《算法与数据结构》实验指导及报告书

**（2019版）**

**2020 - 2021 学年 第 2 学期**

**姓 名：**

**学 号：**

**班 级： 计科2班 \_**

**指导教师：\_\_ 贾春花\_\_\_\_\_\_\_**

**楚雄师范学院信息学院**

# 预备实验 C语言的函数、数组、指针、结构体知识

2020级 计算机科学与技术 专业 2 班 学号： 姓名：

**实验日期**：2021年3月10日---- 3月10日**报告日期**：2021年3月10日

### 一、实验目的

1、复习C语言中函数、数组、指针、结构体等的概念。

2、熟悉利用C语言进行程序设计的一般方法。

### 二、实验预习

复习以下C语言中的概念

1. 函数：函数是一段可以重复使用的代码，用来独立地完成某个功能，它可以接收用户传递的数据，也可以不接收。接收用户数据的函数在定义时要指明参数，不接收用户数据的不需要指明，根据这一点可以将函数分为有参函数和无参函数。
2. 数组：数组（Array）就是一些列具有相同类型的数据的集合，这些数据在内存中依次挨着存放，彼此之间没有缝隙。

3、指针：数据在内存中的地址也称为指针

4、结构体：C语言结构体（Struct）从本质上讲是一种自定义的数据类型，只不过这种数据类型比较复杂，是由 int、char、float 等基本类型组成的。

### 三、实验内容和要求

1、调试程序：输出100以内所有的素数（用函数实现）。

#include<stdio.h>

int isprime(int n){

int m;

for(m=2;m\*m<=n;m++)

if(n%m==0) return 0;

return 1;

}

int main(){

int i;printf("\n");

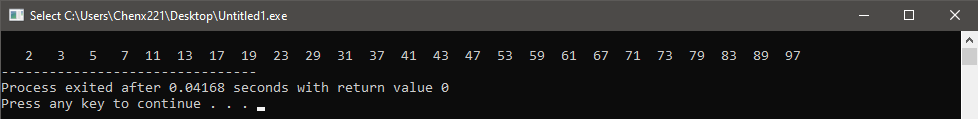
for(i=2;i<100;i++)

if(isprime(i)) printf("%4d",i);

return 0;

}

**运行结果：**



2、 调试程序：对一维数组中的元素进行逆序排列。

#include<stdio.h>

#define N 10

int main(){

int a[N]={0,1,2,3,4,5,6,7,8,9},i,temp;

printf("\nthe original Array is:\n ");

for(i=0;i<N;i++)

printf("%4d",a[i]);

for(i=0;i<(N/2);i++){

temp=a[i];

a[i]=a[N-i-1];

a[N-i-1]=temp;

}

printf("\nthe changed Array is:\n");

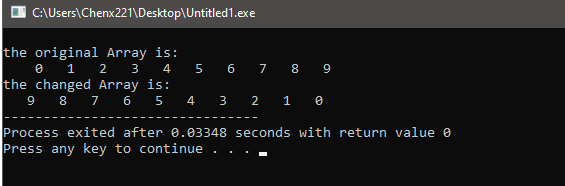
for(i=0;i<N;i++)

printf("%4d",a[i]);

return 0;

}

**运行结果：**



3、调试程序：在二维数组中，若某一位置上的元素在该行中最大，而在该列中最小，则该元素即为该二维数组的一个鞍点。要求从键盘上输入一个二维数组，当鞍点存在时，把鞍点找出来。

#include<stdio.h>

#define M 3

#define N 4

int main(){

int a[M][N],i,j,k;

printf("\nPlease type:\n");

for(i=0;i<M;i++)

for(j=0;j<N;j++)

scanf("%d",&a[i][j]);

for(i=0;i<M;i++){

for(j=0;j<N;j++)

printf("%4d",a[i][j]);

printf("\n");

}

for(i=0;i<M;i++){

k=0;

for(j=1;j<N;j++)

if(a[i][j]>a[i][k])

k=j;

for(j=0;j<M;j++)

if(a[j][k]<a[i][k])

break;

if(j==M)

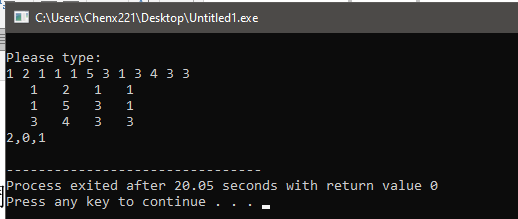
printf("%d,%d,%d\n",a[i][k],i,k);

}

return 0;

}

**运行结果：**



4、 调试程序：利用指针输出二维数组的元素。

#include<stdio.h>

int main(){

int a[3][4]={1,3,5,7,9,11,13,15,17,19,21,23};

int \*p;

for(p=a[0];p<a[0]+12;p++){

if((p-a[0])%4==0) printf("\n");

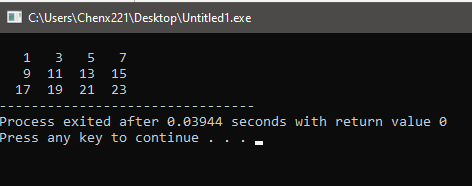
printf("%4d",\*p);

}

return 0;

}

**运行结果：**



5、 调试程序：设有一个学生表格，有姓名、年龄、班级3项，编程输入人员的数据，再以表格输出。

#include <stdio.h>

#define N 10

typedef struct student{

char name[8];

int age;

int classroom;

} PERSON;

int main(){

int i, n;

PERSON stu[N];

printf("\n 人数(<10):\n");

scanf("%d",&n);

for(i=0;i<n;i++){

printf("\n 请输入 %d 人员信息(name age class)\n",i+1);

scanf("%s%d%d",&stu[i].name, &stu[i].age, &stu[i].classroom);

}

printf("name age class/office\n");

for(i=0;i<n;i++){

printf("%s is %3d years old. His class is %d\n",stu[i].name, stu[i].age, stu[i].classroom);

}

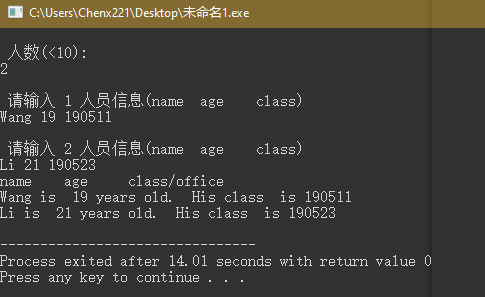
}

输入的数据：2

Wang 19 190511

Li 21190523

**运行结果：**



### 四、实验小结

实验0都是之前C语言学到过的内容，这个应该说是一个复习

\*部分截图是作业发布当晚用服务器截的，后面的截图都是在自己的电脑截的

# 实验1算法分析

2020级 计算机科学与技术 专业 2 班 学号： 姓名：

**实验日期**：2021年3月10日---- 3月10日 **报告日期**：2021年3月10日

### 一、实验目的

1、理解算法分析的含义。

2、掌握算法时间复杂度分析方法，并通过实验进行验证。

### 二、实验预习

复习以下概念

1、算法：算法,算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。

2、时间复杂度：在计算机科学中，时间复杂性，又称时间复杂度，算法的时间复杂度是一个函数，它定性描述该算法的运行时间。这是一个代表算法输入值的字符串的长度的函数。

### 三、实验内容和要求

**问题描述**：公鸡每只5文钱，母鸡每只3文钱，3只小鸡一文钱，现在用100文钱买100只鸡，问这100只鸡里有公鸡、母鸡、小鸡各几只？

1、完成下面的程序，用户输入钱的数目，输出购买的方案，统计该算法基本操作的执行次数，请提交运行结果，并分析该算法的时间复杂度。

#include <stdio.h>

int main(){

int i,j,k,n, m=0;

long long t=0;

printf("input n:");

scanf("%d",&n);

for(i=0;i<=n/5;i++)

for(j=0;j<=n/3; j++)

for(k=0;k<=n;k=k+3){

t++;

if(i\*5+j\*3+k/3==n&&i+j+k==n)

printf("m=%4d %4d %4d %4d\n",++m,i,j,k);

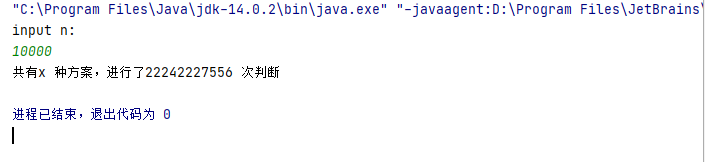
}

printf("共有%d种方案，进行了%d次判断\n",m,t);

return 0;

}

运行结果：

（使用kotlin）

2、请改进第1题的算法，使算法的时间复杂度T(n)=O(n\*n)；

程序：

#include <stdio.h>

int main(){

int i,j,k,n, m=0;

long long t=0;

printf("input n:");

scanf("%d",&n);

for(i=0;i<=n/5;i++)

for(j=0;j<=n/3; j++){

k=n-i-j;

t++;

if(k%3==0&&i\*5+j\*3+k/3==n&&i+j+k==n)

printf("m=%4d %4d %4d %4d\n",++m,i,j,k);

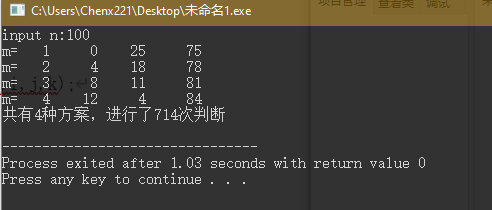
}

printf("共有%d种方案，进行了%d次判断\n",m,t);

return 0;

}

运行结果：



3、补充完成并运行下面程序，分析结果。

#include <stdio.h>

int main(){

int i,j,k,n,t=0,m=0;

printf("input n:");

scanf("%d",&n);

for(i=0;i<=n/5;i++){

t++;

j=(n-7\*i)/4;

k=n-i-j;

if( j>=0 && k%3==0 && i\*5+j\*3+k/3==n)

printf("m=%4d %4d %4d %4d\n",++m,i,j,k);

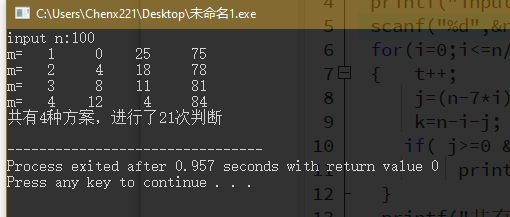
}

printf("共有%d种方案，进行了%d次判断\n",m,t);

return 0;

}

运行结果：



请对上述3个算法进行分析总结：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 购买方案数 | 算法1  判断次数 | 算法2  判断次数 | 算法3  判断次数 |
| 50 | 2 | 3179 | 187 | 11 |
| 100 | 4 | 24276 | 714 | 21 |
| 1000 | 36 | 22422756 | 67134 | 201 |
| 10000 | 358 | 22242227556 | 6671334 | 2001 |
| 100000 | 3572 | 22224222275556 | 666713334 | 20001 |
| 算法的时间复杂度： | | N^3 | N^2 | N |

### 四、实验小结

算法的效率很重要，糟糕的算法会导致低效率