第四、五章串和数组 自测卷答案　　姓名_____　班级_____

题号	一	二	三	四	五	总分
题分	20	15	20	15	30	100
得分						

一、填空题（每空 1 分，共 20 分）

1. 设有两个串 p 和 q，求 q 在 p 中首次出现的位置的运算称作__模式匹配__
2. 设目标 T=”abcdcdccbaa”，模式 P=“cdcc”，则第__6__次匹配成功。
3. 子串的定位运算称为串的模式匹配；__被匹配的主串__称为目标串，__子串__称为模式。
4. 若 n 为主串长，m 为子串长，则串的古典（朴素）匹配算法最坏的情况下需要比较字符的总次数为__(n-m+1)*m__。
5. 〖00 年计算机系考研题〗设数组 a[1…60, 1…70]的基地址为 2048，每个元素占 2 个存储单元，若以列序为主序顺序存储，则元素 a[32,58]的存储地址为__**9188**__。
   答：考虑 0 行 0 列，（58 列×61 行＋32 行）×2 字节＋基址 2048=9188？？
6. __不包含任何字符（长度为 0）的串__称为空串；__由一个或多个空格（仅由空格符）组成的串__称为空白串。
7. 三元素组表中的每个结点对应于稀疏矩阵的一个非零元素，它包含有三个数据项，分别表示该元素的__行下标__、__列下标__和__元素值__。
8. 设 S=“A;/document/Mary.doc”，则 strlen(s)=　20　，“/”的字符定位的位置为　3　。
9. 假设有二维数组 $A_{6\times8}$，每个元素用相邻的 6 个字节存储，存储器按字节编址。已知 A 的起始存储位置（基地址）为 1000，则数组 A 的体积（存储量）为__288 B__；末尾元素 A_{57} 的第一个字节地址为__1282__；若按行存储时，元素 A_{14} 的第一个字节地址为__(8+4)×6+1000=1192__；若按列存储时，元素 A_{47} 的第一个字节地址为__(6×7＋4)×6＋1000）＝1276__。
10. 求下列广义表操作的结果：
    （1） GetHead【((a,b),(c,d))】===__(a, b)__;　　//头元素不必加括号
    （2） GetHead【GetTail【((a,b),(c,d))】】===__(c,d)__;
    （3） GetHead【GetTail【GetHead【((a,b),(c,d))】】】===__b__;
    （4） GetTail【GetHead【GetTail【((a,b),(c,d))】】】===__（d）__;

二、单选题（每小题 1 分，共 15 分）

（ B ）1. 以下对二维数组 a 进行正确初始化的是：

A) int　a[2][3]={ {1,2},{3,4},{5,6} };
B) int　a[ ][3]={1,2,3,4,5,6 };
C) int　a[2][ ]={1,2,3,4,5,6};
D) int　a[2][ ]={ { 1,2},{3,4}};

（ D ）2. 〖李〗设串 s1=’ABCDEFG’，s2=’PQRST’，函数 con(x,y)返回 x 和 y 串的连接串，subs(s, i, j)返回串 s 的从序号 i 开始的 j 个字符组成的子串，len(s)返回串 s 的长度，则 con(subs(s1, 2, len(s2)), subs(s1, len(s2), 2))的结果串是：

A．BCDEF　　B．BCDEFG　　C．BCPQRST　　D．BCDEFEF

解：con(x,y)返回 x 和 y 串的连接串，即 con(x,y)＝‘ABCDEFGPQRST’；
subs(s, i, j)返回串 s 的从序号 i 开始的 j 个字符组成的子串，则
subs(s1, 2, len(s2))＝subs(s1, 2, 5)=’BCDEF’;　subs(s1, len(s2), 2)＝subs(s1, 5, 2)=’EF’;
所以 con(subs(s1, 2, len(s2)), subs(s1, len(s2), 2))＝con(’BCDEF’, ’EF’)之连接，即 BCDEFEF

(　B　)3. 设矩阵 A 是一个对称矩阵，为了节省存储，将其下三角部分（如下图所示）按行序存放在一维数组 B[**1, n(n-1)/2**]中，对下三角部分中任一元素 $a_{i,j}(i\leqslant j)$，在一维数组 B 中下标 k 的值是：

A．i(i-1)/2+j-1　　B．i(i-1)/2+j　　C．i(i+1)/2+j-1　　D．i(i+1)/2+j

解：注意 B 的下标要求从 1 开始。
先用第一个元素去套用，可能有 B 和 C；
再用第二个元素去套用 B 和 C，B=2 而 C＝3（不符）；
所以选 B

$$A=\begin{bmatrix} a_{1,1} & & & \\ a_{2,1} & a_{2,2} & & \\ \Lambda & & & \\ a_{n,1} & a_{n,2} & \Lambda & a_{n,n} \end{bmatrix}$$

（　B　）4. 〖李〗设有两个串 p 和 q，求 q 在 p 中首次出现的位置的运算称作：
A．连接　　B．模式匹配　　C．求子串　　D．求串长

（　A　）5. 〖01 年计算机系考研题〗假设有 60 行 70 列的二维数组 a[1…60, 1…70]以列序为主序顺序存储，其基地址为 10000，每个元素占 2 个存储单元，那么第 32 行第 58 列的元素 a[32,58]的存储地址为_____。（无第 0 行第 0 列元素）

A．16902　　B．16904　　C．14454　　D．答案 A, B, C 均不对

答：（57 列×60 行＋31 行）×2 字节＋10000=16902(A)

6.【94 程 P12】　从供选择的答案中，选出应填入下面叙述___？___内的最确切的解答，把相应编号写在答卷的对应栏内。

有一个二维数组 A，行下标的范围是 1 到 6，列下标的范围是 0 到 7，每个数组元素用相邻的 6 个字节存储，存储器按字节编址。那么，这个数组的体积是___A___个字节。假设存储数组元素 A[1,0]的第一个字节的地址是 0，则存储数组 A 的最后一个元素的第一个字节的地址是___B___。若按行存储，则 A[2,4]的第一个字节的地址是___C___。若按列存储，则 A[5,7]的第一个字节的地址是___D___。

供选择的答案

A～D：①12　② 66　③ 72　④ 96　⑤ 114　⑥ 120
⑦ 156　⑧ 234　⑨ 276　⑩ 282　（11）283　（12）288

答案：ABCD=12,　10,　3,　9

7.【91 初程 P78】　从供选择的答案中，选出应填入下面叙述___？___内的最确切的解答，把相应编号写在答卷的对应栏内。

有一个二维数组 A，行下标的范围是 0 到 8，列下标的范围是 1 到 5，每个数组元素用相邻的 4 个字节存储。存储器按字节编址。假设存储数组元素 A[0,1]的第一个字节的地址是 0。

存储数组 A 的最后一个元素的第一个字节的地址是___A___。若按行存储，则 A[3,5]和 A[5,3]

的第一个字节的地址分别是__B__和__C__。若按列存储，则 A[7,1]和 A[2,4]的第一个字节的地址分别是__D__和__E__。

供选择的答案

A～E：①28　② 44　③ 76　④ 92　⑤ 108

⑥ 116　⑦ 132　⑧ 176　⑨ 184　⑩ 188

答案：ABCDE=8, 3, 5, 1, 6

三、简答题（每小题 5 分，共 15 分）

1. 什么是野指针？如何避免野指针？

答：野指针：指向不确定地址的指针变量。（即没有初始化）使用野指针易因内存泄露出现段错误。而造成内存泄露的原因有两个：

1.访问了没有权限的内存（平时我们正确使用指针的时候，系统应经将相应的内存分配给用户，但是如果指向没有分配的内存，系统会判定我们没有权限）

2.访问了已经释放了的内存。

2. （软件技术？）已知二维数组 Am,m 采用按行优先顺序存放，每个元素占 K 个存储单元，并且第一个元素的存储地址为 Loc(a11)，请写出求 Loc(aij)的计算公式。如果采用列优先顺序存放呢？

答：公式教材已给出，此处虽是方阵，但行列公式仍不相同；

按行存储的元素地址公式是： Loc(aij)= Loc(a11) +[(i-1)*m+(j-1)] * K

按列存储的元素地址公式是： Loc(aij)= Loc(a11) +[(j-1)*m+(i-1)] * K

3.尽可能详细的解释一下 strlen

答：strlen(...)是函数，要在运行时才能计算。参数必须是字符型指针（char*）。当数组名作为参数传入时，实际上数组就退化成指针了。它的功能是：返回字符串的长度。该字符串可能是自己定义的，也可能是内存中随机的，该函数实际完成的功能是从代表该字符串的第一个地址开始遍历，直到遇到结束符 NULL。返回的长度大小不包括 NULL。

四、计算题（每题 10 分，共 20 分）

1. 【严题集 4.8 ②】 已知主串 s=’ADBADABBAABADABBADADA’, 模式串 pat=’ADABBADADA’。写出模式串的 nextval 函数值，并由此画出 KMP 算法匹配的全过程。

解：（由演示程序得知）nextval 函数值为 0 1 0 2 1 0 1 0 4 0　在第 12 个字符处发现匹配！

s=’ADBADABBAABADABBADADA’

pat=’ADABBADADA’

2. （P60 4-19）下列各三元组表分别表示一个稀疏矩阵，试写出它们的稀疏矩阵。

(1)
$$\begin{bmatrix} 6\,4\,6 \\ 1\,2\,2 \\ 1?\,2\,12 \\ 3\,1\,3 \\ 4\,4\,4 \\ 5\,3\,6 \\ 6\,1\,16 \end{bmatrix}$$

(2)
$$\begin{bmatrix} 4\,5\,5 \\ 1\,1\,1 \\ 2\,4\,9 \\ 3\,2\,8 \\ 3\,5\,6 \\ 4\,3\,7 \end{bmatrix}$$

解：（1）为 6×4 矩阵，非零元素有 6 个，但其中一数下标有误？（2）为 4×5 矩阵，非零元素有 5 个

0 2 0 0 12 0 0 0 3 0 0 0 0 0 0 4 0 0 6 0 16 0 0 0	改为 2,1,12

1 0 0 0 0 0 0 0 9 0 0 8 0 0 6 0 0 7 0 0

五、算法设计题（每题 10 分，共 30 分）

1. 给定一整型数字 a[]={a[0],...,a[n])},找出连续子串{a[x]),a[x+1],...,a[y]},使得和最大，其中，0<=x<=y<=n。要求时间复杂度为 O（n）。

解：

算法思路：

（1）和最大的子串一定是以数组 a 中的某一个元素为结束，所以我们分别对每一个元素作为当前子串的结尾计算当前和最大的子串，再对计算出的所有子串和进行比较，最大的那个就是解。

（2）计算以元素 a（i+1）结尾的和最大的子串：假定以 a（i）结尾的和最大子串为 L[i]，和为 sum[i]，则当 sum[i]+a[i+1]>a[i+1],那 L[i+1]为 L[i]加上 a[i+1]，sum[i+1]=sum[i]+a[i+1]；否则 L[i+1]为 a[i]，sum[i+1]=a[i+1]；为什么这样？注意“连续”，也就是说以 a[i+1]结尾的子串必定包含 a[i+1]，那和最大的子串 L[i+1]只能是 a[i+1]或者 L[i]加上 a[i+1]，显然，sum[i]+a[i+1]>a[i+1]则不取 a[i+1]而取 L[i]加上 a[i+1]，否则就取 a[i+1];

```
struct SubMax//最大和的子串
{
    int start;//子串开始位置
    int end;//子串结束位置
    int sum;//和
};
SubMax maxsub(int a[],int size)
{
    int **sub=new int*[size];//以所有元素为结尾的最大子串信息数组，共 size 个一维数组，
//每个一维数组含 3 个元素，0 下标子串为开始位置，1 下标为结束位置，2 下标为和
    for(int k=0;k<size;k++)
    {
        sub[k]=new int[3];
    }
    sub[0][0]=0;
    sub[0][1]=0;
    sub[0][2]=a[0];
    for(int i=0;i<size-1;i++)
```

```
    {
        if(sub[i][2]+a[i+1]>a[i+1])
        {
            sub[i+1][0]=sub[i][0];
            sub[i+1][1]=i+1;
            sub[i+1][2]=sub[i][2]+a[i+1];
        }
        else
        {
            sub[i+1][0]=i+1;
            sub[i+1][1]=i+1;
            sub[i+1][2]=a[i+1];
        }
    }
    int max_sub=0;
    int maxsubsum=0;
    for(int j=0;j<size;j++)
    {
        if(sub[j][2]>maxsubsum)
        {
            maxsubsum=sub[j][2];
            max_sub=j;
        }
    }
    SubMax s;
    s.start=sub[max_sub][0];
    s.end=sub[max_sub][1];
    s.sum=sub[max_sub][2];
    for(int m=0;m<size;m++)
    {
        delete sub[m];
    }
    return s;
}
```

2. 写出将字符串反序的**递归**或**递推**算法，例如字符串为“abcsxw”，反序为“wxscba”(注：这也是严题集 4.10③ 编写对串求逆的递推算法)

算法思路：

① 假定用单链表结构存储字符串；

if 没有到尾部字符就不停调用自身函数，直至到达末尾，再从尾部返回并打印字符；

否则就打印当前字符并返回。

```
Invert(stringlistnode *p){
if(!p)return(0);
else Invert(p->next);
```

递归算法的一般形式：（殷人凯题集 P102）

```
void p(参数表){
if （递归结束条件）
可直接求解步骤；                基本项
else p（较小的参数）；          归纳项
}
```

```
DLR(x*root)
{if(!root)return(0);
printf(“%d”,root->data);
DLR(root->lchild);
DLR(root->rchild);
}
```

printf(“%c”, p->data)
}

如果当前串长为 0,则 return(-1)
否则开始递归：
if　串没有到末尾，则 P=P->next; 再调用 invert(s);
else printf(p->data);
return

4.10

```
void String_Reverse(Stringtype s,Stringtype &r)//求 s 的逆串 r
{
  StrAssign(r,''); //初始化 r 为空串
  for(i=Strlen(s);i;i--)
  {
    StrAssign(c,SubString(s,i,1));
    StrAssign(r,Concat(r,c)); //把 s 的字符从后往前添加到 r 中     /这是递推算法？
  }
}//String_Reverse
```

3. 【严题集 5.18⑤】**试设计一个算法，将数组 An 中的元素 A[0]至 A[n-1]循环右移 k 位，并要求只用一个元素大小的附加存储，元素移动或交换次数为 O(n)**

解：5.18
分析:要把 A 的元素循环右移 k 位,则 A[0]移至 A[k],A[k]移至 A[2k].....直到最终回到 A[0].然而这并没有全部解决问题,因为有可能有的元素在此过程中始终没有被访问过,而是被跳了过去.分析可知,当 n 和 k 的最大公约数为 p 时,只要分别以 A[0],A[1],...A[p-1]为起点执行上述算法,就可以保证每一个元素都被且仅被右移一次,从而满足题目要求.也就是说,A 的所有元素分别处在 p 个"循环链"上面.举例如下:
n=15,k=6,则 p=3.
第一条链:A[0]->A[6],A[6]->A[12],A[12]->A[3],A[3]->A[9],A[9]->A[0]. /已“顺便”移动了 A[6]、A[12]…
第二条链:A[1]->A[7],A[7]->A[13],A[13]->A[4],A[4]->A[10],A[10]->A[1].
第三条链:A[2]->A[8],A[8]->A[14],A[14]->A[5],A[5]->A[11],A[11]->A[2].
恰好使所有元素都右移一次.
虽然未经数学证明,但作者相信上述规律应该是正确的. 程序如下：

```
void RSh(int A[n],int k)//把数组 A 的元素循环右移 k 位,只用一个辅助存储空间
{
  for(i=1;i<=k;i++)
    if(n%i==0&&k%i==0) p=i;//求 n 和 k 的最大公约数 p
  for(i=0;i<p;i++)
  {
    j=i;l=(i+k)%n;temp=A[i];
    while(l!=i)
    {
```

```
            A[j]=temp;
            temp=A[l];
            A[l]=A[j];
            j=l;l=(j+k)%n;
        }//  循环右移一步
        A[i]=temp;
    }//for
}//RSh
```