从2015年开始，长沙理工大学计算机研究生复试考试科目将566改成现在的F0803，这里列举的是15~19年的编程题题目和答案。

统计了一下有2个C++的题目，其余为C语言题目。

2015年

//1.求出10至1000之内能同时被2、3、7整除的数，并输出。

#include<stdio.h>

int main()

{

int i;

for (i=10;i<=1000;i++)

{

if(i%2==0&&i%3==0&&i%7==0)

{

printf("%-4d",i);

}

}

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

/\*2.任意输入一个三位正整数，十位上的数字不动，将个位和百位上的数字交换，

够成一个新的正整数后输出。（例如：484变为684）\*/

#include<stdio.h>

int Exchange(int x);

int main()

{

int i,x;

printf("请输入一个三位正整数：\n");

scanf("%d",&x);

i = Exchange(x);

printf("%d",i);

return 0;

}

int Exchange(int x)

{

int a,b,c;

c = x%10;

b = (x/10)%10;

a = x/100;

return c\*100+b\*10+a;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

/\*3.编写自定义函数prime(int x)，判断x是否为素数。利用此函数编写程序找出

100~2000中的所有素数，并输出素数的个数。\*/

#include<stdio.h>

#include<math.h>

int prime(int x);

int main()

{

int x,i;

int j=0,flag=0;

printf("请输入一个数：\n");

scanf("%d",&x);

flag = prime(x);

if (flag==1)

{

printf("该数是素数\n");

}else{

printf("该数不是素数\n");

}

printf("100至2000之内的素数为\n");

for (i=100;i<=2000;i++)

{

if(prime(i))

{

printf("%-5d",i);

j++;

}

}

printf("\n素数的个数为%d个\n",j);

return 0;

}

int prime(int x)

{

int i;

for(i=2;i<=sqrt(x);i++)

{

if (x%i==0)

{

return 0;

}

}

if (i>sqrt(x))

{

return 1;

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

2016年

/\*1、使用C语言编写程序打印出10000以内的所有“完全数”，每个数之间用英文逗号

隔开。“完全数”是指一个正整数，其所有小于该数的因子之和等于该数本身。

例如：6=1+2+3，又如：28=1+2+4+7+14。\*/

#include<stdio.h>

int main()

{

int i,j,sum;

for (i=1;i<10000;i++)

{

sum = 0;

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

{

if (i%j==0)

{

sum+=j;

}

}

if (i==sum)

{

printf("%d,",i);

}

}

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

//2、用C语言编写程序，根据输入的三角形的三条边（a,b,c），判断是否能组成三

角形（如果任意两边之和大于第三边则认为可组成三角形），若可以则输出三角形的

类型（等边三角形、等腰三角形、直角三角形、一般三角形）及

面积（area=sqrt(s\*(s-a)\*(s-b)\*(s-c)); s=(a+b+c)/2);

否则输出“不能组成三角形”。\*/

#include<stdio.h>

#include<math.h>

#define EPS 1e-2

int main()

{

float a,b,c;

float s,area;

int flag = 0;

printf("请输入三角形的三条边：\n");

scanf("%f%f%f",&a,&b,&c);

if (a+b>c&&a+c>b&&b+c>a)

{

if (fabs(a-b)<=EPS||fabs(b-c)<=EPS||fabs(c-a)<=EPS)

{

printf("等腰");

flag = 1;

}

else if (fabs(a-b)<=EPS&&fabs(b-c)<=EPS&&fabs(c-a)<=EPS)

{

printf("等边");

flag = 1;

}

if (fabs(a\*a+b\*b-c\*c)<=EPS||fabs(a\*a+c\*c-b\*b)<=EPS||

fabs(b\*b+c\*c-a\*a)<=EPS)

{

printf("直角");

flag = 1;

}

if (flag==0)

{

printf("一般");

}

printf("三角形");

}else{

printf("不能组成三角形");

}

printf("\n面积为：");

s = (a+b+c)/2.0;

area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

printf("%f",area);

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

/\*3、已知交通工具类定义如下

class vehicle{

protected:

int wheels; //车轮数

float weight; //重量

public:

void init(int wheels,float weight);

float get\_weight(); //返回重量

int get\_wheels() (); //返回车轮数

void print(); //输出车轮数和重量，格式为“车轮数：x，重量：x“其中x为实

际值，不一定相同

};

要求：（1）实现这个类；

（2）定义并实现一个小车类car，是它的公有派生类，小车本身的私有属性有载人

数，小车的函数有init

（设置车轮数，重量和载人数），getpassenger（获取载人数），print（打印车

轮数，重量和载人数，格式为”

车轮数：x，重量：x载人数：x“x为实际值，不一定相同）。\*/

#include<iostream>

using namespace std;

class Vehicle

{

protected:

int wheels; //车轮数

float weight; //重量

public:

void init(int a,float b)

{

wheels=a;

weight=b;

}

float get\_weight()

{

return weight;

}

int get\_wheels()

{

return wheels;

}

void print()

{

cout<<"车轮数："<<get\_wheels()<<endl;

cout<<"重量："<<get\_weight()<<endl;

}

};

class Car:public Vehicle

{

private:

int people;//载人数

public:

int init(int a,int b,int c)

{

people=a;

wheels=b;

weight=c;

}

int getpassenger()

{

return people;

}

void print()

{

cout<<"车轮数："<<get\_wheels()<<endl;

cout<<"重量："<<get\_weight()<<endl;

cout<<"载人数："<<getpassenger()<<endl;

}

};

int main()

{

Vehicle v;

Car car;

v.init(2,12.2);

v.get\_weight();

v.get\_wheels();

v.print();

car.init(4,4,4);

car.getpassenger();

car.get\_weight();

car.get\_wheels();

car.print();

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

2017年

//1、请用C语言编写程序，求和：S=1-(1/2)+(1/3)-(1/4)+……+(1/n)，其中n=10

#include<stdio.h>

int main()

{

int i;

double sum=0;

int flag;

for (i=1;i<=10;i++)

{

flag = 1;

if(i%2==0)

{

flag=-1;

}

sum+=(1.0/i)\*flag;

}

printf("%lf",sum);

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

/\*2、请用C语言编写一个函数fun(char \*s)，则调用该函数之后，串中的内容为

“gfedcba”。\*/

#include<stdio.h>

#include<string.h>

#define N 20

void fun(char \*s);

int main()

{

char s[N];

fun(s);

puts(s);

}

void fun(char \*s)

{

char a[]="gfedcba";

strcpy(s,a);

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

/\*3、设基类定义如下：

class Basic//基类

{protected:

double r;

public:

Basic(){r=0;}

Basic(double a):r(a){}

};

编写程序从基类派生圆柱，设计成员函数输出它们的面积和体积。\*/

#include<iostream>

#define PI 3.14

using namespace std;

class Basic{

protected:

double r;

public:

Basic(){

r=0;

}

Basic(double a):r(a){

}

};

class Cylinder : public Basic{ //从基类派生圆柱类

private:

double h; //圆柱高

public:

Cylinder(double a,double b){

h=a;

r=b;

}

double getArea(){ //返回圆柱面积

return r\*r\*PI\*2+2\*r\*PI;

}

double getBulk(){ //返回圆柱体积

return r\*r\*PI\*h;

}

};

int main()

{

Cylinder cylinder(2.0,2.0);

cout<<"圆柱的面积："<<cylinder.getArea()<<endl;

cout<<"圆柱的体积："<<cylinder.getBulk()<<endl;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

2018年

//1、一个整数，它加上100后是一个完全平方数，再加上168又是一个完全平方数，

请问该数是多少？（数字小于100000）\*/

#include<stdio.h>

#include<math.h>

int main()

{

int i,j;

int flag;

for(i=1;i<100000;i++)

{

flag = 0;

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

{

if (i+100==j\*j)

{

flag = 1;

}

if (flag==1)

{

if (i+168==j\*j)

{

printf("%d",i);

}

}

}

}

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

//2、用递归法将一个整数转换成字符串形式输出。

#include <stdio.h>

char r[20];

int i=0;

void f(int n){

if (n!=0){

r[i++]=n%10+'0';

n/=10;

f(n);

}

return;

}

int main(void) {

int x,y;

printf("please input a integer:\n");

scanf("%d",&x);

f(x);

for (y=i-1;y>=0;y--){

putchar(r[y]);

}

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

//3、计算n！是一个非常难的任务，事实上n=20时，计算机的长整型就

//存放不了这么大的数了，但是，我们可以编写程序来算出n！的最后一位非0的数。

//20 ,19 ,2432902008176640000

#include<stdio.h>

int main()

{

int i;

int n;

int sum=1;

printf("please input a number:\n");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

sum \*= i;

while(sum%10==0)

{

sum /= 10;

}

if(sum>=100) //这里若设置太小，当阶乘很大时，会出现错误

{

sum %= 100;

}

// printf("%d:%d\n",i,sum);

}

printf("%d",sum%10);

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

2019年

//1、编写程序实现 s=1+2+3+......+n,n<10000; n 由用户输入；输出 s ；要

求运行结果正确，运行时间尽可能短。

#include<stdio.h>

int main()

{

int i,n,s=0;

printf("请输入一个小于10000的正整数：\n");

scanf("%d",&n);

for(i=1;i<n;i++)

{

s += i;

}

printf("s=%d",s);

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

/\*2、假设黑白图像由2维矩阵表示，行数为 M<5，列数为N<5，值分别用0或1表示。

输入两幅相同大小的黑白图像表示，求它们的相似度。

说明：若两幅图像再相同位置上的像素点颜色相同，则称它们在该位置具有

相同的像素点。两幅图像的相似度定义为相同像素点数占总像素点数的百分比。\*/

#include<stdio.h>

#define M 3 //先默认行和列都为3

#define N 3

void input(int x[M][N]);

int main()

{

int i,j;

int count=0,sum;

sum = M\*N;

int a[M][N],b[M][N];

printf("请输入黑白图像的像素点0或1：\n");

input(a);

input(b);

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

{

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

{

if (a[i][j]==b[i][j])

{

count++;

}

}

}

printf("相似度为：%0.2f%/%%",(float)count/sum\*100);

return 0;

}

//输入图像的像素点

void input(int x[M][N])

{

int i,j;

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

{

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

{

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

}

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

/\*3、Pell 数列a1,a2,a3 的定义是这样的。

a1=1; a2=2;......; a(n)=2×a(n-1)+a(n-2) n>2

给出一个正整数k（k<=50，保证a(k)在整型表示范围内），要求Pell数列的第k项

模上32767是多少。\*/

#include<stdio.h>

int Rec(int x);

int main()

{

int k;

int sum=0;

printf("请输入一个正小于等于50的正整数k：\n");

scanf("%d",&k);

sum = Rec(k);

printf("Pell数列的第k项模上32767为：%d",sum%32767);

return 0;

}

int Rec(int x)

{

if (x==1){

return 1;

}else if (x==2){

return 2;

}else {

return 2\*Rec(x-1)%32767+Rec(x-2)%32767;//每次取膜，防止越界

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

/\*4、有函数：

f(x)=x^5-15×x^4+85×x^3-225×x^2+274×x-121

已知f（1.5）>0，f（2.4）<0 且方程f（x）=0 在区间 [ 1.5, 2.4] 有且只有一

个根，请用二分法求出该根。\*/

#include<stdio.h>

#include<math.h>

#define E 1e-6

double Func(double x)

{

return pow(x,5)-15\*pow(x,4)+85\*pow(x,3)-225\*pow(x,2)+274\*x-121;

}

int main()

{

double left=1.5,right=2.4,mid;

do{

mid = (left+right)/2;

if (Func(mid)<0){

right = mid;

}else{

left = mid;

}

}while ((right-left)>E);

printf("该方程根为：%f",mid);

return 0;

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

/\*5、Julius Caesar曾经使用过一种很简单的密码。对于明文中的每个字符，

将它用它字母表中后n（n>0) 位对应的字符来代替，这样就得到了密文。

比如字符A用F来代替。如下是n=5 时密文和明文中字符的对应关系。

密文

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

明文

V W X Y Z A B C D E F G H I J K L M N O P Q R S T U

输入n（0<n<26）及一段密文（长度<200），输出解密得到明文。需要注意的是，

密文中出现的字母都是大写字母。密文中也包括非字母的字符和空格，对这些字符

不用进行解码。\*/

#include<stdio.h>

#include<string.h>

#define N 200

int main()

{

int i,n,len;

char str[N];

printf("请输入一段密文：\n");

gets(str);

len = strlen(str);

printf("请输入n：\n");

scanf("%d",&n);

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

{

if (str[i]>='A'&&str[i]<='Z')

{

str[i] += n;

}

}

printf("解密后明文为：");

puts(str);

return 0;

}

————————————————

版权声明：本文为CSDN博主「sky丶Mamba」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_42535394/article/details/104925501