

C语言第一次作业

填空题

没啥意义，直接填一个上海大学就好

程序题

1 Hello, World!

```
#include<stdio.h>
int main()
{
    printf("Hello,World!\n");
    return 0;
}
```

很简单的C程序，怎么说呢，入门必打的程序吧。

注意下面几个点：

- 英文状态下的标点符号
- 单词不要打错了

2 鸡兔同笼

这道题两种写法

1. 常规循环遍历
2. 推导公式直接求解

常规循环遍历

```
#include<stdio.h>
int main()
{
    int a,b,ji;
    scanf("%d %d",&a,&b);
    for(ji=1;ji<=a;ji++){
        if(2*ji+4*(a-ji)==b){
            printf("chicken=%d,rabbit=%d\n",ji,a-ji);
            return 0;
        }
        //在判断成功后直接结束程序，如果始终没有判断成功
    }
}
```

```

        //则输出, Erro input!!!
    }
}
printf("Erro input\n");
return 0;
}

```

就正常想过oj题的话，没有特殊要求是没有必要这样写的

推导公式直接求解

其实我们很容易得到一个公式：

鸡数 = $(4 * \text{头数} - \text{腿数}) / (4 - 2)$ 和 **兔数 = $(\text{腿数} - 2 * \text{头数}) / (2)$**

推导过程：

- 假设全是兔子，腿数 = $4 * \text{头数}$ ，但是腿不够，少了，少了的就是我们把鸡的腿算成了4条，每只鸡多算2条腿，则 **鸡数 = $(4 * \text{头数} - \text{腿数}) / (4 - 2)$**
- 假设全是鸡，腿数 = $2 * \text{头数}$ ，但是腿多了，多的是我们把兔子的腿算成了2条，每一只兔少算2条腿，则 **兔数 = $(\text{腿数} - 2 * \text{头数}) / (2)$**

```

#include<stdio.h>
int main()
{
    int a,b;
    scanf("%d %d",&a,&b);
    printf("chicken=%d,rabbit=%d\n",(4*a-b)/2,a-(4*a-b)/2);
    return 0;
}

```

3 圆面积计算

这里就要用到小数运算，在C语言中用到小数存储最简单的方式就是 **float类型** 和 **double类型**，由题目可知，我们只需要运算2位小数内容，使用 **float** 类型就好了。

```

#include<stdio.h>
#define PI 3.1415926//这里使用宏定义来定义PI的值
int main()
{
    float r;
    printf("请输入圆的半径：");
    scanf("%f",&r);
    printf("圆的面积为：%.2f\n",PI*r*r);
    return 0;
}

```

这里用到了大家没有学习过的宏定义，其实简单的可以写成下面两种形式：

```
#include<stdio.h>
int main()
{
    float r,PI=3.1415926;//这里直接用一个新的变量来存储PI值
    printf("请输入圆的半径: ");
    scanf("%f",&r);
    printf("圆的面积为: %.2f\n",PI*r*r);
    return 0;
}
```

其实我们也可以不存这个值，直接用：

```
#include<stdio.h>
int main()
{
    float r;
    printf("请输入圆的半径: ");
    scanf("%f",&r);
    printf("圆的面积为: %.2f\n",3.1415926*r*r);//直接使用值
    return 0;
}
```

4 getchar()与putchar()使用

这个完全就是两个函数使用方法练习，区分 `getchar()` 是赋值语句赋值给变量，`putchar()` 是把变量当作参数输出出去。

这里涉及到一个新的变量类型 `char` 字符类型。

```
#include<stdio.h>
int main()
{
    char a;
    a=getchar();
    putchar(a);
    return 0;
}
```

5 N的阶乘的计算

一般来讲常用的是两种实现方式：

1. 使用 `for` 循环依次累乘
2. 通过递归算法进行

对于 `n` 的阶乘，很快就会超过 `int` 的取值范围，当然 `long long` 同样不够用，只是有所缓解，对应于 `long long` 的标准输出码是 `%lld`

使用for循环

其实这里直接放在主函数里面就ok,但是题目要求使用函数,所以:

```
#include<stdio.h>
long long Fun(int n)
{
    long long sum=1;
    for(int i=1;i<=n;i++)
        sum*=i;
    return sum;
}

int main()
{
    int n;
    scanf("%d",&n);
    printf("%lld\n",Fun(n));
    return 0;
}
```

使用递归思想进行

```
#include<stdio.h>
long long Fun(int n)
{
    if(n==1||n==0)
        return 1;
    else
        return n*Fun(n-1); //这里是递归思想, 自己调用自己的思想体现
}

int main()
{
    int n;
    scanf("%d",&n);
    printf("%lld\n",Fun(n));
    return 0;
}
```

改错题

1

原题:

```
#include<stdio.h>
int mian()
{
    printf(Welcome !\n")
    return 0;
}
```

修改:

```
#include<stdio.h>
int main()
{
    printf("Welcome !\n");
    return 0;
}
```

2

原题:

```
#include <stdio.h>
int mian()
{
    printf("*****\n");
    printf("Welcome\n")
    printf("*****\n");
    return 0;
}
```

修改:

```
#include <stdio.h>
int main()
{
    printf("*****\n");
    printf("Welcome\n");
    printf("*****\n");
    return 0;
}
```

3

原题:

```
#include <stdio.h>
int main()
{
    int a,b;
    printf("sum=%d\n",a+b);
    return 0;
}
```

修改:

```
#include<stdio.h>
int main()
{
    int a,b;
    scanf("%d %d",&a,&b);
    printf("sum=%d\n",a+b);
    return 0;
}
```

4

原题:

```
#include <stdio.h >
int main()
{
    int a,b,sum;
    scanf("%d%d",a,b);
    sum=a+b;
    printf("sum=%d\n",sum);
    return 0;
}
```

修改:

```
#include<stdio.h >
int main()
{
    int a,b,sum;
    scanf("%d%d",&a,&b);
    sum=a+b;
    printf("sum=%d\n",sum);
    return 0;
}
```