

After sorting:
1135 aidsvir CGGTATCGACGGATCCAGT
1212 hivirus TCTACTCATTCGACCACTGA
1232 covidl AGTCGACGAGCTCGGACT
1237 jpkvir TGGACCTACTGCC
1244 mkvir AGGCGCATCGCGCTTCC
1256 witvir ATCGGTCAATTAACG
1555 sfavir TGCATGCTAGCT
1776 fatrvir TGCATCGACCGTACGCG

Please input virus's name and sub-sequence: fatrvir CG
loc: 5 9 13 15

Please input virus's name and sub-sequence: mkvir GCG
loc: 2 9

Please input virus's name and sub-sequence: covidl TCCG
NotFound
Please input virus's name and sub-sequence: lovvir TCCA
NoVirus
Please input virus's name and sub-sequence:

```

请在此处粘贴程序和运行结果截图

四、Huffman 树 (10 分)

Huffman 树在编码中有着广泛的应用。在本题中，考虑 Huffman 树的构造过程。

给出一列数 $\{p_i\} = \{p_0, p_1, \dots, p_{n-1}\}$ ，用这列数构造 Huffman 树的过程如下：

- (1) 找到 $\{p_i\}$ 中最小的两个数，设为 p_a 和 p_b ，将 p_a 和 p_b 从 $\{p_i\}$ 中删除掉，然后将它们的和加入到 $\{p_i\}$ 中。这个过程的费用记为 $p_a + p_b$ 。
- (2) 重复步骤 (1)，直到 $\{p_i\}$ 中只剩下一个数。

在上面的操作过程中，把所有的费用相加，就得到了构造 Huffman 树的总费用。

例如，对于数列 $\{p_i\} = \{5, 3, 8, 2, 9\}$ ，Huffman 树的构造过程如下：

- (1) 找到{5, 3, 8, 2, 9}中最小的两个数，分别是 2 和 3，从{ p_i }中删除它们并将和 5 加入，得到{5, 8, 9, 5}，费用为 5。
- (2) 找到{5, 8, 9, 5}中最小的两个数，分别是 5 和 5，从{ p_i }中删除它们并将和 10 加入，得到{8, 9, 10}，费用为 10。
- (3) 找到{8, 9, 10}中最小的两个数，分别是 8 和 9，从{ p_i }中删除它们并将和 17 加入，得到{10, 17}，费用为 17。
- (4) 找到{10, 17}中最小的两个数，分别是 10 和 17，从{ p_i }中删除它们并将和 27 加入，得到{27}，费用为 27。
- (5) 现在，数列中只剩下一个数 27，构造过程结束，总费用为 5+10+17+27=59。

注：该过程每次只删去两个数，如同时有 5,5,5 那么只删去两个 5，三个元素变成两个元素 5,10。

请编制程序，对于给定的一个数列，求出用该数列构造 Huffman 树的总费用。

1. 函数 Input () (3 分)

该函数内，用户从键盘输入两行。第一行包含一个正整数 n 。第二行是用空格分隔的 n 个正整数，表示 p_0, p_1, \dots, p_{n-1} ，构建一个数列，并传回主函数。保证每个数不超过 1000，无需判断。

2. 函数 Huffman() (4 分)

该函数内，根据个人需求自行编写。实现所需功能即可。如不需要，可不编写。

3. 主函数 main() (3 分)

- (1) 定义所需变量。
- (2) 调用 Input ()，从键盘读入数列长度 n ，和原始数列的值。
- (3) 调用 Huffman()，或者在主函数中实现，求解用输入数列构造 Huffman 树的总费用。
- (4) 输出总费用。

[程序示例]

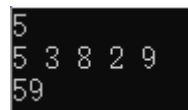
【Input】

```
5
5 3 8 2 9
```

【Output】

```
59
```

【Run】



请在此处粘贴程序和运行结果截图

批阅要点：

1、Input()函数 (3)

- a) 输入正确；1 分
- b) 形参列表有意义、参数传递正确；2 分

2、Huffman()和 main()函数 (7)

- a) 在主函数中能够正确定义数组，及各个参数在传递时的指针；2 分
- b) 循环结构判断终止正确；1 分
- c) 完成数组的一次更新，排序、删除、求和、添加，过程正确；3 分
- d) 输出总费用正确；1 分

//以上为全部题目。