编程题参考答案（**7-9** 周）

第七周**\_**在线测试

**1.** 递归法计算游戏人员的年龄

#include <stdio.h>

unsigned int ComputeAge(unsigned int n);

int main(void)

{

unsigned int n;

scanf("%u",&n);

printf("The person's age is %u\n", ComputeAge(n));

return 0;

}

// 函数功能：用递归算法计算第 n 个人的年龄

unsigned int ComputeAge(unsigned int n)

{

unsigned int age;

if (n == 1)

{

age = 10;

}

else

{

age = ComputeAge(n - 1) + 2;

}

return age;

}

**2.** 魔术师猜数

#include <stdio.h>

int Magic(int m);

int main()

{

int m, ret;

scanf("%d", &m);

ret = Magic(m);

if (ret != 1)

{

printf("The sum you calculated is wrong!\n");

}

return 0;

}

int Magic(int m)

{

int a, b, c, n;

for (a = 1; a < 9; a++)

{

for (b = 1; b < 9; b++)

{

for (c = 1; c < 9; c++)

{

n = 122 \* a + 212 \* b + 221 \* c;

if (m == n)

{

printf("The number is %d\n", 100 \* a + 10 \* b + c);

return 1;

}

}

}

}

return 0;

}

**3.** 寻找中位数 **v1.0**

#include <stdio.h>

#include <stdlib.h>

int mid(int a, int b, int c);

int main()

{

int a,b,c;

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

printf("The result is %d\n",mid(a,b,c));

return 0;

}

int mid(int a, int b, int c)

{

int tmp1=a,tmp2,tmp3,tmp4;

if (tmp1>b)

{

tmp1=b;

tmp2=a;

tmp3=c;

if (tmp1>c)

{

tmp1=c;

tmp3=b;

}

if (tmp2>tmp3)

{

tmp4=tmp3;

tmp3=tmp2;

tmp2=tmp4;

}

}

else

{

tmp2=b;

tmp3=c;

if (tmp1>c)

{

tmp1=c;

tmp3=a;

}

if (tmp2>tmp3)

{

tmp4=tmp3;

tmp3=tmp2;

tmp2=tmp4;

}

}

return tmp2;

}

**4.** 还原算术表达式

#include <stdio.h>

#include <stdlib.h>

int main()

{

int x,y,z,n,flag=0;

printf("Input n(n<1000):\n");

scanf("%d",&n);

for (x=0;x<=5;x++)

{

for (y=0;y<=5;y++)

{

for(z=0;z<=9;z++)

{

if (100\*x+10\*y+z+z+10\*z+100\*y==n)

{

printf("X=%d,Y=%d,Z=%d\n",x,y,z);

flag=1;

break;

}

}

}

}

if(flag==0) printf("Invalid\n");

return 0;

}

第七周**\_**练兵区

**1.** 谐均值计算

#include<stdio.h>

double Calculate(double ,double );

int main(void)

{

double a,b;

printf("Input two doubles:\n");

scanf("%lf%lf",&a,&b);

printf("1/((1/x+1/y)/2) = %0.3f\n",Calculate(a,b));

return 0;

}

double Calculate(double x,double y)

{

return 1/( (1/x+1/y) / 2 );

}

**2.** 输出指定行列数的字符

#include<stdio.h>

void Chline(char ch, int column, int row);

int main(void)

{

int x,y;

char ch;

printf("input a char:\n");

scanf("%c",&ch);

printf("input column and row:\n");

scanf("%d%d",&x,&y);

Chline(ch,x,y);

return 0;

}

void Chline(char ch, int column, int row)

{

int x,y;

for(y=0; y<row; y++)

{

for(x=0; x<column; x++)

printf("%c",ch);

printf("\n");

}

}

第八周**\_**在线测试

**1.** 计算礼炮声响次数

#include <stdio.h>

int main()

{

int t,n=0;

for (t=0;t<=140;t++)

{

if (t%5==0 && t<=21\*5)

{

n++;

continue;

}

if (t%6==0 && t<=21\*6)

{

n++;

continue;

}

}

if (t%7==0 && t<=21\*7)

{

n++;

continue;

}

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

return 0;

}

**2.** 兔子生崽问题

#include <stdio.h>

#define N 12

int Fibonacci(int f[], int n);

int main()

{

}

int f[N], i,num,sum;

printf("Input n(n<=12):\n");

scanf("%d",&num);

sum=Fibonacci(f, num);

printf("\nTotal=%d\n", sum);

return 0;

/\* 函数功能：计算并打印 Fibonacci 数列的前 n 项 \*/

int Fibonacci(int f[], int n)

{

int i,sum=0;

f[0] = 1;

f[1] = 2;

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

{

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

}

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

{

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

}

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

{

sum=sum+f[i];

}

return sum;

}

**3.** 抓交通肇事犯

#include <stdio.h>

int main()

{

int a,b,c,d,i,j;

for (i=31;i<100;i++)

{

j=i\*i;

a=j/1000;

d=j%10;

b=j%1000/100;

c=j%100/10;

if (a==b&&c==d&&a!=c)

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

}

return 0;

}

4. 检验并打印幻方矩阵

#include <stdio.h>

#define

N 5

int main(void)

{

int i, j;

int x[N][N];

for (i=0;i<=4;i++)

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

{

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

}

int rowSum[N], colSum[N], diagSum1, diagSum2;

int flag = 1;

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

{

rowSum[i] = 0;

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

{

rowSum[i] = rowSum[i] + x[i][j];

}

}

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

{

colSum[j] = 0;

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

{

colSum[j] = colSum[j] + x[i][j];

}

}

diagSum1 = 0;

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

{

diagSum1 = diagSum1 + x[j][j];

}

diagSum2 = 0;

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

{

diagSum2 = diagSum2 + x[j][N - 1 - j];

}

if (diagSum1 != diagSum2)

{

flag = 0;

}

else

{

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

{

if ((rowSum[i] != diagSum1) || (colSum[i] != diagSum1))

flag = 0;

}

}

if (flag)

{

printf("It is a magic square!\n");

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

{

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

{

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

}

printf("\n");

}

}

else

{

printf("It is not a magic square!\n");

}

return 0;

}

第八周**\_**练兵区

**1.** 三天打渔两天晒网

#include <stdio.h>

#include <stdlib.h>

int main()

{

int y, m, d;

int sum = 0;

scanf("%4d-%2d-%2d", &y, &m, &d);

if (y < 1990)

{

printf("Invalid input.");

return 0;

}

if ((y % 100 != 0 && y % 4 == 0) || (y % 400 == 0))

{

switch (m)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

if (d > 31 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

case 2:

if (d > 29 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

case 4:

case 6:

case 9:

case 11:

if (d > 30 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

default:

printf("Invalid input");

return 0;

break;

}

}

else

{

switch (m)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

if (d > 31 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

case 2:

if (d > 28 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

case 4:

case 6:

case 9:

case 11:

if (d > 30 || d < 1)

{

printf("Invalid input");

return 0;

}

break;

default:

printf("Invalid input");

return 0;

break;

}

}

int i;

if (y > 1990)

{

for (i = 1990; i < y; i++)

{

if ((i % 100 != 0 && i % 4 == 0) || (i % 400 == 0))

{

sum = sum + 366;

}

else

{

sum = sum + 365;

}

}

}

int i1;

if (m > 1)

{

for (i1 = 1; i1 < m; i1++)

{

if ((y % 100 != 0 && y % 4 == 0) || (y % 400 == 0))

{

switch (i1)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

sum = sum + 31;

break;

case 2:

sum = sum + 29;

break;

case 4:

case 6:

case 9:

case 11:

sum = sum + 30;

break;

}

}

else

{

switch (i1)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

sum = sum + 31;

break;

case 2:

sum = sum + 28;

break;

case 4:

case 6:

case 9:

case 11:

sum = sum + 30;

break;

}

}

}

}

sum = sum + d;

sum = sum % 5;

if (sum == 0 || sum == 4)

{

printf("He is having a rest");

}

else

{

printf("He is working");

}

return 0;

}

**2.** 统计用户输入

#include<stdio.h>

int main(void)

{

int space=0,newline=0,others=0;

char a;

printf("Please input a string end by #:\n");

while((a=getchar()) != '#')

if(a == ' ') space++;

else if (a == '\n') newline++;

else others++;

printf("space: %d,newline: %d,others: %d\n",space,newline,others);

return(0);

}

**3.** 统计正整数中指定数字的个数

#include <stdio.h>

int CountDigit(int number,int digit)

{

int count = 0;

do

{

if (digit == number%10) count++;

number = number/10;

}while (number > 9);

return count;

}

int main()

{

int m, n;

printf ("Input m,n:\n");

scanf ("%d,%d",&m, &n);

printf ("%d\n",CountDigit(m,n));

return 0;

}

**4.** 玫瑰花数

#include <stdio.h>

int main()

{

int i, a, n, s, m;

for (a=1000; a<=9999; a++) //遍历所有的 4 位数

{

n = a;

for (s = 0,i=1; i<=4; i++)

{

m = n % 10;

n = n / 10;

//求出当前 n 的最低位

//将 n 缩小 10 倍，去掉最低位

s = s + m \* m \* m \* m; //计算 4 位数字的 4 次方之和

}

if (a == s)

{

//满足 4 位数等于其 4 位数字的 4 次方之和的即为玫瑰花数

printf("%d\n", a);

}

}

return 0;

}

**5.** 四位反序数

源码 1：

#include <stdio.h>

int main()

{

int i, j, k, m;

for (i=1; i<=9; i++)//千位从 1 变化到 9

{

for (j=0; j<=9; j++)//百位从 0 变化到 9

{

for (k=0; k<=9; k++)//十位从 0 变化到 9

{

for (m=1; m<=9; m++)//个位从 1 变化到 9

{

if (9\*(i\*1000+j\*100+k\*10+m) == m\*1000+k\*100+j\*10+i)

{

printf("%d\n", i\*1000+j\*100+k\*10+m);

}

}

}

}

}

return 0;

}

源码 2：

#include <stdio.h>

int main()

{

int i;

for (i=1001; i<1111; i++)//遍历所有的 9 倍之后仍为 4 位数的数

{

if (i%10\*1000+i/10%10\*100+i/100%10\*10+i/1000 == i\*9)

{

printf("%d\n",i);

}

}

return 0;

}

**6. 8** 除不尽的自然数

源码 1：

#include <stdio.h>

int main()

{

int a, find = 0;

for (a=0; !find; a++)

{

if(((a\*8+7)\*8+1)\*8+1==(34\*a+15)\*17+4)

{

printf("%d\n",(34\*a+15)\*17+4);

find = 1;

}

}

return 0;

}

源码 2：

#include <stdio.h>

int main()

{

int i, find = 0;

for (i=1992; !find; i++)

{

if (i % 8 == 1)

{

if ((i/8)%8 == 1)

{

if ((i/8/8)%8 == 7)

{

if (i%17 == 4)

{

if ((i/17)%17 == 15)

{

if ((i/17/17) == 2\*(i/8/8/8))

{

printf("%d\n", i);

find = 1;

}

}

}

}

}

}

}

return 0;

}

第九周**\_**在线测试

**1. ISBN** 识别码判断

#include <stdio.h>

int main()

{

char isbn[13];

int i, k = 0, last, p;

gets(isbn);

for (i = 0, p = 0; i < 11; i ++)

{

if (isbn[i] == '-') continue;

p ++;

k += (isbn[i] - 48) \* p;

}

last = k % 11;

if (last == 10)

last = 'X';

else

last += 48;

if (last == isbn[12])

printf("Right");

else

{

isbn[12] = last;

printf("%s", isbn);

}

return 0;

}

**2.** 摘苹果

#include <stdio.h>

int main()

{

int a[10];

int height, get = 0;

int i;

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

{

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

}

scanf("%d", &height);

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

{

if((height + 30) >= a[i])

{

get++;

}

}

printf("%d", get);

return 0;

}

**3.** 求最大素数

#include <stdio.h>

#include <math.h>

int sum\_prime(int a[], int n);

int prime(int n);

int main()

{

int i, a[10], sum;

int num;

printf("Input n(n<=500):\n");

scanf("%d",&num);

sum = sum\_prime(a, num);

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

printf("%6d", a[i]);

printf("\nsum=%d\n", sum);

return 0;

}

int sum\_prime(int a[], int n)

{

int i, j = n, sum = 0;

for (i = 0; i < 10; )

{

if (prime(j))

{

sum = sum + j ;

a[i] = j;

i++;

}

j--;

}

return sum;

}

int prime(int n)

{

int i, k;

k = (int) sqrt((double)n);

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

{

if (n % i == 0)

return 0;

}

if (i >= k + 1)

return 1;

}

**4.** 字符串逆序

#include <stdio.h>

#include <string.h>

#define N 80

void Inverse(char str[]);

int main()

{

char a[N];

printf("Input a string:\n");

gets(a);

Inverse(a);

printf("Inversed results:\n");

puts(a);

return 0;

}

void Inverse(char str[])

{

int

len, i, j;

char temp;

len = strlen(str);

for (i=0, j=len-1; i<j; i++, j--)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

第九周**\_**练兵区

**1.** 二分法求根

#include <stdio.h>

#include <math.h>

int main()

{

float x0,x1,x2,fx0,fx1,fx2;

do

{

scanf("%f,%f",&x1,&x2);

fx1=x1\*x1\*x1-x1-1;

fx2=x2\*x2\*x2-x2-1;

} while(fx1\*fx2>0);

do

{

x0=(x1+x2)/2;

fx0=x0\*x0\*x0-x0-1;

if((fx0\*fx1)<0)

{

x2=x0;

fx2=fx0;

}

else

{

x1=x0;

fx2=fx0;

}

}while(fabs((double)fx0)>=1e-6);

printf("x=%6.2f\n",x0);

return 0;

}

**2.** 矩阵转置

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a[6][6];

int i, j, n, tep;

scanf("%d", &n);

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

{

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

{

}

a[i][j] = i \* n + j + 1;

}

printf("The original matrix is:\n");

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

{

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

{

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

}

printf("\n");

}

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

{

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

{

if (i < j)

{

tep = a[i][j];

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

a[j][i] = tep;

}

}

}

printf("The changed matrix is:\n");

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

{

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

{

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

}

printf("\n");

}

return 0;

}

**3.** 程序改错

#include <stdio.h>

#define ARR\_SIZE 10

void MaxMinExchang(int \*a, int n);

int main()

{

int a[ARR\_SIZE], i, n;

printf("Input n(n<=10):\n");

scanf("%d", &n);

printf("Input %d Numbers:\n", n);

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

{

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

}

MaxMinExchang(a, n);

printf("After MaxMinExchange:\n");

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

{

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

}

printf("\n");

return 0;

}

void MaxMinExchang(int \*a, int n)

{

int maxValue = a[0], minValue = a[0], maxPos=0, minPos=0;

int i, temp;

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

{

if (a[i] > maxValue)

{

maxValue = a[i];

maxPos = i;

}

if (a[i] < minValue)

{

minValue = a[i];

minPos = i;

}

}

temp = a[maxPos];

a[maxPos] = a[minPos];

a[minPos] = temp;

}

**4.** 计算三位阶乘和数

#include<stdio.h>

long jc(int x);

int main()

{

int a,b,c,m,n;

for(a=1;a<=9;a++)

for(b=0;b<=9;b++)

for(c=0;c<=9;c++)

{

m=a\*100+b\*10+c;

n=jc(a)+jc(b)+jc(c);

if(m==n)

{

printf("%d\n",m);

}

}

return 0;

}

long jc(int x)

{

int i;

long p=1;

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

p\*=i;

return p;

}