|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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：**    1. **文件包名“学号.zip”，如2019123456.zip**    2. **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。** 7. **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。   1. **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“AGGTTGGATG”。   1. **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”。   1. **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”。   1. **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】    *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 5. **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。   1. **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≤x<m，0≤y<n，0≤z<p，1≤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。   1. **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）：  （3）计算A的L2范数（所有元素的平方之和的平方根）：  则稀疏度可表示为：  该函数内，计算三维矩阵A的稀疏度，并将结果传回主函数。   1. **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  其中为矩阵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。   1. **主函数main()（5分）** 2. 定义所使用到的变量。 3. 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。 4. 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。 5. 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】    *请在此处粘贴程序和运行结果截图*  批阅要点：   1. InputMatrix()函数（8）    1. 函数头及形参表列意义正确；1分    2. 在函数中输入的3个大于0的整数m、n、p正确；1分    3. 根据输入的m、n、p动态生成的矩阵正确（一层1分）；3分    4. 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）    1. 多行输入正确；1分    2. 输入终止条件正确；1分    3. 对A的全部元素均赋值正确；2分    4. 参数传递正确；1分 3. Sparseness ()函数（6）    1. 函数头及形参表列意义正确；1分    2. 范数数值类型、求值正确；3分    3. 求解稀疏度正确；2分 4. Centroid ()函数（6）    1. 函数头及形参表列意义正确；1分    2. 质心数值类型、求值正确；4分    3. 返回值正确；1分 5. main()函数（5）    1. 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    2. 函数调用正确（与定义的函数均能对应）；1分    3. 输出稀疏度，精度和数值均正确；1分    4. 输出质心，精度和数值均正确；1分 6. **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”。 4. **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   1. **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。   1. **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。   1. **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。   1. **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“ATCGGATCGGCTCG”，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 “ATCGGGGCGGCTCG”，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。 3. **主函数main()（4分）** 4. 两次调用Create()，生成两个链表/结构体数组\*list1和\*list2。 5. 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。 6. 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。 7. 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。 8. 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 TCTACTCATTCGACCACTGA  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 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 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】    *请在此处粘贴程序和运行结果截图*   1. **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{pi}={p0, p1, …, pn-1}，用这列数构造Huffman树的过程如下：   1. 找到{pi}中最小的两个数，设为pa和pb，将pa和pb从{pi}中删除掉，然后将它们的和加入到{pi}中。这个过程的费用记为pa + pb。 2. 重复步骤（1），直到{pi}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{pi}={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{pi}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{pi}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{pi}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{pi}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p0, p1, …, pn-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。   1. **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。   1. **主函数main()（3分）** 2. 定义所需变量。 3. 调用Input ()，从键盘读入数列长度n，和原始数列的值。 4. 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。 5. 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）    1. 输入正确；1分    2. 形参列表有意义、参数传递正确；2分 2. Huffman()和main()函数（7）    1. 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    2. 循环结构判断终止正确；1分    3. 完成数组的一次更新，排序、删除、求和、添加，过程正确；3分    4. 输出总费用正确；1分   //以上为全部题目。 |
|  |