## 实验报告三

## 实验二 数组&指针&字符串

### 1.实验目的要求

1. 掌握一维数组和二维数组的定义、初始化赋值、元素引用的方法。
2. 初步了解数组名与地址的关系。
3. 掌握字符数组和字符串函数的使用。
4. 掌握与数组有关的算法（特别是排序和查找算法）。
5. 掌握指针的概念，学会定义和使用指针变量。
6. 学会使用数组指针和指向数组的指针变量。
7. 学会使用字符串指针和指向字符串的指针变量。
8. 了解指向指针的指针的概念以及其使用方法。
9. 掌握指针、引用、数组做函数参数的传递机制。

### 2.实验设备

本人小米电脑

### 3.实验内容

**（1）阅读下列程序，写出其运行结果，并指出其功能。**

<1>

|  |
| --- |
| #include <iostream>  using namespace std;  const int N = 10;  int main()  { int a[N];  int s,i,j,t;  //1如下这段代码功能？  for(i = 0; i<N; i++){  if (i%2) a[i] = i\*i ;  else a[i] = 100-(i/2)\*(i/2);  }  //2如下这段代码功能？  for (i=0; i<N; i++ )  cout<<a[i]<<" ";  cout<<endl ;  //3如下这段代码功能？  s = 0 ;  for( i=0; i<N; i++ )  s = s + a[i];  cout<<"sum of a = "<<s;  cout<<endl;  //4如下这段代码功能？ for(i=0; i<N; i++){  t = a[i] ;  j = i-1 ;  while(j>=0&&t>a[j]){  a[j+1] = a[j] ;  j-- ;  }  a[j+1] = t ;  }  //5如下这段代码功能？  for(i=0; i<N; i++)  cout<<a[i] <<" ";  cout<<endl ;  return 0;  } |

分别描述上面代码中各个部分的功能，并给出输出结果。

**答：**

**1 给数组a赋值，其中下标为奇数的元素赋值为其下标的平方，下标为偶数的元素赋值为100减偶数除2的平方。**

**2 打印数组a的元素，以空格分隔**

****

**3 统计数组元素之和**

****

**4 数组重新排序，按从大到小逆序排列**

**5打印新数组a的元素，并以空格分隔**



<2>

#include <iostream>

#include <string>

using namespace std;

int main()

{ int i=0, base, n, j, num[30] ;

cin>>n>>base ;

do{ i++;

num[i] = n%base ;

n = n/base ;

}while(n!=0);

for(j=i; j>=1; j--){

switch(num[j]){

case 15:cout<<'F';break;

case 14:cout<<'E';break;

case 13:cout<<'D';break;

case 12:cout<<'C';break;

case 11:cout<<'B';break;

case 10:cout<<'A';break;

default: cout<<num[j];

}

}

return 0;

}

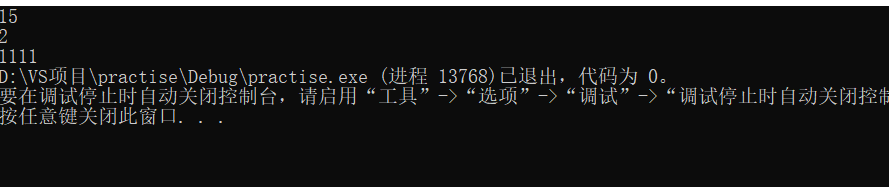
如果输入15，2，输出什么？

**答：**

**预期输出**

**1111**

**实际输出**

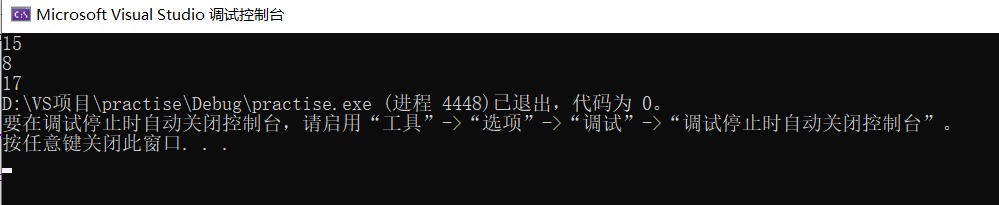


如果输入15，8，输出什么？

**答：**

**预期输出**

**17  
实际输出**



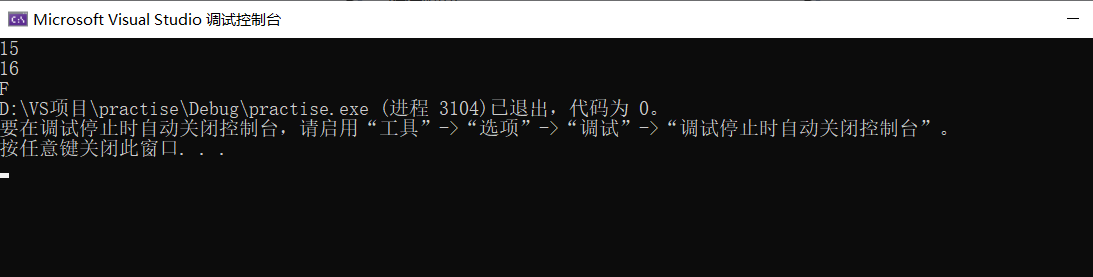
如果输入15，16，输出是什么？

**答：**

**预期输出**

**F**

**实际输出**



总结如上代码的功能是什么？

**答：代码完成将n（十进制表示）转换成所输入的base进制并输出**

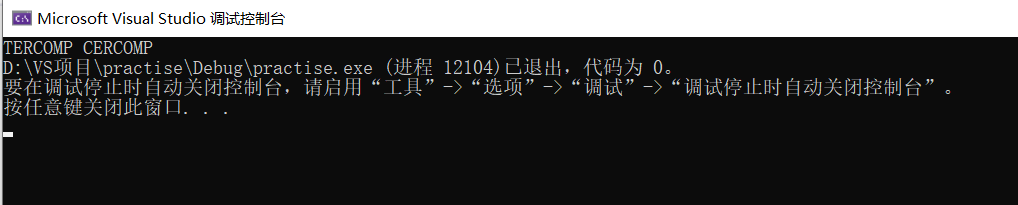
<3>

|  |
| --- |
| #include <iostream>  #include <string.h>  using namespace std;  int main(){  char a[20] = "TER", b[20] = "COMP";  int i = 0 ;  strcat(a,b);  while (a[i++]!='\0')  b[i] = a[i] ;  cout<<a <<" "<<b;  } |

程序输出是什么，分析原因？

**答：**

**输出结果**

****

**分析原因**

**Strcat（m，n）函数将m和n连接后将其赋给m，所以a输出的结果为TERCOMP，“a[i++]!='\0'”表达的意思是a[i]！=’\0’，再i++，最后进行循环中的语句，因此第一次进入循环时i=1，所以只将a[i](i>=1)的值赋给b，因此b输出结果为CERCOMP**

<4>

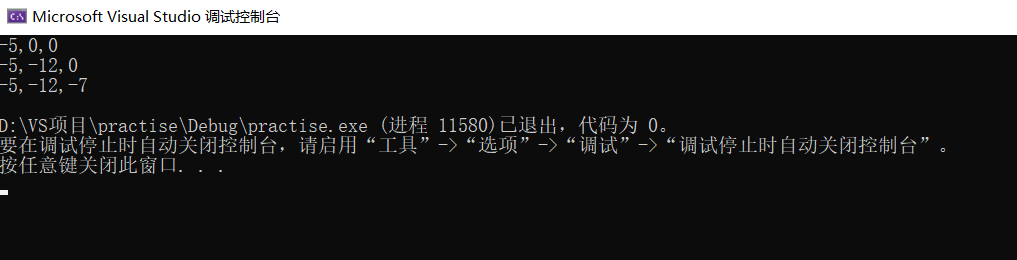
|  |
| --- |
| #include <iostream>  using namespace std;  void sub(int x, int y, int \*z)  { \*z = y - x ;  }  int main()  { int a=0,b=0,c=0;  sub(10, 5, &a);  cout<<a<<','<<b<<','<<c<<endl;  sub(7, a, &b);  cout<<a<<','<<b<<','<<c<<endl;  sub(a, b, &c);  cout<<a<<','<<b<<','<<c<<endl;  return 0;  } |

Sub函数里第三个形参的作用是什么？

**答：记录所传入的两个形参值相减的值**

主函数的输出结果是什么，分析下原因？

**答：**

****

**分析原因**

**第一次调用sub函数，由于a传入的是地址，所以a的值变为-5，因此输出-5，0，0**

**第二次调用sub函数，由于b传入的是地址，且a-7=-5-7=-12，因此输出-5，-12，0**

**第三次调用sub函数，由于c传入的是地址，且b-a=-12-（-5）=-7，因此输出-5，-12，-7**

<5>

|  |
| --- |
| #include <iostream>  #include <string.h>  using namespace std;  int main()  { int stre(char[ ]) ;  char str[10], \*p = str ;  gets(p);  cout<<stre(p)<<endl;  return 0;  }  int stre(char str[ ])  { int num = 0 ;  while(\*(str + num)!='\0') num++ ;  return num;  } |

Stre函数的功能是什么？

**答：计算字符串有几个元素并返回**

如果你输入超过9个字符，发生了什么，结果会怎样？从数组越界的角度分析一下。

**答：结果会发生异常**

**分析原因：存储字符串时，最后一位如果不输入默认为”\0”，因此如果输入10个字符，’\0’将变为第11位，而数组只有10位，此时会发生越界。**

**（2）编写程序实现下列问题的求解。**

<1> 编程产生下列数组，并输出。**（至少选作两题，从一维数组和二维数组各选一题）**

一维数组

1) (1 4 9 16 25 36 49 64 81 100)

2) (1 3 6 10 15 21 28 36 45 55)

3) (1 2 3 5 8 13 21 34 55 89)

二维数组

1) 1 2 3 4 5 6 2) 1 2 3 .....n-1 0

2 3 4 5 6 0 2 3 4 ..... 0 1

3 4 5 6 0 1 3 4 5 ..... 1 2

4 5 6 0 1 2 ................

5 6 0 1 2 3 n-1 0 1 .....n-3 n-2

6 0 1 2 3 4 0 1 2 .....n-2 n-1

3) 1 2 3 4 5 6 4) A B C D E F

2 1 2 3 4 5 B C E H L Q

3 2 1 2 3 4 C D F I M R

4 3 2 1 2 3 D E G J N S

5 4 3 2 1 2 E F H K O T

6 5 4 3 2 1

**答：一维数组1）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

int a[10];

for (int i = 0; i < 10; i++)

{

a[i] = (i + 1) \* (i + 1);

}

for (int i = 0; i < 10; i++)

{

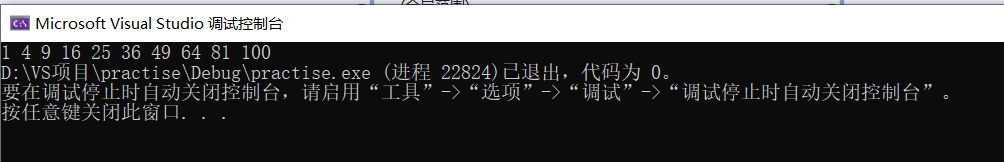
cout << a[i] << " ";

}

return 0;

}

**输出结果**

****

**一维数组2）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

int a[10];

int eve = 1;

int odd = 1;

for (int i = 0; i < 10; i++)

{

if (i % 2 == 0) //偶数

{

a[i] = eve \* (i + 1);

eve++;

}

else //奇数

{

a[i] = odd \* (i + 1) + odd;

odd++;

}

}

for (int i = 0; i < 10; i++)

{

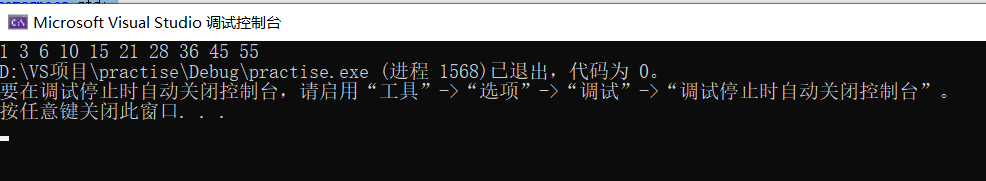
cout << a[i] << " ";

}

return 0;

}

**运行结果**

****

**一维数组3）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

int a[10];

a[0] = 1;

a[1] = 2;

for (int i = 2; i < 10; i++)

{

a[i] = a[i - 1] + a[i - 2];

}

for (int i = 0; i < 10; i++)

{

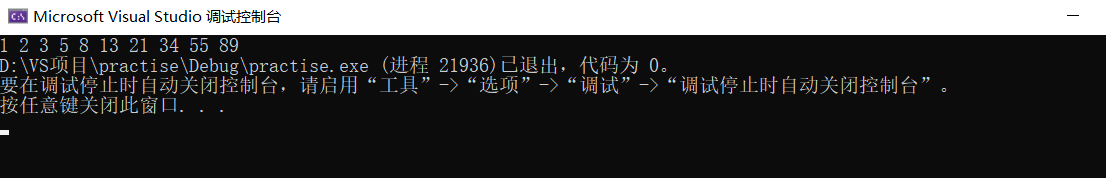
cout << a[i] << " ";

}

return 0;

}

**运行结果**

****

**二维数组1）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

char a[6][6];

string str = "1234560";

int p = 0;

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

a[i][j] = str[p];

p++;

if (p == 7)

{

p = 0;

}

}

p = i+1;

}

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

cout << a[i][j] << " ";

}

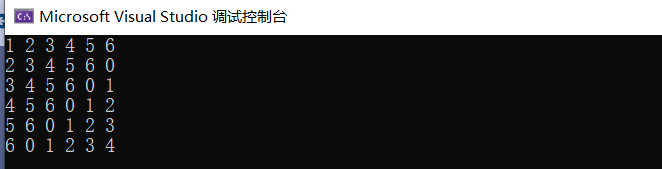
cout << endl;

}

return 0;

}

**运行结果**

****

**二维数组3）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

char a[6][6];

string str = "65432123456";

int p = 5;

for (int i = 0; i < 6; i++)

{

p = 5 - i;

for (int j = 0; j < 6; j++)

{

a[i][j] = str[p];

p++;

}

}

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

cout << a[i][j] << " ";

}

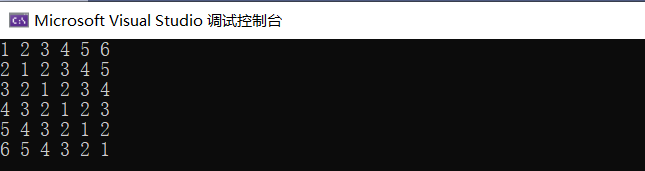
cout << endl;

}

return 0;

}

**运行结果**

****

**二维数组4）**

**源代码**

#include<iostream>

using namespace std;

int main()

{

char a[5][6];

string str = "ABCDEFGHIJKLMNOPQRST";

int p = 0;

for (int j = 0; j < 6; j++)

{

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

{

a[i][j] = str[p];

p++;

}

p = j + 1;

}

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

{

for (int j = 0; j < 6; j++)

{

cout << a[i][j] << " ";

}

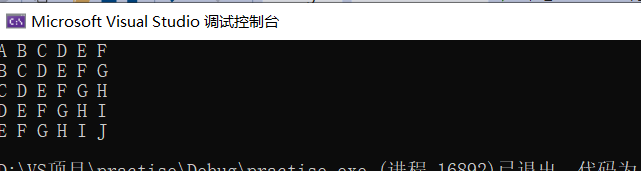
cout << endl;

}

return 0;

}

**运行结果**

****

**<2>- <10> 至少选做5题**

<2> 用**函数**实现将一个以字符串形式表示的十六进制数转换为一个十进制整数。例如，输入”A2”转换为162。

**源代码**

#include<iostream>

using namespace std;

#include<cstring>

#include<cmath>

int ChangeNum(string str)

{

int sum = 0;

int temp = 0;

bool flag = 1;

int m = 0;

for (int i = str.length()-1; i >=0; i--)

{

switch (str[i])

{

case 'A':

temp = 10;

flag = 0;

break;

case 'B':

temp = 11;

flag = 0;

break;

case 'C':

temp = 12;

flag = 0;

break;

case 'D':

temp = 13;

flag = 0;

break;

case 'E':

temp = 14;

flag = 0;

break;

case 'F':

temp = 15;

flag = 0;

break;

}

if (flag)

{

temp = str[i]-48;

}

sum += temp \* pow(16, m);

m++;

}

return sum;

}

int main()

{

string str;

cin >> str;

cout << "原先为："<<str << endl;

cout<<"转变后为："<<ChangeNum(str)<<endl;

return 0;

}

**运行结果**

****

<3> 编写出判断一个整数是否为素数的**函数**，并求出在2000以内的有十个以上的所有连续的非素数组。

**源代码**

#include<iostream>

using namespace std;

#include<cmath>

bool IsPrimeNum(int num)

{

int m = sqrt(num) + 1;

for (int i = 2; i < m; i++)

{

if (num % i == 0)

{

return false;

}

}

return true;

}

int main()

{

int a[10];

int temp = 0;

for (int i = 3; i < 2000; i++)

{

if (!IsPrimeNum(i))

{

a[temp] = i;

temp++;

if (temp == 10) //满10个连续非素数输出

{

for (int k = 0; k < 10; k++)

{

cout << a[k] << " ";

}

cout << endl;

temp = 0;

if (!IsPrimeNum(i + 1))

{

i = i - 9;

a[temp] = i;

}

}

if (IsPrimeNum(i + 1))

{

temp = 0;

}

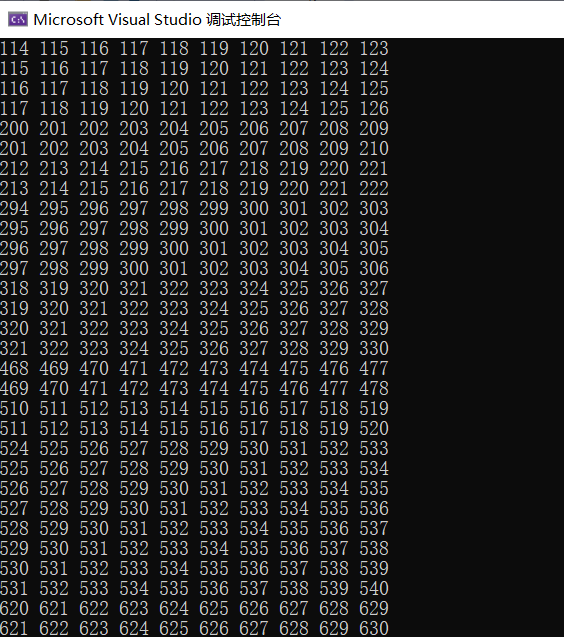
}

}

return 0;

}

**运行结果**

****

<4>将一个3\*3的矩阵转置，用一个**函数**实现。在主函数中输入以下矩阵元素：{2，4，6，8，10，12，14，16，18}。将数组名作为函数参数。函数调用后在主函数中输出已转置的矩阵。

**源代码**

#include<iostream>

using namespace std;

#include<cmath>

void ChangeMatrix(int a[][3])

{

int temp = 0;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < i; j++)

{

temp = a[i][j];

a[i][j] = a[j][i];

a[j][i] = temp;

}

}

}

int main()

{

int a[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cin >> a[i][j];

}

}

cout << "原先矩阵为：" << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << a[i][j] << " ";

}

cout << endl;

}

ChangeMatrix(a);

cout << "转置矩阵为：" << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << a[i][j] << " ";

}

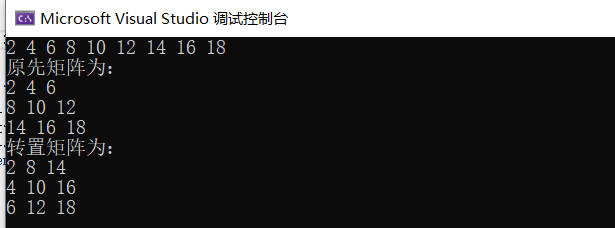
cout << endl;

}

return 0;

}

**运行结果**

****

<5>自己写一个strcmp函数，实现两个字符串的比较。当s1<s2时，返回-1；当s1=s2时，返回值= 0；当s1>s2时，返回正数。两个字符串s1,s2由main函数输入，strcmp**函数**的返回值在main函数中输出。

**源代码**

#include<iostream>

using namespace std;

#include<cmath>

#include<cstring>

int Mystrcmp(string s1,string s2)

{

int minlength = s1.length() > s2.length() ? s2.length() : s1.length();

for (int i = 0; i < minlength; i++)

{

if (s1[i] > s2[i])

{

return 1;

}

else if (s1[i] < s2[i])

{

return -1;

}

return 0;

}

}

int main()

{

string s1, s2;

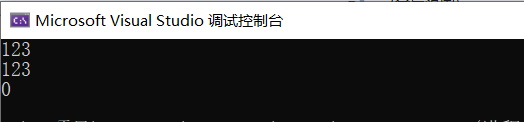
cin >> s1 >> s2;

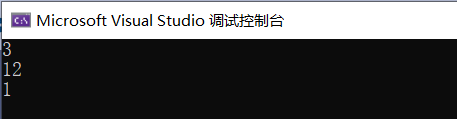
cout << Mystrcmp(s1,s2) << endl;

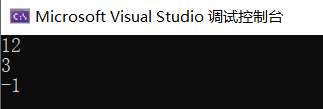
return 0;

}

**运行结果**

****

****

****

<6> 判断一个二维数组是否有“鞍点”，即某位置上的元素在该行上最大，在该列上最小。如有，输出其行列号和值；若无，给出提示。

**源代码**

#include<iostream>

using namespace std;

#include<cmath>

void IsSaddle(int a[3][3])

{

int k = 0;

bool flag = 0;

for (int i = 0; i < 3; i++)

{

int max = a[i][0];

flag = 0;

for (int j = 0; j < 3; j++)

{

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

{

max = a[i][j];

k = j;

}

}

for (int m = 0; m< 3; m++)

{

int min = a[i][k];

if (a[m][k] <min)

{

flag = 1;

}

}

if (flag == 0)

{

cout << "鞍点在" << i+1 << "行" << k+1 << "列" << endl;

return;

}

}

cout << "无鞍点" << endl;

}

int main()

{

int a[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cin >> a[i][j];

}

}

cout << "矩阵为：" << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout<< a[i][j]<<" ";

}

cout << endl;

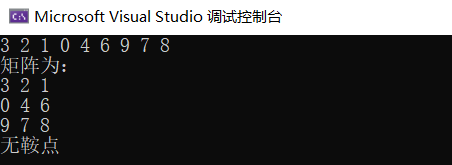
}

IsSaddle(a);

return 0;

}

**运行结果**

****

****

<7> 编程求一组整数的最大公因子和最小公倍数，讲任务分解为多个函数实现。

**源代码**

#include<iostream>

using namespace std;

int Max(int n1, int n2)

{

if (n1 < n2)//n1为较大数，n2为较小数

{

int temp = n1;

n1 = n2;

n2 = temp;

}

int r = 1;

while (r != 0)

{

r = n1 % n2;

n1 = n2;

n2 = r;

}

return n1;

}

int Min(int n1, int n2)

{

int max = Max(n1, n2);//最大公因子

return n1 \* n2 / max;

}

int main()

{

int n1, n2;

cin >> n1 >> n2;

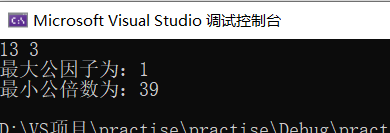
cout << "最大公因子为：" << Max(n1, n2) << endl;

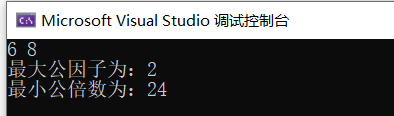
cout << "最小公倍数为：" << Min(n1, n2) << endl;

return 0;

}

**运行结果**

****

****

<8> 对于一个有序的数组，输入一个数，用折半查找法在数组中查找，如在数组中，则输出元素在数组中的位置；如不在，则输出-1。用函数实现。

**源代码**

#include<iostream>

using namespace std;

int binary\_search(int arr[],int num,int length)

{

int first = 0;

int last = length - 1;

while (first <= last)

{

int mid = (last + first) / 2;

if (arr[mid] < num)

{

first = mid + 1;

}

else if (arr[mid] > num)

{

last = mid - 1;

}

else

{

return mid + 1;

}

}

}

int main()

{

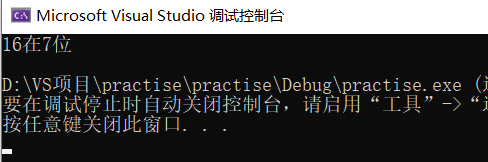
int arr[10] = { 1,2,3,6,8,10,16,18,19,33 };

cout << "16在" << binary\_search(arr, 16, 10) << "位" << endl;

return 0;

}

**运行结果**

****

<9> 用筛选法求出2到1000之间的所有素数，用函数实现。数组依次存储1~1000数据，函数运行结束后，该数组中非素数位置变为0，非零的素组元素即为素数。

注意： 筛选法求出2～N间的所有素数的方法是：首先将这些数全部放入一个数组中，然后重复下面的操作直到数组为空为止：

A）.找出其中的最小数K，则K一定是一个素数，因此可输出。

B）.从数组中删除K及其所有倍数。

**源代码**

#include<iostream>

using namespace std;

void IsPrime(int arr[], int N)

{

for (int i = 1; i < N; i++)

{

if (arr[i]) //最小素数

{

for (int j = 2; arr[i] \* j <= N; j++)

{

arr[arr[i]\*j- 1] = 0;

}

}

}

}

int main()

{

int a[1000];

for (int i = 0; i < 1000; i++) //1-1000

{

a[i] = i+1;

}

IsPrime(a, 1000);

for (int j = 0; j < 1000; j++)

{

cout << a[j] << " ";

if ((j+1) % 10 == 0)

{

cout << endl;

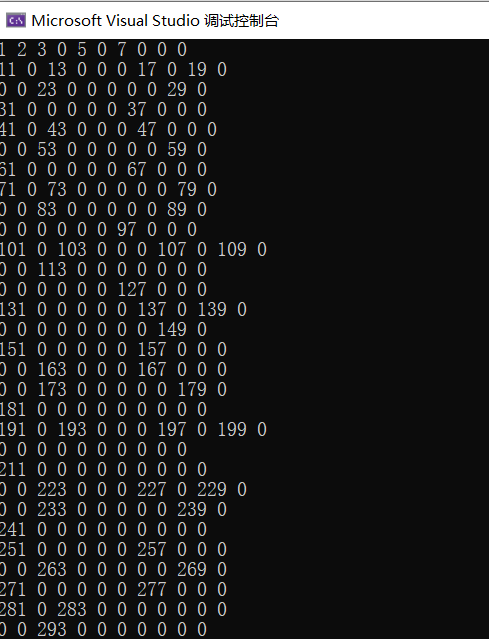
}

}

return 0;

}

**运行结果**

****

<10> 在某系的成绩登记册中，每个班最多有40个学生，每份成绩表中的成绩信息包括：学号（9位字符），姓名（8位字符），成绩（百分制），备注（20位字符）。设计程序以处理一个班级的成绩信息，包括输入、输出、查询（给定分数以上或以下的学生信息）、按分数排序等。

**源代码**

#include<iostream>

using namespace std;

#include<cstring>

void Sort(Student s[], int num)//成绩升序排序

{

Student temp;

for (int i = 0; i < num; i++)

{

for (int j = i; j < num-1; j++)

{

if (s[j].Getscore() > s[j + 1].Getscore())

{

temp = s[j];

s[j] = s[j+1];

s[i + 1] = temp;

}

}

}

}

void PrintSort(Student s[], int num)//输出按成绩升序排序后的学生信息

{

for (int i = 0; i < num; i++)

{

s[i].PrintInfor();

}

}

void PrintLower(Student s[], int num, int N)//输出比num成绩低的学生信息

{

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

{

if (s[i].Getscore() < num)

{

s[i].PrintInfor();

}

}

}

void PrintUpper(Student s[], int num, int N)//输出比num成绩高的学生信息

{

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

{

if (s[i].Getscore() > num)

{

s[i].PrintInfor();

}

}

}

class Student

{

public:

Student() {};//无参构造

Student(char id[9], char name[8], int score, char remark[20] = { 0 })//有参构造

{

for (int i = 0; i < 9; i++)

{

ID[i] = id[i];

}

for (int i = 0; i < 8; i++)

{

Name[i] = name[i];

}

Score = score;

for (int i = 0; i < 20; i++)

{

Remark[i] = remark[i];

}

}

void PrintInfor()//输出信息

{

cout << "学号为：" << endl;

for (int i = 0; i < 9; i++)

{

cout << ID[i];

}

cout << endl;

cout << "姓名为：" << endl;

for (int i = 0; i < 8; i++)

{

cout << Name[i];

}

cout << endl;

cout << "分数为：" << endl;

cout << Score << endl;

if (Remark[0])

{

cout << "备注：" << endl;

for (int i = 0; i < 20; i++)

{

if (!Remark[i])

{

break;

}

cout << Remark[i];

}

}

else

{

cout << "无备注" << endl;

}

}

int Getscore()//获取分数

{

return Score;

}

private:

char ID[9];

char Name[8];

int Score;

char Remark[20];

};

int main()

{

return 0;

}

### 4.分析与思考

<1> 在本实验内容2- <10>习题中，若要求成绩部分不仅可以是百分数，而且也可能是五分制（优、良、中、及格和不及格），应如何存储数据以及处理？

**答：若要求百分制成绩和五分制成绩只能存取一种，可使用结构体**

**源代码**

Student类中：

private:

char ID[9];

char Name[8];

union Score

{

char grade;

int percent;

};

char Remark[20];

**若百分制成绩和五分制成绩可以同时存取，则可以用结构体或类定义成绩，再进行类的嵌套**

**源代码**

class Score

{

public:

Score() {};

Score(int percent)

{

m\_percent = percent;

}

Score(char grade, int percent = 0)

{

m\_grade = grade;

m\_percent = percent;

}

private:

char m\_grade;

int m\_percent;

};

Student类中：

private:

char ID[9];

char Name[8];

Score score;

char Remark[20];

### 5.实验总结

1.通过这部分练习，在设计函数方面有了一些提高。学到了一些设计方法，比如筛选法等等，同时巩固了之前求最大公约数的辗转相除法、冒泡算法等。

2.通过这部分练习，加深了对数组、指针、字符串的理解，学到了部分字符串中库函数的处理方法，学会了部分指针与数组之间的转换，同时也复习了之前类的部分知识。

### 6.参考资料

1.[(24条消息) c++语言获得字符长度,c++怎么获取字符串长度？\_清风明月人间的博客-CSDN博客](https://blog.csdn.net/weixin_31304817/article/details/117095680)