

# 计算概论（C语言）习题课讲义07

## 内容概要

- 习题讲解
- 数组初步
- 课堂练习

## 习题讲解

### 输出一定范围勾股数

代码点评一:

```
#include <stdio.h>

void output(int min, int max)
{
    int i,j,k;
    //简单粗暴,三重循环;
    for(i=min;i<=max;i++)
    {
        for(j=i;j<=max;j++)
        {
            for(k=j;k<=max;k++)
            {
                if(i*i+j*j==k*k)
                {
                    printf("%d %d %d\n",i,j,k);
                }
            }
        }
    }
}

int main()
{
    int min,max;
    scanf("%d%d", &min, &max);
    output(min,max);
    return 0;
}
```

代码点评二:

```
#include <stdio.h>
#include <math.h>
int perfect(int s){
```

```

int n;
for(n=1;n*n<=s;n=n+1)
    if(n*n==s) return 1;
return 0;
}
int ssqrt(int s){
int n;
for(n=1;n*n<=s;n=n+1)
    if(n*n==s) return n;
return 0;
}

void haha(int m,int n){
int s; int a;int b;
for(s=m;s<n;s=s+1)
{for(a=m+1;a<=n;a=a+1)
    if(perfect(a*a-s*s)&(s*s<a*a/2))
        printf("%d %d %d\n",s,ssqrt(a*a-s*s),a);
}
}

int main(){
int min,max;
scanf("%d%d",&min,&max);
haha(min,max);

}

```

## Collatz函数

两个注意点:

- 输入的处理
- 溢出的判断

代码点评一:

```

#include <stdio.h>
#include <limits.h>

int Collatz(int x)
{
    int i=0;
    double tmp=(INT_MAX-1)/3.0;
    for(;i<10000;i++)
    {
        printf("%d\n",x);
        if(x==1)
        {
            break;
        }
        if(x%2==1)

```

```

        {
            if(x>tmp)
            {
                printf("Error:overflow!\n");
                return 1;
            }
            x=x*3+1;
        }
        else
        {
            x=x/2;
        }
    }
    printf("Recursive steps:%d\n\n",i);
    return 0;
}

int main()
{
    int x;
    //while loop
    while(scanf("%d",&x)==1)
    {
        Collatz(x);
    }
    return 0;
}

```

代码点评二:

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void collaze(int n){
    int a=0,b,c,d;
    if(n==1){
        printf("1\nRecursive steps:0\n\n");
    }
    else{
        for(;;){
            if(n>=2&&n<=2147000000){
                printf("%d\n",n);
                if(n%2==0) n=n/2;
                else n=3*n+1;
                a=a+1;
            }
            else if(n==1){
                printf("1\nRecursive steps:%d\n\n",a);
                break;
            }
            else{
                printf("Error:overflow!\n\n");
                break;
            }
        }
    }
}

```

```

    }
}
}

int main(){
    int n;
    while (scanf ("%d",&n) !=EOF) {
        collaze(n);
    }

    return 0;
}

```

代码点评三:

```

#include <stdio.h>
#include <math.h>
void Collatz(int n){
    int steps=0;
    printf ("%d\n",n);
    label:
    if(n==1)
        printf("Recursive steps:%d\n\n", steps);
    else if ((n>715827882) && (n%2)) printf("Error:overflow!\n\n");
    else if ((n%2) && (steps<10000))
        {++steps;
        n=3*n+1;
        printf ("%d\n",n);

        goto label;}
    else if (steps<10000)
        {++steps;
        n=n/2;
        printf ("%d\n",n);

        goto label;}

}

int main(){
    int a;
    while (scanf ("%d",&a) ==1) {
        Collatz(a) ;
    }
}

```

代码点评四:

```

#include <stdio.h>

int s=0,m=0;

```

```

void Collatz(int n)
{
    if (n==1) printf("1\nRecursive steps:%d\n\n",s);
    else if(s==10001) printf("1\nRecursive steps:%d\n\n",s-1);
    else if (n%2==0)
    {
        s++;
        printf("%d\n",n);
        Collatz(n/2);
    }
    else if (n%2==1)
    {
        s++;
        m=n*3+1;
        printf("%d\n",n);
        if (m<0) printf("Error:overflow!\n\n");
        else Collatz(n*3+1);
    }
}

int main()
{
    int n;
    while(scanf("%d",&n)==1)
    {
        Collatz(n);
    }
    return 0;
}

```

## 圆周率

代码点评一:

```

#include<stdio.h>
#include<stdlib.h>
int main() {
    int b;
    double x,y;
    for(int i=1;i<=1000000;i++){
        x=rand()/(double)RAND_MAX;
        y=rand()/(double)RAND_MAX;
        if(x*x+y*y<1)b++;
    }
    printf("%.2f\n",4.0*b/1000000);
    return 0;
}

```

代码点评二:

```

#include <stdio.h>

```

```

#include <stdlib.h>

double pi()
{
    int Max=1000000;
    int N=0;
    for(int i=0;i<Max;i++)
    {
        double x=(double)rand()/RAND_MAX;
        double y=(double)rand()/RAND_MAX;
        if(x*x+y*y<=1)
        {
            N=N+1;
        }
    }
    double res=(double)N/Max;
    return res*4.0;
}

int main()
{
    printf("%.2f\n",pi());
    return 0;
}

```

代码点评三:

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>

int main(){
    int c;
    double a,b,d=0,e=0;
    time_t h;
    srand((int)time(&h));

    for(c=1;c<=1000000;++c){
        a=1.0*rand()/RAND_MAX;
        b=1.0*rand()/RAND_MAX;
        if(a*a+b*b<1){
            d=d+1.0;
            e=e+1.0;
        }
        else e=e+1.0;
    }
    printf("%.2f",d*4/1000000.0);
    return 0;
}

```

## 数组初步

C语言支持数组这种数据结构，它可以存储一个固定大小的相同类型元素的顺序集合。

几个要点:

1. 固定大小; 区别于链表结构. 不可变长或缩短,使得数组在应用上不是十分灵活.
2. 相同类型; 区别于结构类型.
3. 顺序集合; 区别于链表结构.

## 数组声明

```
type arrayName [ arraySize ];
```

其中, arraySize必须是一个大于零的整数常量, type可以是任意有效的C数据类型。

如下定义数组的方式, 可以么? C89 vs C99;

```
int n;  
int a[n];
```

## 初始化数组

```
//初始化  
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};  
  
// 非初始化, 错误  
double balance[5];  
balance[5]={1000.0, 2.0, 3.4, 7.0, 50.0};
```

## 数组元素的使用

```
int a[10];  
a[0]  
a[1]  
...  
a[n-1]
```

下标范围为0到n-1,n为数组长度. 小心数组越界问题.

## 课堂练习

- (P143 15) 借用数组, 探索 `rand() % n` 方式产生的随机数, 是均匀分布么?
- (P142 5) 借助数组, 使用字符A拼出一个大A.
- (P174 1) 写一个程序, 输出杨辉三角的前n行.
- 约瑟夫问题/猴子选王: 一堆猴子都有编号, 编号是1, 2, 3 ... m, 这群猴子 (m个) 按照1-m的顺序围坐一圈, 从第1开始数, 每数到第N个, 该猴子就要离开此圈, 这样依次下来, 直到圈中只剩下最后一只猴子, 则该猴子为大王. 求m=18, n=6时的猴王编号.