**哈尔滨工程大学**

**《程序设计基础》实验报告**

**基础实践二**

姓 名： 班 级：

学 号：

实验时间: 年 月 日

|  |  |
| --- | --- |
| 成绩 |  |

**哈尔滨工程大学计算机基础课程教学中心**

实验五：数组

实验题目1：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：使用选择排序法对数组进行排序，每次循环都会使当前元素为最大元素。

实验代码及注释：

#**include** <**stdio**.**h**>

**void** **selection\_sort**(**int** a[], **int** size)

{

**int** temp,max;

**for** (**int** i = 0; i < size; ++i) {

max = i;

**for** (**int** j = i + 1; j < size; ++j)

**if** (a[j] > a[max]) max = j; *//遍历数组找到最大元素的索引*

temp = a[i]; *//swap*

a[i] = a[max];

a[max] = temp;

}

**for** (**int** i = 0; i < size; ++i) *//输出*

printf(**"%d** **"**, a[i]);

printf(**"\n"**);

}

**int** **main**(**int** argc, **char**\* argv[])

{

**int** arr[5];

**for** (**int** i = 0; i < 5; ++i)

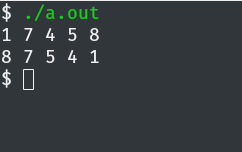
scanf(**"%d"**, arr + i);

selection\_sort(arr, 5);

**return** 0;

}

验证与结论：



总结与心得体会：只要求输出的话就无需进行元素swap，从而优化运行时间

#**include** <**stdio**.**h**>

**void** **selection\_sort**(**int** a[], **int** size)

{

**int** max;

**for** (**int** i = 0; i < size; ++i) {

max = i;

**for** (**int** j = i + 1; j < size; ++j)

**if** (a[j] > a[max]) max = j; *//遍历数组找到最大元素的索引*

printf(**"%d** **"**,a[max]); *//输出最大元素*

a[max]=a[i]; *//让max元素等于当前元素*

}

printf(**"\n"**);

}

**int** **main**(**int** argc, **char**\* argv[])

{

**int** arr[5];

**for** (**int** i = 0; i < 5; ++i)

scanf(**"%d"**, arr + i);

selection\_sort(arr, 5);

**return** 0;

}

实验题目2：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：将矩阵的行列互换得到的新矩阵称为转置矩阵

实验代码及注释：

#**include** <**stdio**.**h**>

**void** **matrix\_transpose**(**int** m[4][4])

{

**int** out[4][4]; *//转置后的数组*

**for** (**int** i = 0; i < 4; ++i) {

**for** (**int** j = 0; j < 4; ++j)

out[i][j] = m[j][i]; *//转置*

}

**for** (**int** i = 0; i < 4; ++i) { *//输出数组*

**for** (**int** j = 0; j < 4; ++j)

printf(**"%3d** **"**,out[i][j]);

printf(**"\n"**);

}

}

**int** **main**(**int** argc, **char**\* argv[])

{

**int** m[4][4];

**for** (**int** i = 0; i < 16; ++i)

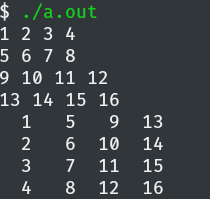
scanf(**"%d"**, (**int**\*)m + i);

matrix\_transposed(m);

**return** 0;

}

验证与结论：



总结与心得体会：只要求输出的话就无需用out数组

#**include** <**stdio**.**h**>

**void** **matrix\_transpose**(**int** m[4][4])

{

**for** (**int** i = 0; i < 4; ++i) {

**for** (**int** j = 0; j < 4; ++j)

printf(**"%3d"**, m[j][i]); *//直接输出转置后的数组*

printf(**"\n"**);

}

}

**int** **main**(**int** argc, **char**\* argv[])

{

**int** m[4][4];

**for** (**int** i = 0; i < 16; ++i)

scanf(**"%d"**, (**int**\*)m + i);

matrix\_transposed(m);

**return** 0;

}

实验六：结构体

实验题目1：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：熟练掌握struct的初始化

实验代码及注释：

#**include** <**stdio**.**h**>

**typedef** **struct** {

**int** **num**;

**float** **score**;

**char** **name**[1024];

} stu;

**int** **main**(**int** argc, **char**\* argv[])

{

stu s = { 741, 100.0f ,**"Lin"**}; *//值初始化*

printf(**"name=%s\n"** *//利用规则“相间的字符串字面值会拼接”*

**"score=%f\n"** *//来达到美化代码的效果*

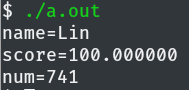
**"num=%d\n"**,

s.**name**, s.**score**, s.**num**);

**return** 0;

}

验证与结论：



总结与心得体会：分别初始化结构的数据

#**include** <**stdio**.**h**>

#**include** <**string**.**h**>

**typedef** **struct** {

**int** **num**;

**float** **score**;

**char** **name**[1024];

} stu;

**int** **main**(**int** argc, **char**\* argv[])

{

stu s; *//分别初始化*

s.**num**=557;

s.**score**=100.0f;

strcpy(s.**name**, **"Lin"**);

printf(**"name=%s\n"**

**"score=%f\n"**

**"num=%d\n"**,

s.**name**, s.**score**, s.**num**);

**return** 0;

}

实验题目2：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：初始化结构数组，用循环求平均值

实验代码及注释：

#**include** <**stdio**.**h**>

**typedef** **struct** {

**int** **num**;

**float** **score**;

**char** **name**[1024];

} stu;

**int** **main**(**int** argc, **char**\* argv[])

{

stu sa[]={ *//初始化数组*

{0,100.0f,**"foo0"**},

{1,99.9f,**"foo1"**},

{2,98.8f,**"foo2"**},

{3,97.7f,**"foo3"**},

{4,96.6f,**"foo4"**},

};

**float** average=0; *//利用变量计算平均值*

**for**(**int** i=0;i<5;++i)

average+=sa[i].**score**;

printf(**"average=%f\n"**,average/5);

**return** 0;

}

验证与结论：

2019-05-21_14-21

总结与心得体会：用原始的相加在处求平均值

#**include** <**stdio**.**h**>

**typedef** **struct** {

**int** **num**;

**float** **score**;

**char** **name**[1024];

} stu;

**int** **main**(**int** argc, **char**\* argv[])

{

stu sa[]={ *//初始化数组*

{0,100.0f,**"foo0"**},

{1,99.9f,**"foo1"**},

{2,98.8f,**"foo2"**},

{3,97.7f,**"foo3"**},

{4,96.6f,**"foo4"**},

};

printf(**"average=%f\n"**, *//直接相加*

(sa[0].**score**+

sa[1].**score**+

sa[2].**score**+

sa[3].**score**+

sa[4].**score**)/5);

**return** 0;

}

实验七：指针

实验题目1：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：用传递指针的方式调用函数来对三个数进行排序

实验代码及注释：

#**include** <**stdio**.**h**>

**void** **sort**(**int** \*a,**int** \*b,**int** \*c){

**int** t; *//swap中使用的中间变量*

**if**(\*a<\*b) t=\*b,\*b=\*a,\*a=t; *//swap*

**if**(\*b<\*c) t=\*c,\*c=\*b,\*b=t;

**if**(\*a<\*b) t=\*b,\*b=\*a,\*a=t;

}

**int** **main**(**int** argc, **char** \*argv[])

{

**int** a,b,c;

scanf(**"%d** **%d** **%d"**,&a,&b,&c);

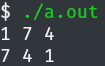
sort(&a,&b,&c);

printf(**"%d** **%d** **%d\n"**,a,b,c);

**return** 0;

}

验证与结论：



总结与心得体会：用异或的性质省去了中间变量

#**include** <**stdio**.**h**>

**void** **sort**(**int** \*a,**int** \*b,**int** \*c){

**if**(\*a<\*b) \*a^=\*b,\*b^=\*a,\*a^=\*b; *//利用异或的性质swap*

**if**(\*b<\*c) \*b^=\*c,\*c^=\*b,\*b^=\*c;

**if**(\*a<\*b) \*a^=\*b,\*b^=\*a,\*a^=\*b;

}

**int** **main**(**int** argc, **char** \*argv[])

{

**int** a,b,c;

scanf(**"%d** **%d** **%d"**,&a,&b,&c);

sort(&a,&b,&c);

printf(**"%d** **%d** **%d\n"**,a,b,c);

**return** 0;

}

实验题目2：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：用传递指针的方式调用函数来逆向输出数组

实验代码及注释：

#**include** <**stdio**.**h**>

**void** **reverse**(**int** \*a,**int** n){

**for**(**int** i=n-1;i>=0;--i) *//使用for结构来输出数组*

printf(**"%d** **"**,\*(a+i));

}

**int** **main**(**int** argc, **char** \*argv[])

{

**int** a[5];

**for**(**int** i=0;i<5;++i)

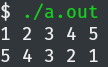
scanf(**"%d"**,a+i);

reverse(a,5);

**return** 0;

}

验证与结论：



总结与心得体会：用while结构输出

#**include** <**stdio**.**h**>

**void** **reverse**(**int** \*a,**int** n){

**int** \*e=a+n-1;

**while**(e>=a) *//使用while结构输出数组*

printf(**"%d** **"**,\*(e--));

}

**int** **main**(**int** argc, **char** \*argv[])

{

**int** a[5];

**for**(**int** i=0;i<5;++i)

scanf(**"%d"**,a+i);

reverse(a,5);

**return** 0;

}

实验八：文件

实验题目1：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：使用标准库的I/O函数操纵文件

实验代码及注释：

#**include** <**stdio**.**h**>

**int** **main**(**int** argc, **char** \*argv[])

{

FILE \*fd;

**if**((fd=fopen(**"file1"**, **"w"**))==**NULL**) *//判断函数调用是否成功*

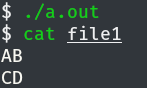
**return** 1;

fputs(**"AB\nCD\n"**, fd); *//写入*

**return** 0;

}

验证与结论：



总结与心得体会：w和wb的区别在不同的系统上实现不同，在Windows上文本文件的换行符是\r\n所以用w打开文件每写入一个\n前面都会补上\r，用wb打开则不会这样而Unix系的系统会忽略w和wb的区别。

实验题目2：

出处：（教材例4-1 | 实验例4-1 | 实验题目1 | 自编）

设计思想：使用标准库的I/O函数的EOF概念的实现来判断是否读到文件尾

实验代码及注释：

#**include** <**stdio**.**h**>

**int** **main**(**int** argc, **char**\* argv[])

{

FILE\* fd;

**int** c;

**if** ((fd = fopen(**"file1"**, **"r"**)) == **NULL**)

**return** 1;

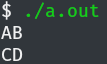
**while**((c=getc(fd))!=**EOF**) //判断是否到文件尾

putchar(c); //输出到stdout

**return** 0;

}

验证与结论：



总结与心得体会：

#**include** <**stdio**.**h**>

**int** **main**(**int** argc, **char**\* argv[])

{

FILE\* fd;

**int** c;

**if** ((fd = fopen(**"file1"**, **"r"**)) == **NULL**)

**return** 1;

c = getc(fd);

**while** (!feof(fd)) { //判断是否到文件尾

putchar(c);

c=getc(fd); //避免出现把EOF写入到文件的现象

}

**return** 0;

}

**哈尔滨工程大学**

**《程序设计基础》实验报告**

**综合实践**

小组成员：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **班级** | **学号** | **分工** | **成绩** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

实验时间: 年 月 日

**哈尔滨工程大学计算机基础课程教学中心**

实验题目：

设计思想：

成员任务分工：

运行环境：

编程语言：

成员1的工作：

姓名：

学号：

任务分工：

设计思想：

代码行数：

实验代码及注释：

验证与结论：

成员2的工作：

姓名：

学号：

任务分工：

设计思想：

代码行数：

实验代码及注释：

验证与结论：