

天津大学



程序设计综合实践课程报告

基础算法实验

学生姓名 陈秋澄

学院名称 智能与计算学部

专 业 大类

学 号 3022244290

1. 士兵队列训练问题

1.1 题目分析

通过循环，在人数大于 3 时，进行相关计算。

使用数组：先将数组初始化，当每次运算符合要求时，将其改为 1，重新进入循环计算。

这样，输出数组中编号为 0 的项即可满足题意。

1.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int m;
    cin>>m;
    int a[5001];
    while(m--){
        int n=0;
        cin>>n;
        memset(a,0,sizeof(a)); //数组初始化
        int num=n;
        int k=2;
        while(num>3){
            int c=0; //用于计数
            for(int i=1;i<=n;i++){
                if(!a[i]){
                    c++;
                    if(c==k){
                        c=0;
                        a[i]=1;
                        num--; //报数是 2 的人变为 1，总人数同时进行更新
                    }
                }
            }
            k=(k==2?3:2); //改变报数是 2 还是 3 的条件
        }
        for(int i=1;i<=n;i++){
```

```
        if(!a[i]&& i==1){
            cout<<"1"<<" ";//第一个人留下来
            continue;
        }
        if(!a[i])
            cout<<i<<" ";
    }
    cout<<endl;
}
return 0;
}
```

2. 斐波那契数列

2.1 题目分析

斐波那契数列的基本公式是： $f(0)=0$, $f(1)=1$, $f(x)=f(x-1)+f(x-2)$ 。

可以用循环输出答案。

2.2 题目代码

```
#include <iostream>
using namespace std;
int main(){
    long long F[100];
    F[0]=1;
    F[1]=1;
    int n,a;
    cin>>n;
    for(int i=2;i<100;i++){
        F[i]=F[i-1]+F[i-2];//递推公式
    }
    while(n--){
        cin>>a;
        cout<<F[a-1]<<endl;//输出答案
    }
    return 0;
}
```

3. [数值问题]高精度加法

3.1 题目分析

由于位数限制，可将大数用字符串储存起来。

运用数组，参考竖式计算，逐位计算。

3.2 题目代码

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;
int main(){
    char a1[5001], b1[5001];          //用字符储存数字
    int a[5001], b[5001], c[5001];    //c[i]储存每位相加的结果
    int len_a, len_b, len_c = 1, x, i;
    memset(a, 0, sizeof(a));
    memset(b, 0, sizeof(b));
    memset(c, 0, sizeof(c));
    cin>>a1>>b1;                     //输入两个加数
    len_a = strlen(a1);
    len_b = strlen(b1);
    for(i=0; i<len_a; i++) a[len_a - i] = a1[i] - '0'; //加数放入 a 数组
    for(i=0; i<len_b; i++) b[len_b - i] = b1[i] - '0'; //另一加数放 b 数组
    x = 0;                             // x 为进位
    while(len_c <= len_a || len_c <= len_b) {
        c[len_c] = a[len_c] + b[len_c] + x; //两数相加，再加上前两个数进
                                           //位的

        x = c[len_c] / 10;               //刷新进位
        c[len_c] %= 10;                  //进位后剩下的
        len_c++;                          //位数+1
    }
    c[len_c] = x;
    if(c[len_c] == 0) {                  //判断首位是否为 0
        len_c--;                          //不输出此位
    }
    for(int i=len_c; i>=1; i--) {
        cout<<c[i];                      //输出每一位的数
    }
}
```

```
    return 0;  
}
```

4. 三角形个数

4.1 题目分析

根据三角形两边之和大于第三边，逐个找出符合条件的三角形个数。

4.2 题目代码

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    int c=0;
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            for(int k=j+1;k<n;k++){
                if(a[i]+a[j]>a[k]&& a[k]+a[j]>a[i]&& a[i]+a[k]>a[j]){
                    c++;    //三角形两边之和大于第三边
                           //个数加 1
                }
            }
        }
    }
    cout<<c<<endl;
}
```

5. 找零

5.1 题目分析

从大面值至小面值依次寻找，每找完一次就求出剩余金额。

若剩余为 0，则可支付；反之，则不能支付。

5.2 题目代码

```
#include <iostream>
using namespace std;
int main(){
    int d[7]={1,2,5,10,20,50,100};    //纸币面值
    int n;
    int c=0;
    int s[n];
    cin>>n;
    for(int i=0;i<n;i++){
        int N,a[7];                    // a[7]表示纸币数量
        int num=0;
        cin>>N;
        for(int j=0;j<7;j++){
            cin>>a[j];
        }    //每种纸币数量
        for(int j=6;j>=0;j--){        //从大至小寻找
            int x=N/d[j];
            if(N>=d[j])
                N-=d[j]*min(a[j],x);
            num+=min(a[j],x);    //找出剩余金额
        }
        if(N==0){
            s[c]=num;
        }else
        {
            s[c]=-1;
        }
        c++;
    }
    for(int i=0;i<c;i++){
        cout<<s[i]<<endl;        //输出答案
    }
```



```
}  
}
```

6. 汉诺塔

6.1 题目分析

经过分析，输出结果即为 $2^n - 1$ 。

可以通过循环，进行 2 的 n 次方的计算，之后再减 1，就能得到结果。

6.2 题目代码

```
#include <iostream>
using namespace std;
int main(){
    int n;
    cin>>n;

    while(n--){
        long long a;
        cin>>a;
        long long m=1;
        for(int i=0;i<a;i++){
            m=m*2;

        }           //计算 2 的 n 次方
        m-=1;       // 计算 2 的 n 次方减 1
        cout<<m<<endl;
    }
    return 0;
}
```

7. 绝对值排序

7.1 题目分析

首先，判断每组输入的第一个数是否为 0：若为 0，退出循环。
然后，逐项进行比较、交换操作，以求出答案。

7.2 题目代码

```
#include<iostream>
#include<cmath>
using namespace std;
int a[101],temp;
int b[100];
int main(){
    cin >> b[0];
    while (b[0]!=0) {        //进入循环条件
        for(int i = 0; i <b[0]; i++)
            cin >> a[i];
        for (int i = 0; i <b[0]; i++) {

            for (int j = i ; j <b[0]; j++) {
                if (fabs(a[i]) > fabs(a[j])) {
                    swap(a[j],a[i]);    //进行比较并交换
                }
            }
        }
        for (int i =b[0]-1; i >=0; i--)
            cout << a[i] << " ";        //输出答案
        cout << endl;
        cin >>b[0];
    }
    return 0;
}
```