1.定义一个”数据类型” datatype类，能处理包含字符型、整型、浮点型三种类型的数据，给出其构造函数。（）

**#include <iostream.h>**

**class datatype{**

**enum{**

**character,**

**integer,**

**floating\_point**

**} vartype;**

**union**

**{**

**char c;**

**int i;**

**float f;**

**};**

**public:**

**datatype(char ch) {**

**vartype = character;**

**c = ch;**

**}**

**datatype(int ii) {**

**vartype = integer;**

**i = ii;**

**}**

**datatype(float ff) {**

**vartype = floating\_point;**

**f = ff;**

**}**

**void print();**

**};**

**void datatype::print() {**

**switch (vartype) {**

**case character:**

**cout << "字符型: " << c << endl;**

**break;**

**case integer:**

**cout << "整型: " << i << endl;**

**break;**

**case floating\_point:**

**cout << "浮点型: " << f << endl;**

**break;**

**}**

**}**

**void main() {**

**datatype A('c'), B(12), C(1.44F);**

**A.print();**

**B.print();**

**C.print();**

**}**

**程序运行输出：**

**字符型: c**

**整型: 12**

**浮点型: 1.44**

2．用穷举法找出1~100间的质数，显示出来

**使用while循环语句：**

**#include <iostream.h>**

**#include <math.h>**

**void main()**

**{**

**int i,j,k,flag;**

**i = 2;**

**while(i <= 100)**

**{**

**flag = 1;**

**k = sqrt(i);**

**j = 2;**

**while (j <= k)**

**{**

**if(i%j == 0)**

**{**

**flag = 0;**

**break;**

**}**

**j++;**

**}**

**if (flag)**

**cout << i << "是质数." << endl;**

**i++;**

**}**

**}**

3.在程序中定义一个整型变量，赋以1~100的值，要求用户猜这个数，比较两个数的大小，把结果提示给用户，直到猜对为止。

**//使用do…while语句**

**#include <iostream.h>**

**void main() {**

**int n = 18;**

**int m = 0;**

**do{**

**cout << "请猜这个数的值为多少？(0~~100):";**

**cin >> m;**

**if (n > m)**

**cout << "你猜的值太小了！" << endl;**

**else if (n < m)**

**cout << "你猜的值太大了！" << endl;**

**else**

**cout << "你猜对了！" << endl;**

**}while(n != m);**

**}**

4.编写函数判断一个数是否是质数，在主程序中实现输入、输出。

**解：**

**#include <iostream.h>**

**#include <math.h>**

**int prime(int i); //判一个数是否是质数的函数**

**void main()**

**{**

**int i;**

**cout << "请输入一个整数：";**

**cin >> i;**

**if (prime(i))**

**cout << i << "是质数." << endl;**

**else**

**cout << i << "不是质数." << endl;**

**}**

**int prime(int i)**

**{**

**int j,k,flag;**

**flag = 1;**

**k = sqrt(i);**

**for (j = 2; j <= k; j++)**

**{**

**if(i%j == 0)**

**{**

**flag = 0;**

**break;**

**}**

**}**

**if (flag)**

**return 1;**

**else**

**return 0;**

**}**

**程序运行输出：**

**请输入一个整数：1151**

**1151是质数.**

5.编写函数求两个整数的最大公约数和最小公倍数。（第三章、10）

**#include <iostream.h>**

**#include <math.h>**

**int fn1(int i,int j); //求最大公约数的函数**

**void main()**

**{**

**int i,j,x,y;**

**cout << "请输入一个正整数：";**

**cin >> i ;**

**cout << "请输入另一个正整数：";**

**cin >> j ;**

**x = fn1(i,j);**

**y = i \* j / x;**

**cout << i << "和" << j << "的最大公约数是：" << x << endl;**

**cout << i << "和" << j << "的最小公倍数是：" << y << endl;**

**}**

**int fn1(int i, int j)**

**{**

**int temp;**

**if (i < j)**

**{**

**temp = i;**

**i = j;**

**j = i;**

**}**

**while(j != 0)**

**{**

**temp = i % j;**

**i = j;**

**j = temp;**

**}**

**return i;**

**}**

**程序运行输出：**

**请输入一个正整数：120**

**请输入另一个正整数：72**

**120和72的最大公约数是：24**

**120和72的最小公倍数是：360**

6.编写递归函数GetPower(int x， int y)计算x的y次幂， 在主程序中实现输入输出。

源程序：

**#include<iostream>**

**using namespace std;**

**long GetPower(int x,int y);**

**int main()**

**{**

**int i,j;**

**long s;**

**cout<<"Enter a i:";**

**cin>>i;**

**cout<<"j:";**

**cin>>j;**

**s=GetPower(i,j);**

**cout<<i<<"de"<<j<<"ci mi wei"<<s<<endl;**

**return 0;**

**}**

**long GetPower(int m,int n)**

**{**

**if(n==1)**

**return m;**

**else**

**return (m\*GetPower(m,n-1));**

**}程序运行输出：**

**Enter a number: 3**

**To what power? 4**

**3 to the 4th power is 81**

7.定义一个矩形类，有长、宽两个属性，有成员函数计算矩形的面积

**#include <iostream.h>**

**class Rectangle**

**{**

**public:**

**Rectangle(float len, float width)**

**{**

**Length = len;**

**Width = width;**

**}**

**~Rectangle(){};**

**float GetArea() { return Length \* Width; }**

**float GetLength() { return Length; }**

**float GetWidth() { return Width; }**

**private:**

**float Length;**

**float Width;**

**};**

**void main()**

**{**

**float length, width;**

**cout << "请输入矩形的长度：";**

**cin >> length;**

**cout << "请输入矩形的宽度：";**

**cin >> width;**

**Rectangle r(length, width);**

**cout << "长为" << length << "宽为" << width << "的矩形的面积为："**

**<< r.GetArea () << endl;**

**}**

**程序运行输出：**

**请输入矩形的长度：5**

**请输入矩形的宽度：4**

**长为5宽为4的矩形的面积为：20**

8.编写一个函数，统计一个英文句子中字母的个数，在主程序中实现输入、

**#include <iostream.h>**

**#include <stdio.h>**

**int count(char \*str)**

**{**

**int i,num=0;**

**for (i=0; str[i]; i++)**

**{**

**if ( (str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z') )**

**num++;**

**}**

**return num;**

**}**

**void main()**

**{**

**char text[100];**

**cout << "输入一个英语句子：" << endl;**

**gets(text);**

**cout << "这个句子里有" << count(text) << "个字母。" << endl;**

**}**

**程序运行输出：**

**输入一个英语句子：**

**It is very interesting!**

**这个句子里有19个字母。**

9.编写一个矩阵转置的函数，矩阵的维数在程序中由用户输入。

**#include <iostream.h>**

**void move (int matrix[3][3])**

**{**

**int i, j, k;**

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

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

**{**

**k = matrix[i][j];**

**matrix[i][j] = matrix[j][i];**

**matrix[j][i] = k;**

**}**

**}**

**void main()**

**{**

**int i, j;**

**int data[3][3];**

**cout << “输入矩阵的元素” << endl;**

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

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

**{**

**cout << “第” << i+1 << “行第” << j+1**

**<<”个元素为：“;**

**cin >> data[i][j];**

**}**

**cout << “输入的矩阵的为：” << endl;**

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

**{**

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

**cout << data[i][j] << ” “;**

**cout << endl;**

**}**

**move(data);**

**cout << “转置后的矩阵的为：” << endl;**

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

**{**

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

**cout << data[i][j] << ” “;**

**cout << endl;**

**}**

**}**

**程序运行输出：**

**输入矩阵的元素**

**第 1 行第1 个元素为：1**

**第1 行第2 个元素为：2**

**第1 行第3 个元素为：3**

**第2 行第1 个元素为：4**

**第2 行第2 个元素为：5**

**第2 行第3 个元素为：6**

**第3 行第1 个元素为：7**

**第3 行第2 个元素为：8**

**第3 行第3 个元素为：9**

**输入的矩阵的为：**

**1 2 3**

**4 5 6**

**7 8 9**

**转置后的矩阵的为：**

**1 4 7**

**2 5 8**

**3 6 9**

10.编写函数int index(char \*s， char \*t)，返回字符串t 在字符串s中出现的最左边的位置，如果在s中没有与t匹配的子串，就返回-1。

解：

源程序：

**#include <iostream.h>**

**int index( char \*s, char \*t)**

**{**

**int i,j,k;**

**for(i = 0; s[i] != ‘\0′; i++)**

**{**

**for(j = i, k = 0; t[k] != ‘\0′ && s[j] == t[k]; j++, k++);**

**if (t[k] ==’\0′)**

**return i;**

**}**

**return -1;**

**}**

**void main()**

**{**

**int n;**

**char str1[20],str2[20];**

**cout << “输入一个英语单词：”;**

**cin >> str1;**

**cout << “输入另一个英语单词：”;**

**cin >> str2;**

**n = index(str1,str2);**

**if (n > 0)**

**cout << str2 << “在” << str1 << “中左起第” << n+1**

**<< “个位置。”<<endl;**

**else**

**cout << str2 << “不在” << str1 << “中。” << endl;**

**}**

**11.编写函数reverse(char \*s)的倒序递归程序，使字符串s倒序。**

**解：**

**源程序：**

**#include <iostream.h>**

**#include <string.h>**

**void reverse(char \*s, char \*t)**

**{**

**char c;**

**if (s < t)**

**{**

**c = \*s;**

**\*s = \*t;**

**\*t = c;**

**reverse(++s, –t);**

**}**

**}**

**void reverse( char \*s)**

**{**

**reverse(s, s + strlen(s) – 1);**

**}**

**void main()**

**{**

**char str1[20];**

**cout << “输入一个字符串：”;**

**cin >> str1;**

**cout << “原字符串为：” << str1 << endl;**

**reverse(str1);**

**cout << “倒序反转后为：” << str1 << endl;**

**}**

**12.**一个Shape基类，在此基础上派生出Rectangle和Circle，二者都有GetArea()函数计算对象的面积。使用Rectangle类创建一个派生类Square。

解：

**#include <iostream.h>**

**#include <math.h>**

**#define pi 3.14**

**class shape**

**{ public:**

**virtual float area()=0;**

**};**

**class circle:public shape**

**{**

**private:**

**float r;**

**public:**

**circle(float r1)**

**{**

**r=r1;**

**}**

**float area()**

**{**

**return (float)pi\*r\*r;**

**}**

**};**

**class rectangle:public shape**

**{**

**private:**

**float width,height;**

**public:**

**rectangle(float w1,float h1)**

**{**

**width=w1;height=h1;**

**}**

**float area()**

**{**

**return width\*height;**

**}**

**};**

**class square : public rectangle**

**{**

**public:**

**square(float len):rectangle(len,len){};**

**~square(){};**

**float area(float len)**

**{**

**return len \* len;**

**};**

**};**

**int main()**

**{**

**shape\* s[2];**

**s[0]=new circle(1);**

**cout<<s[0]->area()<<endl;**

**s[1]=new rectangle(2,4);**

**cout<<s[1]->area()<<endl;**

**s[ 2 ] = new square( 3 );**

**cout << s[2]->area() << endl;**

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

**{**

**delete [] s[ i ];**

**}**

**return 0;**

}