

《计算机程序设计》作业 No-09及第8次上机

作业内容要点： 指针

【姓名_____学号_____】

【要求】本次作业程序要求用指针实现

- (一)在计算机上编程程序，加上必要的注释。
(二)上机实验，经助教检查通过后，复制源码并记录实验结果，完成报告。
(三)实验报告：记录调试及改错过程；知识点或方法技巧的收获心得。



1、将 $n \times n$ 方阵中的前4个最小元素放置到方阵的四个角。【实验要点：二维数组、指针、函数】

说明：

- 1) 设计一个函数，实现将任意的 $n \times n$ 方阵 ($n \geq 3$) 的前四个最小元素放置到方阵四个角的位置（顺序为：左上、右上、左下、右下）；元素集合不变，四角之外的其余元素位置变动不作限制。
- 2) 在主函数中输入，输入内容第一行为 n ，第二行为 n^2 个整数即方阵的元素。调用上述函数，然后输出处理后的方阵。

输入样例：

3

4 2 3 1 5 7 6 8 9

输出样例：

1 6 2

9 5 7

3 8 4

(一)【源码】

```
1.  //
2.  //  main.c
3.  //  121001
4.  //
5.  //  Created by 李佩哲 on 2021/11/30.
6.  //
7.
8.  #include <stdio.h>
9.  #include <stdlib.h>
10. #include <memory.h>
```

```

11.
12. int compar(const void *a, const void *b)
13.     {return *(double*)a - *(double*)b;}
14.
15. double **change(double pm[][100], int n) {
16.
17.     double matrix[n*n];
18.     double pmatrix[n][n];
19.     for(int i=0; i<n; i++)
20.         for(int j=0; j<n; j++)
21.             pmatrix[i][j] = pm[i][j];
22.     int r=0;
23.     for(int i=0; i<n; i++)
24.         for(int j=0; j<n; j++) {
25.             matrix[r] = pmatrix[i][j];
26.             r++;
27.         }
28.     qsort(matrix, n*n, sizeof(double), compar);
29.     pmatrix[0][0] = *matrix;
30.     pmatrix[0][n-1] = *(matrix+1);
31.     pmatrix[n-1][0] = *(matrix+2);
32.     pmatrix[n-1][n-1] = *(matrix+3);
33.     r=4;
34.     for (int i=0; i<n; i++)
35.         for (int j=0; j<n; j++)
36.             if (!(i==0 || i==n-1) && (j==0 || j==n-1))) {
37.                 pmatrix[i][j] = *(matrix+r);
38.                 r++;
39.             }
40.
41.     double **p = (double **) malloc(n * sizeof(double *)); //先申请M
    个指针型字节的空间
42.     for (int i=0; i<n; i++)
43.         p[i] = (double *) malloc(n * sizeof(double)); //然后依次按一维
    申请
44.     for(int i=0; i<n; i++)
45.         for(int j=0; j<n; j++)
46.             *((double *)p+n*i+j) = *((double *)pmatrix+n*i+j)
    + 1; //找地址, 如同数据结构中矩阵找地址相同, 首地址+ (次数行数-1) *总列数
    +次数列数-1, i, j都是从0开始, 可以不用减1
47.     return p;
48. }
49.
50. int main(int argc, const char * argv[]) {
51.     int n;
52.     scanf("%d", &n);
53.     double pmatrix[100][100] = {0};
54.     for(int i=0; i<n; i++)
55.         for(int j=0; j<n; j++)
56.             scanf("%lf", &pmatrix[i][j]);
57.     double **p = change(pmatrix, n);
58.     for(int i=0; i<n; i++) {

```

```

59.         for(int j=0;j<n;j++)
60.             printf("%lf ",*((double *)p+n*i+j)-1);
61.         printf("\n");
62.     }
63.     return 0;
64. }

```

(二) 【运行结果】

```

1. 3
2. 1 2 3 4 5 6 7 8 9
3. 1.000000 5.000000 2.000000
4. 6.000000 7.000000 8.000000
5. 3.000000 9.000000 4.000000
6. Program ended with exit code: 0

```

(三) 【实验报告】

使用二维指针以及malloc在堆中申请的内存来传递函数返回值二维数组

2、 编程实现求方差的命令行。【实验要点：命令行参数，指针】

说明：

- 1) 通过命令行参数输入若干实数，把这些实数的字符串形式转换为double类型的数据，并根据参数个数将这些数据存储在动态分配的内存中。
- 2) 计算这些数据的方差并输出。
- 3) 提示：可使用库函数pow，atof。原型如下：

```
#include <math.h>
```

```
double pow(double x, double y); //幂函数，返回xy
```

```
#include <string.h>
```

```
double atof(const char *str); //将字符串str转化为浮点数并返回该浮点数
```

输入样例：(假定可执行程序的名称是fangcha.exe)

fangcha.exe 12 3 5 9

输出样例：

12.187500

(一) 【源码】

```

65. //
66. //  main.c
67. //  121002
68. //
69. //  Created by 李佩哲 on 2021/12/1.
70. //
71.
72. #include <stdio.h>
73. #include <stdlib.h>
74. #include <string.h>
75. #include <math.h>
76.
77. int main(int argc, const char * argv[]) {
78.     float val[100]={0},s=0,x=0;//val各个数值, s为方差, x为平均值
79.     for(int i=1;i<argc;i++)//argv转换为val
80.         val[i]=atof(argv[i]);
81.     for (int i=1;i<argc;i++)//求平均值
82.         x+=val[i]/(argc-1);
83.     for (int i=1; i<argc; i++)//求方差
84.         s+=powf(val[i]-x, 2)/(argc-1);
85.     printf("\ns^2=%f\n",s);
86.     return 0;
87. }

```

(二) 【运行结果】

```

7.  s^2=12.187500
8.  Program ended with exit code: 0

```

(三) 【实验报告】

命令行参数argc *argv[]的使用

3. 将一个字符串循环右移n位。用指针实现。【实验要点：字符串，指针】

说明：

- 1) 输入字符串，长度限50字符以内，若超出50则只取前50个；
- 2) n是用户输入的一个正整数；
- 3) 循环右移一位是指将最后一个字符移到字符串最左面，其余字符均向右移动一个位置。循环右移n位是将上述循环右移一位的动作重复n次。**(怎样减少循环次数从而提高程序效率？ 是否可以不用循环嵌套？)**
- 4) 编写函数完成循环右移功能，在主函数中输入字符串和整数n，调用循环右移函数，输出右移之后的字符串。

输入样例：

abcdefghijklmn

10

输出样例：

efghijklmnabcd

(一)【源码】

```
88. //
89. //  main.c
90. //  121003
91. //
92. //  Created by 李佩哲 on 2021/12/1.
93. //
94.
95. #include <stdio.h>
96. #include <stdlib.h>
97. #include <string.h>
98.
99. char *right(char str[51],int n){
100.     char *p=(char *)malloc(50*sizeof(char));
101.     unsigned long l=strlen(str);
102.     char ans[l+1];
103.     for(unsigned long i=0;i<n;i++)
104.         ans[i]=str[i+(int)(l-(unsigned long)n)];
105.     for(int i=n;i<l;i++)
106.         ans[i]=str[i-n];
107.     p=&ans[0];
108.     return p;
109. }
110.
111. int main(int argc, const char * argv[]) {
112.     char str0[51],str[51];
113.     gets(str0);
114.     for(int i=0;i<50;i++)
115.         str[i]=str0[i];
116.     str[50]='\0';
117.     int n;
118.     scanf("%d",&n);
119.     char *p=right(str, n);
120.     unsigned long l=strlen(str);
121.     for(int i=0;i<(int)l;i++)
122.         printf("%c",*(p+i));
123.     printf("\n");
124.     return 0;
```

```
125.}
```

(二) 【运行结果】

```
126.warning: this program uses gets(), which is unsafe.
```

```
127.abcdefghijklmn
```

```
128.10
```

```
129.efghijklmnabcd
```

```
130.Program ended with exit code: 0
```

(三) 【实验报告】

同上，用一维指针与malloc来传递返回值