

2152118 史君宝 汇编语言的第一次作业

问题一：

输出 浮点数据 的 二进制表达形式

```
1 #include <iostream>
2 #include<bitset>
3 using namespace std;
4
5 union FLO{
6     float FL;
7     char fl[4];
8 };
9
10 union DOU{
11     double DO;
12     char dou[8];
13 };
14
15 int main()
16 {
17     cout<<"输入转换的单精度数据"<<endl;
18     FLO F;
19     cin>>F.FL;
20
21     cout<<"输入转换的双精度数据"<<endl;
22
23     DOU D;
24     cin>>D.dou;
25
26     for(int i=3;i>=0;--i)
27         cout<<bitset<8>(int(F.fl[i]))<<endl;
28
29     cout<<endl;
30
31     for(int i=7;i>=0;--i)
32         cout<<bitset<8>(int(D.dou[i]))<<endl;
33
34     return 0;
35
36
37 }
```

程序结果：

"D:\桌面资料\浮点数据.exe"

输入转换的单精度数据

5.5

输入转换的双精度数据

5.5

01000000

10110000

00000000

00000000

00000000

01100110

11111111

01100000

01000000

10110000

00000000

00000000

请按任意键继续. . .

问题二：

问题：图书馆门口的台阶有 25 阶，一步一阶或者一步两阶，有多少种不同的通过方式？

思路：这是一个经典的斐波那契数列的问题，比较常用的是两种方法，一种是递归算法，还有一种是动态规划的算法。

总的来说：我们从最终结果的来看，比如第 25 步，要么是前面的 23 步走两步要么是 24 步走一步，然后问题就转换为求对应的 23 步和 24 步的问题了。

方法一：

递归算法：

```
1 #include <iostream>
2 using namespace std;
3
4 int library(int num)
5 {
6     if(num==1)
7         return 1;
8     if(num==2)
9         return 2;
10
11     return library(num-1) + library(num-2);
12 }
13
14 int main()
15 {
16     int number, sum = 0;
17     cout<<"输入图书馆的台阶数量"<<endl;
18     cin>> number;
19
20     sum = library(number);
21
22     cout<<"上台阶的所有方法个数为"<<endl;
23     cout<<sum<<endl;
24
25
26     return 0;
27 }
28 |
```

方法二：动态规划，利用数组来进行求解。

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int number, a, b, c;
7     cout<<"输入图书馆的台阶数量"<<endl;
8     cin>> number;
9
10    a=0;
11    c=b=1;
12
13    for(int i=0;i<number;++i)
14    {
15        c=a+b;
16        a=b;
17        b=c;
18    }
19
20    cout<<"上台阶的所有方法个数为"<<endl;
21    cout<<c<<endl;
22
23
24    return 0;
25 }

```

程序运行结果：

"D:\桌面资料\浮点数据.exe"

输入图书馆的台阶数量

25

上台阶的所有方法个数为

121393

请按任意键继续. . .

问题三：

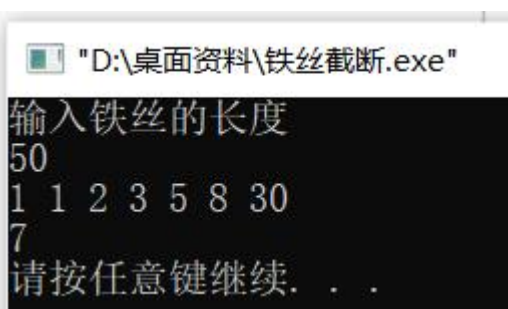
一条铁丝长度 50 米，需要把它截成一节节，每节最短 1 米，要求 任意 3 条 不能组成三角形，求出最多节数的截断方案？

思路：如何不构成一个三角形，我们可以知道，只要按照递增的顺序去截取，同时满足两边之和大于第三边就可以了，由上面我们可以知道我们可以按照斐波那契数列来进行截取，依次比较，将最后的长度加在最后一个上就可以了。

程序代码：

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout<<"输入铁丝的长度"<<endl;
7     int lon;
8     cin>>lon;
9
10    int a=0,b=1,c,number=1;
11    lon-=b;
12
13    while(lon){
14        if(lon>a+b){
15            lon-=a+b;
16            c=a+b; a=b; b=c;
17            cout<<a<<" ";
18            ++number;
19        }
20        else{
21            a=b+lon;
22            cout<<a<<endl;
23            break;
24        }
25    } |
26    cout<<number<<endl;
27    return 0;
28 }
```

运行结果：



```
"D:\桌面资料\铁丝截断.exe"
输入铁丝的长度
50
1 1 2 3 5 8 30
7
请按任意键继续...
```