**XXXX 大 学**

**计算机与信息 学院实验报告**

XXXXXX 学年 XXX 学期

课程名称： 算法设计与分析

指导教师： XXXX

学号姓名： XXXXXXX

学 院： 计算机与信息学院

专 业： 计算机科学与技术

年级班级： XXXX 级 XX 班

**学生实验守则**

一、应按照课程教学计划安排，准时上课，不得迟到早退。

二、实验前应认真阅读实验指导书，明确实验目的、原理、步骤，预习有关理论知识。

三、遵守实验室规章制度，不得喧哗和打闹，不准抽烟、随地吐痰和乱丢纸屑等杂物；有净化要求的实验室，进室须换鞋。

四、严格遵守仪器设备操作规程，爱护仪器设备，节约使用材料，服从实验教师和技术人员的指导。未经许可，不得动用与实验无关的仪器设备及其他物品。

五、实验中不得擅自离开操作岗位，应仔细观察，认真记录，不得抄袭他人实验数据。

六、实验时须注意安全，杜绝发生人为安全事故。遇到问题，应及时向指导教师报告。

七、实验完毕，应清理实验现场。经指导教师检查仪器设备和实验记录后，方可离开。

八、实验后应认真完成实验报告，包括数据处理、图表绘制、结果分析等。实验报告应在规定时间内交给指导教师批阅。

九、实验过程中，因不慎造成实验设备损坏，应填写仪器损坏登记表，由指导教师根据情况进行处理。

十、因违反操作规程、擅自动用与实验无关的仪器设备、私自拆卸实验设备等不当行为，造成损失和事故的，须做出书面检查，视情节轻重和认错态度，按规定予以赔偿。

十一、进入开放实验室做实验时，应遵守《XXXXXXX》（校教字〔2018〕6号）。

目 录

|  |  |  |
| --- | --- | --- |
| **序 号** | **实验名称** | **页码** |
| 1 | 模拟与枚举 |  |
| 2 | 递归与分治（一） |  |
| 3 | 递归与分治（二） |  |
| 4 | 贪心算法 |  |
| 5 | 动态规划 |  |
| 6 | 搜索算法 |  |

# 《算法设计与分析》实验报告1

实验名称：  **模拟与枚举**

实验日期： 2021年3月15日-3月28日

实验地点： A608

1. **实验目的**

**二、实验内容（编程题）**

Title:A-与7无关的数Description: 一个正整数,如果它能被7整除,或者它的十进制表示法中某一位上的数字为7,则称其为与7相关的数.现求所有小于等于n(n < 100)的与7无关的正整数的平方和. Input: 输入为一行,正整数n(n < 100) Output: 输出一行，包含一个整数，即小于等于n的所有与7无关的正整数的平方和。 SampleInput: 21 SampleOutput: 2336 Title:B-细菌繁殖Description: 一种细菌的繁殖速度是每天成倍增长。例如：第一天有10个，第二天就变成20个，第三天变成40个，第四天变成80个，……。现在给出第一天的日期和细菌数目，要你写程序求出到某一天的时候，细菌的数目。Input: 第一行有一个整数n，表示测试数据的数目。其后n行每行有5个整数，整数之间用一个空格隔开。第一个数表示第一天的月份，第二个数表示第一天的日期，第三个数表示第一天细菌的数目，第四个数表示要求的那一天的月份，第五个数表示要求的那一天的日期。已知第一天和要求的一天在同一年并且该年不是闰年，要求的一天一定在第一天之后。数据保证要求的一天的细菌数目在长整数（long）范围内。Output: 对于每一组测试数据，输出一行，该行包含一个整数，为要求的一天的细菌数。SampleInput: 21 1 1 1 22 28 10 3 2SampleOutput: 240Title:C-求一元二次方程的根Description: 利用公式x1 = (-b + sqrt(b\*b-4\*a\*c))/(2\*a), x2 = (-b - sqrt(b\*b-4\*a\*c))/(2\*a)求一元二次方程ax2 + bx + c =0的根，其中a不等于0。Input: 第一行是待解方程的数目n。 其余n行每行含三个浮点数a, b, c（它们之间用空格隔开），分别表示方程ax2 + bx + c =0的系数。Output: 输出共有n行，每行是一个方程的根： 若是两个实根，则输出：x1=...;x2 = ... 若两个实根相等，则输出：x1=x2=... 若是两个虚根，则输出：x1=实部+虚部i; x2=实部-虚部i 所有实数部分要求精确到小数点后5位，数字、符号之间没有空格。 x1和x2的顺序：x1的实部>Re的实部||(x1的实部==x2的实部&&x1的虚部>=x2的虚部)SampleInput: 31.0 3.0 1.02.0 -4.0 2.01.0 2.0 8.0SampleOutput: x1=-0.38197;x2=-2.61803x1=x2=1.00000x1=-1.00000+2.64575i;x2=-1.00000-2.64575iTitle:D-合唱队形Description: N位同学站成一排，音乐老师要请其中的(N-K)位同学出列，使得剩下的K位同学不交换位置就能排成合唱队形。 合唱队形是指这样的一种队形：设K位同学从左到右依次编号为1, 2, …, K，他们的身高分别为T1, T2, …, TK，则他们的身高满足T1 < T2 < … < Ti , Ti > Ti+1 > … > TK (1 <= i <= K)。 你的任务是，已知所有N位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。 Input: 输入的第一行是一个整数N（2 <= N <= 100），表示同学的总数。第一行有n个整数，用空格分隔，第i个整数Ti（130 <= Ti <= 230）是第i位同学的身高（厘米）。Output: 输出包括一行，这一行只包含一个整数，就是最少需要几位同学出列。SampleInput: 8186 186 150 200 160 130 197 220SampleOutput: 4Title:E-质数的和与积Description: 两个质数的和是S，它们的积最大是多少？ Input: 一个不大于10000的正整数S，为两个质数的和。 Output: 一个整数，为两个质数的最大乘积。数据保证有解。 SampleInput: 50 SampleOutput: 589 Title:F-不定方程求解Description: 给定正整数a，b，c。求不定方程 ax+by=c 关于未知数x和y的所有非负整数解组数。 Input: 一行，包含三个正整数a，b，c，两个整数之间用单个空格隔开。每个数均不大于1000。 Output: 一个整数，即不定方程的非负整数解组数。 SampleInput: 2 3 18 SampleOutput: 4 Title:G-砝码称重Description: 设有1g、2g、3g、5g、10g、20g的砝码各若干枚（其总重<=1000），要求：计算用这些砝码能称出的不同重量的个数，但不包括一个砝码也不用的情况。 Input: 一行，包括六个正整数a1,a2,a3,a4,a5,a6，表示1g砝码有a1个，2g砝码有a2个，……，20g砝码有a6个。相邻两个整数之间用单个空格隔开。 Output: 以“Total=N”的形式输出，其中N为可以称出的不同重量的个数。 SampleInput: 1 1 0 0 0 0 SampleOutput: Total=3 Title:H-垃圾炸弹Description:     2018年俄罗斯世界杯（2018 FIFA World Cup）开踢啦！为了方便球迷观看比赛，莫斯科街道上很多路口都放置了的直播大屏幕，但是人群散去后总会在这些路口留下一堆垃圾。为此俄罗斯政府决定动用一种最新发明——“垃圾炸弹”。这种“炸弹”利用最先进的量子物理技术，爆炸后产生的冲击波可以完全清除波及范围内的所有垃圾，并且不会产生任何其他不良影响。炸弹爆炸后冲击波是以正方形方式扩散的，炸弹威力（扩散距离）以d给出，表示可以传播d条街道。     例如下图是一个d=1的“垃圾炸弹”爆炸后的波及范围。     假设莫斯科的布局为严格的1025\*1025的网格状，由于财政问题市政府只买得起一枚“垃圾炸弹”，希望你帮他们找到合适的投放地点，使得一次清除的垃圾总量最多（假设垃圾数量可以用一个非负整数表示，并且除设置大屏幕的路口以外的地点没有垃圾）。Input: 第一行给出“炸弹”威力d(1 <= d <= 50)。第二行给出一个数组n(1 <= n <= 20)表示设置了大屏幕(有垃圾)的路口数目。接下来n行每行给出三个数字x, y, i, 分别代表路口的坐标(x, y)以及垃圾数量i. 点坐标(x, y)保证是有效的（区间在0到1024之间），同一坐标只会给出一次。Output: 输出能清理垃圾最多的投放点数目，以及能够清除的垃圾总量。SampleInput: 124 4 106 6 20SampleOutput: 1 30

**三、程序代码（可附算法）**

Title: A-与7无关的数Code: #include<stdio.h>int f(int n){ if(n % 7 == 0) return 0; else{ int a = n; while(a!=0){ int i = a % 10; if(i == 7) return 0; a = a / 10; } return 1; } }int main(){ int n; long long ans = 0; scanf("%d",&n); for(int i = 1; i <= n; i++){ if(f(i)) ans += i\*i; } printf("%lld\r
",ans); return 0;}Title: B-细菌繁殖Code: #include<stdio.h>#include<math.h>int month[] = {0,31,28,31,30,31,30,31,31,30,31,30,31};int main(){ int n; long ans; int sm,sd,num,em,ed; scanf("%d",&n); for(int i = 1; i <= n; i++){ scanf("%d %d %d %d %d",&sm,&sd,&num,&em,&ed); ans = num; int cnt = 0; for(int j = sm; j < em; j++){ cnt += month[j]; } cnt -= sd; cnt += ed; for(int index = 1; index <= cnt; index++) ans = ans \* 2; printf("%ld\r
",ans); } return 0;}Title: C-求一元二次方程的根Code: #include<stdio.h>#include<math.h>int main(){ double a,b,c; int n; scanf("%d",&n); for(int i=1;i<=n;i++){ scanf("%lf %lf %lf",&a,&b,&c); double delt = b\*b-4\*a\*c; double x1,x2; double x; if(a!=0){ if(delt>0){ x1 = (-b+sqrt(delt))/(a\*2); x2 = (-b-sqrt(delt))/(a\*2); printf("x1=%.5f;x2=%.5f\r
",x1,x2); } else if(delt < 0){ if(b!=0) { double real = (-b)\*1.0/(2\*a); double imag = sqrt(-delt)/(2\*a); printf("x1=%.5f%+.5lfi;x2=%.5f%+.5lfi\r
",real,imag,real,-imag); }else{ double x = sqrt(-delt)\*1.0/(2\*a); printf("x1=%.5f%+.5lfi;x2=%.5f%+.5lfi\r
",0.0,x,0.0,-x); } } else{ printf("x1=x2=%.5f\r
",-b\*1.0/(2\*a)); } } } return 0;}Title: D-合唱队形Code: #include<cstring>#include<iostream>#include<algorithm>using namespace std;int a[100];int dp[100];int shangsheng(int begin,int end){memset(dp, 0, 100 \* sizeof(int));int ans = 0;for (int i = begin;i < end;i++){dp[i] = 1;for (int j = 0;j < i;j++)if (a[j] < a[i])dp[i] = max(dp[j] + 1, dp[i]);ans = max(ans, dp[i]);}return ans;}int xiajiang(int begin,int end){memset(dp, 0, 100 \* sizeof(int));int ans = 0;for (int i = begin;i < end;i++){dp[i] = 1;for (int j = 0;j < i;j++)if (a[j] > a[i])dp[i] = max(dp[j] + 1, dp[i]);ans = max(ans, dp[i]);}return ans;}int main(){int n;cin >> n;int ans = 0;for (int i = 0;i < n;i++)cin >> a[i];for (int i = 0;i < n ;i++){ans = max(ans, shangsheng(0, i) + xiajiang(i, n));}cout << n - ans;return 0;}Title: E-质数的和与积Code: #include <stdio.h>#include <string.h>#define N 10000int u[N+1]; //筛子 int su[N+1],len; //素数表su，其长度为len void getPrime() //使用埃拉托斯特尼筛法构建素数表su[2..N]，时间复杂度O(NlogN) { int i,j; memset(u,1,sizeof(u)); //初始时所有素数都在筛中 for(i=2;i<=N;i++) //顺序搜索筛中的最小数 if(u[i]) { if(i>N/i) break; for(j=i\*i;j<=N;j+=i) //将i的倍数从筛子中筛去 u[j]=0; } for(i=2;i<=N;i++) //将筛中的所有素数放入su中 if(u[i]) su[++len]=i; }int main(){ getPrime(); int n,ans; scanf("%d",&n); for(int i = n/2; i >= 2; i--){ if(u[i]&&u[n-i]){ ans = i\*(n-i); break; } } printf("%d\r
",ans); return 0;}Title: F-不定方程求解Code: import java.util.Scanner;public class Main { public static void main(String[] args) { Scanner in = new Scanner(System.in); int a = in.nextInt(); int b = in.nextInt(); int c = in.nextInt(); int sum = 0; for(int i = 0; i<= 1000; i++) for(int j = 0; j <= 1000; j++){ if(i\*a + j\*b == c) sum++; } System.out.println(sum); }}Title: G-砝码称重Code: #include<stdio.h>#include<math.h>int m[1005];int main(){ int a1,a2,a3,a4,a5,a6,t,ans; scanf("%d %d %d %d %d %d",&a1,&a2,&a3,&a4,&a5,&a6); for(int a = 0; a <= a1; a++) for(int b = 0; b <= a2; b++) for(int c = 0; c <= a3; c++) for(int d = 0; d <= a4; d++) for(int e = 0; e <= a5; e++) for(int f = 0; f <= a6; f++){ t = a + 2\*b + 3\*c + 5\*d + 10\*e + 20\*f; m[t] = 1; } ans = 0; for(int i = 1; i <= 1005; i++){ if(m[i]) ans ++; } printf("Total=%d\r
",ans); return 0;}Title: H-垃圾炸弹Code: /\*遍历所有的垃圾点\*/#include<bits/stdc++.h>using namespace std;const int MAX = 1025;int a[MAX][MAX]; int px,py,pm,maxx,cnt;int main(){ int d,n; scanf("%d",&d); scanf("%d",&n); for(int i = 1; i <= n; i++){ scanf("%d %d %d",&px,&py,&pm); for(int x = px - d; x <= px + d; x++){ if(x >= 0 && x <= 1024){ for(int y = py - d; y <= py + d; y++){ if(y >= 0 && y <= 1024){ a[x][y] += pm; } } } } } cnt = 0; maxx = -1e9; for(int i = 0; i <= 1024; i++) for(int j = 0; j <= 1024; j++){ if(a[i][j] > maxx){ cnt = 1; maxx = a[i][j]; } else if(a[i][j] == maxx){ cnt++; } } printf("%d %d\r
",cnt,maxx); return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

# 《算法设计与分析》实验报告2

实验名称：  **递归与分治（一）**

实验日期： 2021年3月26日-4月12日

实验地点：  A608

**一、实验目的**

**二、实验内容（编程题）**

Title:A-菲波那契数列Description: 菲波那契数列是指这样的数列: 数列的第一个和第二个数都为1，接下来每个数都等于前面2个数之和。 给出一个正整数a，要求菲波那契数列中第a个数是多少。 Input: 第1行是测试数据的组数n，后面跟着n行输入。每组测试数据占1行，包括一个正整数a(1 <= a <= 20)Output: 输出有n行，每行输出对应一个输入。输出应是一个正整数，为菲波那契数列中第a个数的大小SampleInput: 452191SampleOutput: 5141811Title:B-最大公约数Description: 给定两个正整数，求它们的最大公约数。Input: 有多组数据，每行为两个正整数，且不超过int可以表示的范围。Output: 行对应输出最大公约数。SampleInput: 4 88 6200 300SampleOutput: 42100Title:C-Placing applesDescription:把 M 个同样的苹果放在 N 个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的放法 (用 K 表示) ？注：{5, 1, 1} 和 {1, 5, 1} 是同一种放法。Input:第一行是测试数据的数目 \*t\* (\*0 ≤ t ≤ 20\*)。以下每行均包含两个以空格分隔的整数 \*M\* 和 \*N\* (\*1 ≤ M, N ≤ 10\*)。Output:对输入的每组数据，用一行输出相应的 K。SampleInput: 17 3SampleOutput:8Title:D-简单的整数划分问题Description: 将正整数n 表示成一系列正整数之和，n=n1+n2+…+nk, 其中n1>=n2>=…>=nk>=1 ，k>=1 。正整数n 的这种表示称为正整数n 的划分。正整数n 的不同的划分个数称为正整数n 的划分数。 Input: 标准的输入包含若干组测试数据。每组测试数据是一个整数N(0 < N <= 50)。 Output: 对于每组测试数据，输出N的划分数。 SampleInput: 5 SampleOutput: 7 Title:E-分解因数Description: 给出一个正整数a，要求分解成若干个正整数的乘积，即a = a1 \* a2 \* a3 \* ... \* an，并且1 < a1 <= a2 <= a3 <= ... <= an，问这样的分解的种数有多少。注意到a = a也是一种分解。Input: 第1行是测试数据的组数n，后面跟着n行输入。每组测试数据占1行，包括一个正整数a (1 < a < 32768)Output: n行，每行输出对应一个输入。输出应是一个正整数，指明满足要求的分解的种数SampleInput: 2220SampleOutput: 14Title:F-扩号匹配问题Description: 在某个字符串（长度不超过100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用"$"标注,不能匹配的右括号用"?"标注.Input: 输入包括多组数据，每组数据一行，包含一个字符串，只包含左右括号和大小写字母，字符串长度不超过100 注意：cin.getline(str,100)最多只能输入99个字符！Output: 对每组输出数据，输出两行，第一行包含原始输入字符，第二行由"$","?"和空格组成，"$"和"?"表示与之对应的左括号和右括号不能匹配。SampleInput: ((ABCD(x))(rttyy())sss)(SampleOutput: ((ABCD(x)$$)(rttyy())sss)(? ?$Title:G-DryingDescription: 每件衣服都有一定单位水分,在不使用烘干器的情况下,每件衣服每分钟自然流失1个单位水分,但如果使用了烘干机则每分钟流失K个单位水分,但是遗憾是只有1台烘干机，每台烘干机同时只能烘干1件衣服,请问要想烘干N件衣服最少需要多长时间？ Input:多组输入。 每组第一行输入N，表示有N件衣服，第二行输入N件衣服的水分ai，第三行表示烘干机每分钟烘干水分K 其中 1 ≤ N ≤ 100 000，1 ≤ ai ≤ 10^9,1 ≤ K≤ 10^9本题数据较多，请使用scanf/printf。Output: 输出烘干N件衣服所需要的最短时间 SampleInput: 32 3 9532 3 65SampleOutput:32Title:H-AssembleDescription:该题描述出现异常，请查看网站Input:该题描述出现异常，请查看网站Output:该题描述出现异常，请查看网站SampleInput: 该题描述出现异常，请查看网站SampleOutput:该题描述出现异常，请查看网站

**三、程序代码（可附算法）**

Title: A-菲波那契数列Code: #include<iostream>using namespace std;int a[25];int f(int n){ if(n == 1 || n == 2){ return 1; } else if(a[n]){ return a[n]; } else{ int ans = f(n-1) + f(n-2); a[n] = ans; return ans; } }int main(){ int t,num; cin >> t; while(t--){ cin >> num; int ans = f(num); cout << ans << endl; } return 0;} Title: B-最大公约数Code: #include<bits/stdc++.h>using namespace std;int main(){ int x,y; while(cin>>x>>y) cout << \_\_gcd(x,y) << endl; return 0;} Title: C-Placing applesCode: #include<bits/stdc++.h>using namespace std;int f(int x,int y){ if(y == 1 || x == 0) return 1; if(x < y) return f(x,x); return f(x-y,y) + f(x,y-1);}int main(){ int t,x,y; scanf("%d",&t); for(int i = 1; i <= t; i++){ scanf("%d %d",&x,&y); printf("%d\r
",f(x,y)); } return 0;} Title: D-简单的整数划分问题Code: #include<iostream>#include<stack>#include<string>#include<cstring>#include<vector>using namespace std;int dp[105];int main(){ int n; dp[0] = 1; for(int i = 1; i <= 50; i++) for(int j = 1; j <= 50; j++){ if(j-i >= 0) dp[j] += dp[j-i]; } while(cin >> n){ cout << dp[n] << endl; } return 0;} Title: E-分解因数Code: #include<iostream> #include<cstring> #include <cstdlib>#include <algorithm>#include <cstdio>using namespace std; int n; int dfs(int nn,int m){ if(nn == 1) return 1; if(m == 1) return 0; if( nn % m == 0) return dfs(nn,m - 1) + dfs(nn / m,m); return dfs(nn,m-1); } int main(){ int t; cin >> t; while( t-- ){ cin >> n; cout << dfs(n,n) << endl; } return 0; } Title: F-扩号匹配问题Code: #include<stdio.h>#include<iostream>#include<string>#include<cstring>using namespace std; int a[110],j=0;int main(){ string s; while(cin>>s) { memset(a,0,sizeof(a)); j=0; a[j] = -1; cout<<s<<endl; for(int i=0;i<s.length();i++) { if((s[i]>='a'&&s[i]<='z')||(s[i]>='A'&&s[i]<='Z')) { s[i]=' '; } if(s[i]=='(') { a[++j]=i; s[i]='$'; } else if(s[i]==')') { if(a[j]>=0) { s[a[j]]=s[i]=' '; j--; } else s[i]='?'; } } cout<<s<<endl; } return 0;}Title: G-DryingCode: #include<cstdio>#include<iostream>#include<algorithm>#include<cmath>using namespace std;long long a[100005];int n;long long k,maxx;int check(long long mid){ long long sum = 0; for(int i = 1; i <= n; i++){ if(a[i] > mid) sum += ceil((a[i] - mid)\*1.0 / (k - 1)); } return sum <= mid;}int main(){ maxx = -1e9; scanf("%d",&n); for(int i = 1; i <= n; i++){ scanf("%lld",&a[i]); maxx = max(maxx,a[i]); } scanf("%lld",&k); if(k == 1){ printf("%d\r
",maxx); return 0; } long long l, r; l = 1; r = maxx; while(l < r){ long long mid = (l + r) / 2; if(check(mid)) r = mid; else l = mid + 1; } printf("%lld\r
",r);}Title: H-AssembleCode: #include<iostream>#include<cstdlib>#include<cstdio>#include<vector>#include<map>#define Lint long long intusing namespace std;const int INF=1e9+7;const int MAXN=1010;struct node{ Lint p,q;};int T,n,cnt;Lint b;map<string,int> id;vector<node> t[MAXN];bool check(Lint lim){ Lint ret=0,tmp; for(int i=1,x;i<=cnt;i++) { x=t[i].size()-1,tmp=INF; for(int j=0;j<=x;j++) if( t[i][j].q>=lim ) tmp=min( tmp,t[i][j].p ); ret+=tmp; } return ret<=b;}int main(){ string type,name; Lint x,y,l,r; scanf("%d",&T); while( T-- ) { l=r=cnt=0; scanf("%d%lld",&n,&b); for(int i=1;i<=n;i++) { cin>>type>>name; scanf("%lld%lld",&x,&y); if( !id[type] ) id[type]=++cnt; t[id[type]].push\_back( (node){ x,y } ); r=max( r,y ); } while( l<=r ) { int mid=(l+r)/2; if( check( mid ) ) l=mid+1; else r=mid-1; } printf("%lld\r
",l-1); id.clear(); for(int i=1;i<=cnt;i++) t[i].clear(); } return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

# 《算法设计与分析》实验报告3

实验名称：  **递归与分治（二）**

实验日期： 2021年4月11日-4月30日

实验地点： A608

**一、实验目的**

**二、实验内容（编程题）**

Title:A-查找最接近的元素Description: 在一个非降序列中，查找与给定值最接近的元素。Input: 第一行包含一个整数n，为非降序列长度。1 <= n <= 100000。 第二行包含n个整数，为非降序列各元素。所有元素的大小均在0-1,000,000,000之间。 第三行包含一个整数m，为要询问的给定值个数。1 <= m <= 10000。 接下来m行，每行一个整数，为要询问最接近元素的给定值。所有给定值的大小均在0-1,000,000,000之间。Output: m行，每行一个整数，为最接近相应给定值的元素值，保持输入顺序。若有多个值满足条件，输出最小的一个。SampleInput: 32 5 82105SampleOutput: 85Title:B-二分法求函数的零点Description: 有函数： f(x) = x5 - 15 \* x4+ 85 \* x3- 225 \* x2+ 274 \* x - 121 已知 f(1.5) > 0 , f(2.4) < 0 且方程 f(x) = 0 在区间 [1.5,2.4] 有且只有一个根，请用二分法求出该根。 Input: 无。 Output: 该方程在区间[1.5,2.4]中的根。要求四舍五入到小数点后6位。 SampleInput: 无 SampleOutput: 不提供 Title:C-和为给定数Description: 给出若干个整数，询问其中是否有一对数的和等于给定的数。Input: 共三行： 第一行是整数n(0 < n <= 100,000)，表示有n个整数。 第二行是n个整数。整数的范围是在0到10^8之间。 第三行是一个整数m（0 <= m <= 2^30)，表示需要得到的和。Output: 若存在和为m的数对，输出两个整数，小的在前，大的在后，中间用单个空格隔开。若有多个数对满足条件，选择数对中较小的数更小的。若找不到符合要求的数对，输出一行No。SampleInput: 42 5 1 46SampleOutput: 1 5Title:D-全排列Description: 对于数组[1, 2, 3]，他们按照从小到大的全排列是 1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1 现在给你一个正整数n，n小于8，输出数组[1, 2, …，n]的从小到大的全排列。 Input: 输入有多行，每行一个整数。当输入0时结束输入。Output: 对于每组输入，输出该组的全排列。每一行是一种可能的排列，共n个整数，每个整数用一个空格隔开，每行末尾没有空格。SampleInput: 230SampleOutput: 1 22 11 2 31 3 22 1 32 3 13 1 23 2 1Title:E-全排列Description: 给定一个由不同的小写字母组成的字符串，输出这个字符串的所有全排列。 我们假设对于小写字母有'a' < 'b' < ... < 'y' < 'z'，而且给定的字符串中的字母已经按照从小到大的顺序排列。 Input: 输入只有一行，是一个由不同的小写字母组成的字符串，已知字符串的长度在1到6之间。Output: 输出这个字符串的所有排列方式，每行一个排列。要求字母序比较小的排列在前面。字母序如下定义： 已知S = s1s2...sk , T = t1t2...tk，则S < T 等价于，存在p (1 <= p <= k)，使得 s1 = t1, s2 = t2, ..., sp - 1 = tp - 1, sp < tp成立。SampleInput: abcSampleOutput: abcacbbacbcacabcbaTitle:F-N皇后问题Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
在N\*N的方格棋盘放置了N个皇后，使得它们不相互攻击（即任意2个皇后不允许处在同一排，同一列，也不允许处在与棋盘边框成45角的斜线上。
你的任务是，对于给定的N，求出有多少种合法的放置方法。
Input:共有若干行，每行一个正整数N≤10，表示棋盘和皇后的数量；如果N=0，表示结束。Output:共有若干行，每行一个正整数，表示对应输入行的皇后的不同放置数量。SampleInput: 1850SampleOutput:19210Title:G-Pell数列Description: Pell数列a1, a2, a3, ...的定义是这样的，a1 = 1, a2 = 2, ... , an = 2 \* an − 1 + an - 2 (n > 2)。 给出一个正整数k，要求Pell数列的第k项模上32767是多少。 Input: 第1行是测试数据的组数n，后面跟着n行输入。每组测试数据占1行，包括一个正整数k (1 ≤ k < 1000000)。Output: n行，每行输出对应一个输入。输出应是一个非负整数。SampleInput: 218SampleOutput: 1408Title:H-Monthly ExpenseDescription:农夫约翰是一个精明的会计师。他意识到自己可能没有足够的钱来维持农场的运转了。他计算出并记录下了接下来 N (1 ⩽ N ⩽ 100,000) 天里，每天需要的支出金额 (1 ⩽ money\_i ⩽ 10,000)。农夫约翰打算为连续的 M (1 ⩽ M ⩽ N) 个财务周期建立预算，这样的财务周期被称之为 “fajo 月”。每个 fajo 月包含一天或连续的多天，每天被恰好包含在一个 fajo 月内。农夫约翰的目标是合理安排所有的 fajo 月，使得开销最高的 fojo 月的支出最少，以便判断自己的财务周期支出上限。Input:第 1 行包含以空格分隔的两个整数 N, M。第 2 到 N + 1 行，每行包含一个金额整数，第 i + 1 行表示农夫约翰在第 i 天的支出。Output:输出农夫约翰的财务周期最大支出的最小值。SampleInput: 7 5100400300100500101400SampleOutput:500Title:I-矩形分割Description: 平面上有一个大矩形，其左下角坐标（0，0），右上角坐标（R,R)。大矩形内部包含一些小矩形，小矩形都平行于坐标轴且互不重叠。所有矩形的顶点都是整点。要求画一根平行于y轴的直线x=k（k是整数) ，使得这些小矩形落在直线左边的面积必须大于等于落在右边的面积，且两边面积之差最小。并且，要使得大矩形在直线左边的的面积尽可能大。注意：若直线穿过一个小矩形，将会把它切成两个部分，分属左右两侧。Input: 第一行是整数R，表示大矩形的右上角坐标是(R,R) (1 <= R <= 1,000,000)。 接下来的一行是整数N,表示一共有N个小矩形(0 < N <= 10000)。 再接下来有N 行。每行有4个整数，L,T, W 和 H, 表示有一个小矩形的左上角坐标是(L,T),宽度是W，高度是H (0<=L,T <= R, 0 < W,H <= R). 小矩形不会有位于大矩形之外的部分。Output: 输出整数n，表示答案应该是直线 x=n。 如果必要的话，x=R也可以是答案。SampleInput: 100021 1 2 15 1 2 1SampleOutput: 5Title:J-Cable masterDescription: Inhabitants of the Wonderland have decided to hold a regional programming contest. The Judging Committee has volunteered and has promised to organize the most honest contest ever. It was decided to connect computers for the contestants using a "star" topology - i.e. connect them all to a single central hub. To organize a truly honest contest, the Head of the Judging Committee has decreed to place all contestants evenly around the hub on an equal distance from it.To buy network cables, the Judging Committee has contacted a local network solutions provider with a request to sell for them a specified number of cables with equal lengths. The Judging Committee wants the cables to be as long as possible to sit contestants as far from each other as possible.The Cable Master of the company was assigned to the task. He knows the length of each cable in the stock up to a centimeter,and he can cut them with a centimeter precision being told the length of the pieces he must cut. However, this time, the length is not known and the Cable Master is completely puzzled.You are to help the Cable Master, by writing a program that will determine the maximal possible length of a cable piece that can be cut from the cables in the stock, to get the specified number of pieces.Input: The first line of the input file contains two integer numb ers N and K, separated by a space. N (1 = N = 10000) is the number of cables in the stock, and K (1 = K = 10000) is the number of requested pieces. The first line is followed by N lines with one number per line, that specify the length of each cable in the stock in meters. All cables are at least 1 meter and at most 100 kilometers in length. All lengths in the input file are written with a centimeter precision, with exactly two digits after a decimal point.Output: Write to the output file the maximal length (in meters) of the pieces that Cable Master may cut from the cables in the stock to get the requested number of pieces. The number must be written with a centimeter precision, with exactly two digits after a decimal point. If it is not possible to cut the requested number of pieces each one being at least one centimeter long, then the output file must contain the single number "0.00" (without quotes).SampleInput: 4 118.027.434.575.39SampleOutput: 2.00

**三、程序代码（可附算法）**

Title: A-查找最接近的元素Code: #include<bits/stdc++.h>using namespace std;const int MAXN = 100000 + 5;int a[MAXN];int judge(int x, int len){ int left = 1; int right = len; while(left <= right){ int mid = (left + right) / 2; if(a[mid] > x) right = mid - 1; else if(a[mid] < x) left = mid + 1; else return x; } //处理边界情况 if(left > len) return a[right]; if(right < 1) return a[left]; //处理未越界不相等的情况 return (x - a[right]) <= (a[left] - x) ? a[right] : a[left]; }int main(){ int n; scanf("%d",&n); for(int i = 1; i <= n; i++) scanf("%d",&a[i]); int m,t; scanf("%d",&m); for(int i = 1; i <= m; i++){ scanf("%d",&t); int ans = judge(t, n); printf("%d\r
", ans); }}Title: B-二分法求函数的零点Code: #include<bits/stdc++.h>using namespace std;double f(double x){ double ans = x\*x\*x\*x\*x - 15\*x\*x\*x\*x + 85\*x\*x\*x- 225\*x\*x + 274\*x - 121; return ans > 0.0 ? 1 : 0;}int main(){ double x1, x2, mid; x1 = 1.5; x2 = 2.4; while(x2-x1 > 0.0000001){ mid = (x1 + x2) / 2; if(f(mid)) x1 = mid; else x2 = mid; } printf("%.6lf\r
", mid);}Title: C-和为给定数Code: #include<stdio.h>#include<algorithm>using namespace std;int main(){ int n,a[100005]; scanf("%d",&n); for(int i=0;i<n;i++) { scanf("%d",&a[i]); } sort(a,a+n); int y; scanf("%d",&y); int l=0,r=n-1,b=0; while(l<r) { if(a[l]+a[r]<y) { l++; } else if(a[l]+a[r]==y) { printf("%d %d",a[l],a[r]);b=1; break; } else { r--; } } if(b==0) { printf("No"); } return 0;}Title: D-全排列Code: #include<bits/stdc++.h>using namespace std;int a[18];int main(){ int n; while(~scanf("%d",&n) && n){ for(int i = 1; i <= n; i++) a[i] = i; do{ for(int i = 1; i <= n; i++) printf(i==1?"%d":" %d",a[i]); printf("\r
"); }while(next\_permutation(a+1,a+1+n)); } return 0;} Title: E-全排列Code: #include<bits/stdc++.h>using namespace std;char s[100005];int main(){ gets(s); int len = strlen(s); do{ for(int i = 0; i < len; i++) printf("%c",s[i]); printf("\r
"); }while(next\_permutation(s,s+len)); return 0;} Title: F-N皇后问题Code: #include <stdio.h>int a[10] = {1,0,0,2,10,4,40,92,352,724};int main() { int n; while (scanf("%d", &n) && n) { printf("%d\r
", a[n-1]); } return 0;}Title: G-Pell数列Code: #include<bits/stdc++.h>using namespace std;long long a[1000005];int main(){ int t,x; a[1] = 1; a[2] = 2; for(int i = 3; i <= 1000000; i++) a[i] = (2\*a[i-1] + a[i-2]) % 32767; cin >> t; while(t--){ cin >> x; cout << a[x] << endl; } }Title: H-Monthly ExpenseCode: #include <bits/stdc++.h>#include<bits/stdc++.h>#define endl '\r
'#define FOR(i, x, y) for (decay<decltype(y)>::type i = (x), \_##i = (y); i < \_##i; ++i)#define FORD(i, x, y) for (decay<decltype(x)>::type i = (x), \_##i = (y); i > \_##i; --i)using namespace std;typedef long long LL;template<typename T>inline T read() { T x; cin >> x; return x;}int main() { ios::sync\_with\_stdio(false); cin.tie(nullptr); cout.tie(nullptr); // freopen("input.txt", "r", stdin); int n = read<int>(); vector<int> a(n); int m(read<int>()); FOR(i, 0, n)a[i] = read<int>(); LL left = \*max\_element(a.begin(), a.end()), right = accumulate(a.begin(), a.end(), 0LL); LL best = right; while (left <= right) { LL mid = (left + right) / 2; LL tot = 0, s = 0; FOR(i, 0, n) { if (s + a[i] > mid) { s = a[i]; ++tot; } else { s += a[i]; } } if (s != 0) { ++tot; } if (tot > m) { left = mid + 1; } else { best = min(best, mid); right = mid - 1; } } cout << best << endl; return 0;}Title: I-矩形分割Code: #include <bits/stdc++.h>#include <iostream>#include <algorithm>#include <cstdio>#include <string>#include <cstring>#include <cstdlib>#include <map>#include <vector>#include <set>#include <queue>#include <stack>#include <cmath>using namespace std;#define mem(s,t) memset(s,t,sizeof(s))#define pq priority\_queue#define pb push\_back#define fi first#define se second#define ac return 0;#define ll long long#define rep(xx,yy) for(int i=xx;i<=yy;i++)#define TLE std::ios::sync\_with\_stdio(false); cin.tie(NULL); cout.tie(NULL); cout.precision(10);const double eps = 1e-14;const int mxn = 1e3;int woshichaode;string str;ll r,n,mx=-1;struct node{ ll x,y,r,h; ll area;} no[10000];ll check( ll mid){ ll larea = 0, rarea = 0; for(int i=1; i<=n; i++) { if( no[i].x>=mid ) rarea += no[i].area; else if(no[i].r<=mid) larea += no[i].area; else { larea += (mid-no[i].x)\*no[i].h; rarea += (no[i].r-mid)\*no[i].h; } } return larea-rarea;}int main(){ TLE; cin>>r; cin>>n; for(int i=1; i<=n; i++) { cin>>no[i].x>>no[i].y>>no[i].r>>no[i].h; no[i].area = no[i].r\*no[i].h; no[i].r += no[i].x; } if(n==1&&no[n].r-no[n].x==1) {cout<<r<<endl;return 0;} ll l = 0,ok=0; while(l<r) { ll mid = (l+r)>>1; if(!check(mid)) {ok = mid;break;} else if( check(mid)<0 ) l = mid+1; else r = mid; } if(ok) { while(ok<=r && check(ok)==check(ok+1)) ok++; cout<<ok<<endl; } else { while(l<=r && check(l)==check(l+1)) l++; cout<<l<<endl; } return 0;}Title: J-Cable masterCode: //化成整数解 , 避免精度问题 #include <cstdio>#include <cmath>using namespace std;const int MAXN = 10000 + 5;int n,k;int data[MAXN];int judge(int t){ int sum = 0; for(int i = 0; i < n; i++){ sum += (int)(data[i] / t); } return sum >= k;}int main(){ scanf("%d %d", &n, &k); for(int i = 0; i < n; i++){ double t; scanf("%lf", &t); data[i] = t\*100; } int l = 0; int r = 1e9; int m; /\* 左开右闭区间[l,r) ---> r一开始必须指定为不符合的不然r考虑不到 mid 符合 --> 答案在[m,r) mid 不符合 --> 答案在[l,m) 结束条件 --> 区间内只有一个值：[target,target+1) ---> r-l = 1结束 0 1 2 3 4 5 6 7 8 Y Y Y Y Y Y N N N 开始： l m r m符合： l m r m不符合： l m r m符合： l r ---- > 结束输出l \*/ while(r-l > 1){ m = (r-l)/2 + l; if(judge(m)) l = m; else r = m; } /\* 闭区间[l,r] mid 符合 --> 答案在[mid,r] mid 不符合 --> 答案在[mid, r-1] 结束条件 --> 区间只有一个值 ---> [target,target] 0 1 2 3 4 5 6 7 8 Y Y Y Y Y Y N N N 开始： l m r m符合： l m r m不符合： l r ----> 关键点mid的计算方式+1,保证相邻的元素取第二个不然会死循环 此时mid = (l+r)/2 + 1 = 5; m符合： l(r) ---> 结束输出(l==m==r) \*/// while(l < r){// m = (r-l)/2 + l + 1;// if(judge(m))// l = m;// else // r = m - 1;// } printf("%.2lf\r
",l/100.0); return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

# 《算法设计与分析》实验报告4

实验名称： **贪心算法**

实验日期： 2021年5月6日-5月24日

实验地点： A608

**一、实验目的**

**二、实验内容（编程题）**

Title:A-金银岛Description: 某天KID利用飞行器飞到了一个金银岛上，上面有许多珍贵的金属，KID虽然更喜欢各种宝石的艺术品，可是也不拒绝这样珍贵的金属。但是他只带着一个口袋，口袋至多只能装重量为w的物品。岛上金属有s个种类, 每种金属重量不同，分别为n1, n2, ... , ns，同时每个种类的金属总的价值也不同，分别为v1,v2, ..., vs。KID想一次带走价值尽可能多的金属，问他最多能带走价值多少的金属。注意到金属是可以被任意分割的，并且金属的价值和其重量成正比。Input: 第1行是测试数据的组数k，后面跟着k组输入。 每组测试数据占3行，第1行是一个正整数w (1 <= w <= 10000)，表示口袋承重上限。第2行是一个正整数s (1 <= s <=100)，表示金属种类。第3行有2s个正整数，分别为n1, v1, n2, v2, ... , ns, vs分别为第一种，第二种，...，第s种金属的总重量和总价值(1 <= ni <= 10000, 1 <= vi <= 10000)。 Output: k行，每行输出对应一个输入。输出应精确到小数点后2位。SampleInput: 250410 100 50 30 7 34 87 1001000051 43 43 323 35 45 43 54 87 43SampleOutput: 171.93508.00Title:B-Doing Homework againDescription:众所周知小A同学根本不学习。但是期末到了，平时不写作业的他，现在有很多作业要做。小A的老师很严格，每个老师都会给他一个deadline。如果小A在deadline后交作业，老师就会扣他的分。现在假设小A做每门作业都需要一天。所以小A想到要安排做作业的顺序，这样才能尽可能扣少一点分。请帮帮小A吧。Input:输入包含T个测试用例。输入的第一行是单个整数T，为测试用例的数量。每个测试用例以一个正整数N开头(1<=N<=1000)，表示作业的数量。然后两行。第一行包含N个整数(每个整数均小于10^9)，表示deadline，下一行包含N个整数，表示扣的分。Output:对于每个测试用例，您应该输出最小的总降低分数，每个测试用例一行。SampleInput: 333 3 310 5 131 3 16 2 371 4 6 4 2 4 33 2 1 7 6 5 4SampleOutput:035Title:C-Moving TablesDescription:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
The famous ACM (Advanced Computer Maker) Company has rented a floor of a building whose shape is in the following figure.
The floor has 200 rooms each on the north side and south side along the corridor. Recently the Company made a plan to reform its system. The reform includes moving a lot of tables between rooms. Because the corridor is narrow and all the tables are big, only one table can pass through the corridor. Some plan is needed to make the moving efficient. The manager figured out the following plan: Moving a table from a room to another room can be done within 10 minutes. When moving a table from room i to room j, the part of the corridor between the front of room i and the front of room j is used. So, during each 10 minutes, several moving between two rooms not sharing the same part of the corridor will be done simultaneously. To make it clear the manager illustrated the possible cases and impossible cases of simultaneous moving.
For each room, at most one table will be either moved in or moved out. Now, the manager seeks out a method to minimize the time to move all the tables. Your job is to write a program to solve the manager’s problem.
Input:The input consists of T test cases. The number of test cases ) (T is given in the first line of the input. Each test case begins with a line containing an integer N , 1<=N<=200 , that represents the number of tables to move. Each of the following N lines contains two positive integers s and t, representing that a table is to move from room number s to room number t (each room number appears at most once in the N lines). From the N+3-rd line, the remaining test cases are listed in the same manner as above.
Output:The output should contain the minimum time in minutes to complete the moving, one per line.
SampleInput: 3 4 10 20 30 40 50 60 70 80 2 1 3 2 200 3 10 100 20 80 30 50 SampleOutput:102030Title:D-电影节Description: 大学生电影节在北大举办! 这天，在北大各地放了多部电影，给定每部电影的放映时间区间，区间重叠的电影不可能同时看（端点可以重合），问李雷最多可以看多少部电影。Input: 多组数据。每组数据开头是n(n<=100)，表示共n场电影。 接下来n行，每行两个整数(0到1000之间)，表示一场电影的放映区间 n=0则数据结束Output: 对每组数据输出最多能看几部电影SampleInput: 83 40 7 3 8 15 1915 2010 158 18 6 12 0SampleOutput: 3Title:E-今年暑假不ACDescription:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
“今年暑假不AC？”
“是的。”
“那你干什么呢？”
“看世界杯呀，笨蛋！”
“@#$%^&\*%...”
确实如此，世界杯来了，球迷的节日也来了，估计很多ACMer也会抛开电脑，奔向电视了。
作为球迷，一定想看尽量多的完整的比赛，当然，作为新时代的好青年，你一定还会看一些其它的节目，比如新闻联播（永远不要忘记关心国家大事）、非常6+7、超级女生，以及王小丫的《开心辞典》等等，假设你已经知道了所有你喜欢看的电视节目的转播时间表，你会合理安排吗？（目标是能看尽量多的完整节目）
Input:输入数据包含多个测试实例，每个测试实例的第一行只有一个整数n(n<=100)，表示你喜欢看的节目的总数，然后是n行数据，每行包括两个数据Ti\_s,Ti\_e (1<=i<=n)，分别表示第i个节目的开始和结束时间，为了简化问题，每个时间都用一个正整数表示。n=0表示输入结束，不做处理。
Output:对于每个测试实例，输出能完整看到的电视节目的个数，每个测试实例的输出占一行。SampleInput: 121 33 40 73 815 1915 2010 158 186 125 104 142 90SampleOutput:5Title:F-最短前缀Description: 一个字符串的前缀是从该字符串的第一个字符起始的一个子串。例如 "carbon"的字串是: "c", "ca", "car", "carb", "carbo", 和 "carbon"。注意到这里我们不认为空串是字串, 但是每个非空串是它自身的字串. 我们现在希望能用前缀来缩略的表示单词。例如, "carbohydrate" 通常用"carb"来缩略表示. 现在给你一组单词, 要求你找到唯一标识每个单词的最短前缀 在下面的例子中，"carbohydrate" 能被缩略成"carboh", 但是不能被缩略成"carbo" (或其余更短的前缀) 因为已经有一个单词用"carbo"开始 一个精确匹配会覆盖一个前缀匹配，例如，前缀"car"精确匹配单词"car". 因此 "car" 是 "car"的缩略语是没有二义性的 , “car”不会被当成"carriage"或者任何在列表中以"car"开始的单词. Input: 输入包括至少2行，至多1000行. 每行包括一个以小写字母组成的单词，单词长度至少是1，至多是20.Output: 输出的行数与输入的行数相同。每行输出由相应行输入的单词开始，后面跟着一个空格接下来是相应单词的没有二义性的最短前缀标识符。SampleInput: carbohydratecartcarburetorcaramelcariboucarboniccartilagecarboncarriagecartoncarcarbonateSampleOutput: carbohydrate carbohcart cartcarburetor carbucaramel caracaribou caricarbonic carbonicartilage carticarbon carboncarriage carrcarton cartocar carcarbonate carbonaTitle:G-最小新整数Description: 给定一个十进制正整数n(0 < n < 1000000000)，每个数位上数字均不为0。n的位数为m。现在从m位中删除k位(0<k < m)，求生成的新整数最小为多少？例如: n = 9128456, k = 2, 则生成的新整数最小为12456Input: 第一行t, 表示有t组数据； 接下来t行，每一行表示一组测试数据，每组测试数据包含两个数字n, k。Output: t行，每行一个数字，表示从n中删除k位后得到的最小整数。SampleInput: 29128456 21444 3SampleOutput: 124561Title:H-IntervalsDescription: There is given the series of n closed intervals [ai; b i], where i=1,2,...,n. The sum of those intervals may berepresented as a sum of closed pairwise non−intersecting intervals. The task is to find such representation with the minimal number of intervals. The intervals of this representation should be written in the output file in acceding order. We say that the intervals [a; b] and [c; d] are in ascending order if, and only if a <= b < c <= d. Task Write a program which: .reads from the std input the description of the series of intervals, .computes pairwise non−intersecting intervals satisfying the conditions given above, .writes the computed intervals in ascending order into std outputInput: In the first line of input there is one integer n, 3 <= n <= 50000. This is the number of intervals. In the (i+1)−st line, 1 <= i <= n, there is a description of the interval [ai; bi] in the form of two integers ai and bi separated by a single space, which are respectively the beginning and the end of the interval,1 <= ai <= bi <= 1000000.Output: The output should contain descriptions of all computed pairwise non−intersecting intervals. In each line should be written a description of one interval. It should be composed of two integers, separated by a single space, the beginning and the end of the interval respectively. The intervals should be written into the output in ascending order.SampleInput: 55 61 410 106 98 10SampleOutput: 1 45 10Title:I-饭卡Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
电子科大本部食堂的饭卡有一种很诡异的设计，即在购买之前判断余额。如果购买一个商品之前，卡上的剩余金额大于或等于5元，就一定可以购买成功（即使购买后卡上余额为负），否则无法购买（即使金额足够）。所以大家都希望尽量使卡上的余额最少。
某天，食堂中有n种菜出售，每种菜可购买一次。已知每种菜的价格以及卡上的余额，问最少可使卡上的余额为多少。
Input:多组数据。对于每组数据：
第一行为正整数n，表示菜的数量。n<=1000。
第二行包括n个正整数，表示每种菜的价格。价格不超过50。
第三行包括一个正整数m，表示卡上的余额。m<=1000。
n=0表示数据结束。
Output:对于每组输入,输出一行,包含一个整数，表示卡上可能的最小余额。SampleInput: 1505101 2 3 2 1 1 2 3 2 1500SampleOutput:-4532Title:J-TaskDescription:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
Today the company has m tasks to complete. The ith task need xi minutes to complete. Meanwhile, this task has a difficulty level yi. The machine whose level below this task’s level yi cannot complete this task. If the company completes this task, they will get (500\*xi+2\*yi) dollars.
The company has n machines. Each machine has a maximum working time and a level. If the time for the task is more than the maximum working time of the machine, the machine can not complete this task. Each machine can only complete a task one day. Each task can only be completed by one machine.
The company hopes to maximize the number of the tasks which they can complete today. If there are multiple solutions, they hopes to make the money maximum.
Input:The input contains several test cases.
The first line contains two integers N and M. N is the number of the machines.M is the number of tasks(1 < =N <= 100000,1<=M<=100000).
The following N lines each contains two integers xi(0<xi<1440),yi(0=<yi<=100).xi is the maximum time the machine can work.yi is the level of the machine.
The following M lines each contains two integers xi(0<xi<1440),yi(0=<yi<=100).xi is the time we need to complete the task.yi is the level of the task.Output:For each test case, output two integers, the maximum number of the tasks which the company can complete today and the money they will get.SampleInput: 1 2100 3100 2100 1SampleOutput:1 50004

**三、程序代码（可附算法）**

Title: A-金银岛Code: #include <bits/stdc++.h>using namespace std;const int N = 100;struct Node { int w, v; double avg;} a[N];bool cmp(Node a, Node b){ return a.avg > b.avg;}int main(){ int k, w, s; scanf("%d", &k); while(k--) { scanf("%d%d", &w, &s); for(int i = 0; i < s; i++) { scanf("%d%d", &a[i].w, &a[i].v); a[i].avg = (double)a[i].v / (double)a[i].w; } sort(a, a + s, cmp); double tot = 0; for(int i = 0; i < s; i++) if(w >= a[i].w) { w -= a[i].w; tot += a[i].v; } else { tot += w \* a[i].avg; break; } printf("%.2f\r
", tot); } return 0;}Title: B-Doing Homework againCode: #include<bits/stdc++.h>using namespace std;const int MAXN = 1000+5;\typedef struct { int ddl; int fen;}node;node a[MAXN];int book[MAXN];int cmp(node x, node y){ if(x.fen > y.fen) return 1; else if(x.fen == y.fen) return x.ddl < y.ddl; else return 0;}int main(){ int T,n,ans; cin >> T; while(T--){ cin >> n; ans = 0; for(int i = 1; i <= n; i++) book[i] = 0; for(int i = 1; i <= n; i++) cin >> a[i].ddl; for(int i = 1; i <= n; i++){ cin >> a[i].fen; ans += a[i].fen; } // printf("ans = %d",ans);// for(int i = 1; i <= n; i++)// printf("i:%d ddl:%d, fen:%d\r
",i,a[i].ddl,a[i].fen); sort(a+1,a+1+n,cmp); // printf("\r
");// for(int i = 1; i <= n; i++)// printf("i:%d ddl:%d, fen:%d\r
",i,a[i].ddl,a[i].fen); for(int i = 1; i <= n; i++){// cout <<"xh i:" <<i<<endl;// cout <<"book 3"<<book[3] << endl; if(book[a[i].ddl] == 0){ book[a[i].ddl] = 1; ans -= a[i].fen; }else{ for(int j = a[i].ddl -1; j > 0; j--){ if(book[j] == 0){ book[j] = 1; ans -= a[i].fen; break; } } } } cout << ans << endl; } return 0;}Title: C-Moving TablesCode: #include<iostream>#include<cstring>#include<algorithm>int a[402];using namespace std;int main(){ int t,n,x,y; cin>>t; while(t--){ cin>>n; memset(a,0,sizeof(a)); while(n--){ cin>>x>>y; if(x>y) swap(x,y); if(x%2==0) x = x-1; if(y%2!=0) y = y+1; for(int i = x; i <= y; i++){ a[i]++; } } int maxx = -1e8; for(int i = 1; i <= 401; i++) maxx = max(maxx,a[i]); //sort(a+1,a+401); cout<<maxx\*10<<endl; } return 0;} Title: D-电影节Code: #include <bits/stdc++.h>using namespace std;const int MAXN = 100+5;struct Node { int s; int e;} a[MAXN];int cmp(Node a, Node b){ return a.e < b.e;}int main(){ int n,ans; while(cin >> n && n){ for(int i = 1; i <= n; i++) cin >> a[i].s >> a[i].e; sort(a+1,a+1+n,cmp); // for(int i = 1; i <= n; i++){// printf("%d %d\r
",a[i].s,a[i].e);// } ans = 1; int right = a[1].e; for(int i = 2; i <= n; i++){ if(a[i].s >= right){ right = a[i].e; ans ++; } } cout << ans <<endl; } return 0;}Title: E-今年暑假不ACCode: #include <bits/stdc++.h>using namespace std;const int MAXN = 100+5;struct Node { int s; int e;} a[MAXN];int cmp(Node a, Node b){ return a.e < b.e;}int main(){ int n,ans; while(cin >> n && n){ for(int i = 1; i <= n; i++) cin >> a[i].s >> a[i].e; sort(a+1,a+1+n,cmp); // for(int i = 1; i <= n; i++){// printf("%d %d\r
",a[i].s,a[i].e);// } ans = 1; int right = a[1].e; for(int i = 2; i <= n; i++){ if(a[i].s >= right){ right = a[i].e; ans ++; } } cout << ans <<endl; } return 0;}Title: F-最短前缀Code: #include<iostream>#include<algorithm>using namespace std;const int MAXN = 1000 + 5;string s[MAXN];int main(){ int len = 1; while(cin >> s[len++]){} for(int i = 1; i < len; i++){ int j; for(j = 0; j < s[i].length(); j++){ string st(s[i].begin(), s[i].begin() + j); int cnt = 0; for(int k = 1; k < len; k++){ if(!s[k].find(st)) cnt ++; } if(cnt == 1){ cout << s[i] << " " << st << endl; break; } } if(j == s[i].length()) cout << s[i] << " " << s[i] << endl; } return 0;}Title: G-最小新整数Code: #include <bits/stdc++.h>using namespace std;const int MAXN = 15+5;char a[MAXN]; int main(){ int T,k,ans; cin >> T; while(T--){ cin >> a+1 >> k; int len = strlen(a+1); for(int i = 1; i <= k; i++){ int j,flag = 0; for(j = 1; j < len; j++){ int k = j+1; while(a[k] == '@') k++; if(a[j] != '@' && a[j] > a[k]){ a[j] = '@'; flag = 1; break; } } if(!flag){ a[j] = '@'; len--; } } // for(int i = 1; i <= strlen(a+1); i++)// printf("%c ",a[i]);// cout << endl; int ans = 0; for(int i = 1; i <= len; i++){ if(a[i] != '@') ans = ans \* 10 + (a[i] - '0'); } cout << ans << endl; } return 0;}Title: H-IntervalsCode: #include<bits/stdc++.h>using namespace std;const int MAXN = 50005;struct node{ int l; int r;}a[MAXN];bool cmp(node a,node b){ if(a.l == b.l) return a.r < b.r; return a.l < b.l;}int main(){ int n,flag = 1; cin >> n; for(int i = 1;i <= n;i++) cin >> a[i].l >> a[i].r; sort(a+1,a+1+n,cmp); for(int i=1;i<=n;i++) { if(a[i].l <= a[flag].r) a[flag].r=max(a[flag].r,a[i].r); else{ cout << a[flag].l << " " <<a[flag].r << endl; flag = i; } } cout << a[flag].l << " " << a[flag].r << endl;}Title: I-饭卡Code: #include <iostream>#include <algorithm>#include <cstring>using namespace std;const int MAX = 1000+50;int dp[MAX],a[MAX];int main(){ int n; while(cin >> n && n){ memset(dp,0,sizeof(dp)); for (int i = 0; i < n;i++){ cin >> a[i]; } sort(a,a+n); int m; cin >> m; if(m<5){ cout << m << endl; continue; } for (int i = 0; i < n-1;i++){ for (int j = m-5 ; j >= a[i];j--){ dp[j]=max(dp[j],dp[j-a[i]]+a[i]); } } cout << m-dp[m-5]-a[n-1] << endl; } return 0;}Title: J-TaskCode: #include <stdio.h>#include <string.h>#include <iostream>#include <map>#include <algorithm>using namespace std;struct note{ int x,y;}a[100005],b[100005];bool cmp(note a,note b){ if(a.x==b.x) return a.y>b.y; return a.x>b.x;}map<int,int>p;int main(){ int n,m; while(~scanf("%d%d",&n,&m)) { for(int i=0;i<n;i++) scanf("%d%d",&a[i].x,&a[i].y); for(int i=0;i<m;i++) scanf("%d%d",&b[i].x,&b[i].y); sort(a,a+n,cmp); sort(b,b+m,cmp); int j=0,count=0; long long sum=0; p.clear(); for(int i=0;i<m;i++) { while(j<n&&a[j].x>=b[i].x) { p[a[j].y]++; j++; } map<int,int>::iterator it=p.lower\_bound(b[i].y); if(it!=p.end()) { count++; sum+=(500\*b[i].x+2\*b[i].y); int t=it->first; p[t]--; if(p[t]==0) p.erase(t); } } printf("%d %I64d\r
",count,sum); } return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

# 《算法设计与分析》实验报告5

实验名称：  **动态规划**

实验日期： 2021年5月25日-6月15日

实验地点： A608

**一、实验目的**

1．

2．

3．

**二、实验内容（编程题）**

Title:A-简单的整数划分问题Description: 将正整数n 表示成一系列正整数之和，n=n1+n2+…+nk, 其中n1>=n2>=…>=nk>=1 ，k>=1 。正整数n 的这种表示称为正整数n 的划分。正整数n 的不同的划分个数称为正整数n 的划分数。 Input: 标准的输入包含若干组测试数据。每组测试数据是一个整数N(0 < N <= 50)。 Output: 对于每组测试数据，输出N的划分数。 SampleInput: 5 SampleOutput: 7 Title:B-复杂的整数划分问题Description: 将正整数n 表示成一系列正整数之和，n=n1+n2+…+nk, 其中n1>=n2>=…>=nk>=1 ，k>=1 。正整数n 的这种表示称为正整数n 的划分。Input: 标准的输入包含若干组测试数据。每组测试数据是一行输入数据,包括两个整数N 和 K。 (0 < N <= 50, 0 < K <= N)Output: 对于每组测试数据，输出以下三行数据: 第一行: N划分成K个正整数之和的划分数目 第二行: N划分成若干个不同正整数之和的划分数目 第三行: N划分成若干个奇正整数之和的划分数目SampleInput: 5 2SampleOutput: 233Title:C-Coin ChangeDescription:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
Suppose there are 5 types of coins: 50-cent, 25-cent, 10-cent, 5-cent, and 1-cent. We want to make changes with these coins for a given amount of money.
For example, if we have 11 cents, then we can make changes with one 10-cent coin and one 1-cent coin, or two 5-cent coins and one 1-cent coin, or one 5-cent coin and six 1-cent coins, or eleven 1-cent coins. So there are four ways of making changes for 11 cents with the above coins. Note that we count that there is one way of making change for zero cent.
Write a program to find the total number of different ways of making changes for any amount of money in cents. Your program should be able to handle up to 100 coins.
Input:The input file contains any number of lines, each one consisting of a number ( ≤250 ) for the amount of money in cents.Output:For each input line, output a line containing the number of different ways of making changes with the above 5 types of coins.SampleInput: 1126SampleOutput:413Title:D-最大报销额Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
现有一笔经费可以报销一定额度的发票。允许报销的发票类型包括买图书（A类）、文具（B类）、差旅（C类），要求每张发票的总额不得超过1000元，每张发票上，单项物品的价值不得超过600元。现请你编写程序，在给出的一堆发票中找出可以报销的、不超过给定额度的最大报销额。
Input:测试输入包含若干测试用例。每个测试用例的第1行包含两个正数 Q 和 N，其中 Q 是给定的报销额度，N（<=30）是发票张数。随后是 N 行输入，每行的格式为：
m Type\_1:price\_1 Type\_2:price\_2 ... Type\_m:price\_m
其中正整数 m 是这张发票上所开物品的件数，Type\_i 和 price\_i 是第 i 项物品的种类和价值。物品种类用一个大写英文字母表示。当N为0时，全部输入结束，相应的结果不要输出。
Output:对每个测试用例输出1行，即可以报销的最大数额，精确到小数点后2位。
SampleInput: 200.00 32 A:23.50 B:100.001 C:650.003 A:59.99 A:120.00 X:10.001200.00 22 B:600.00 A:400.001 C:200.501200.50 32 B:600.00 A:400.001 C:200.501 A:100.00100.00 0SampleOutput:123.501000.001200.50Title:E-FATEDescription:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
最近xhd正在玩一款叫做FATE的游戏，为了得到极品装备，xhd在不停的杀怪做任务。久而久之xhd开始对杀怪产生的厌恶感，但又不得不通过杀怪来升完这最后一级。现在的问题是，xhd升掉最后一级还需n的经验值，xhd还留有m的忍耐度，每杀一个怪xhd会得到相应的经验，并减掉相应的忍耐度。当忍耐度降到0或者0以下时，xhd就不会玩这游戏。xhd还说了他最多只杀s只怪。请问他能升掉这最后一级吗？
Input:输入数据有多组，对于每组数据第一行输入n，m，k，s(0 < n,m,k,s < 100)四个正整数。分别表示还需的经验值，保留的忍耐度，怪的种数和最多的杀怪数。接下来输入k行数据。每行数据输入两个正整数a，b(0 < a,b < 20)；分别表示杀掉一只这种怪xhd会得到的经验值和会减掉的忍耐度。(每种怪都有无数个)Output:输出升完这级还能保留的最大忍耐度，如果无法升完这级输出-1。SampleInput: 10 10 1 101 110 10 1 91 19 10 2 101 12 2SampleOutput:0-11Title:F-最少拦截系统Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } }); setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000); 某国为了防御敌国的导弹袭击,发展出一种导弹拦截系统.但是这种导弹拦截系统有一个缺陷:虽然它的第一发炮弹能够到达任意的高度,但是以后每一发炮弹都不能超过前一发的高度.某天,雷达捕捉到敌国的导弹来袭.由于该系统还在试用阶段,所以只有一套系统,因此有可能不能拦截所有的导弹. 怎么办呢?多搞几套系统呗!你说说倒蛮容易,成本呢?成本是个大问题啊.所以俺就到这里来求救了,请帮助计算一下最少需要多少套拦截系统. Input:输入若干组数据.每组数据包括:导弹总个数(正整数),导弹依此飞来的高度(雷达给出的高度数据是不大于30000的正整数,用空格分隔)Output:对应每组数据输出拦截所有导弹最少要配备多少套这种导弹拦截系统.SampleInput: 8 389 207 155 300 299 170 158 65SampleOutput:2Title:G-Common SubsequenceDescription: 　　郭哥与瑞瑞在玩一个游戏。
　　他们先各自写下一串字符，然后互相展示。展示过后，他们再从自己写的那串字符中依次挑出若干字符（保持原有顺序不变），组成新的一串。他们希望自己新组成的字符串与对方新组成的完全相同，并且尽可能长。
　　例如，郭哥写下abcde，瑞瑞写下aeiou，然后郭哥挑出自己那串里的第1和第5个字符组成新串ae，瑞瑞挑出自己那串中的第1、2个字符，也组成字符串ae。ae就是他们能共同挑出的最长串。
　　现在，郭哥和瑞瑞分别写出了自己的字符串，请帮他们算一下他们能共同挑出组成的字符串最长有多长。Input: 　　输入包含多组数据，处理至文件结尾。
　　每组数据占一行，包括以空格分隔的两个字符串，分别是郭哥和瑞瑞写下的字符串。两个字符串长度都在1000以内。Output: 　　对于每组输入，输出一个整数，即他们能共同挑出组成的字符串的最大长度。SampleInput: abcfbc abfcabprogramming contest abcd mnpSampleOutput:420Title:H-Eating TogetherDescription:FJ的奶牛们在吃晚饭时很傻。他们把自己组织成三组（方便编号为1, 2和3），坚持一起用餐。当他们在谷仓排队进入喂食区时，麻烦就开始了。每头奶牛都随身带着一张小卡片，小卡片上刻的是Di（1≤Di≤3）表示她属于哪一组。所有的N（1≤N≤30000）头奶牛排队吃饭，但他们并不能按卡片上的分组站好。FJ的工作并不是那么难。他只是沿着牛的路线走下去，把旧的号码标出来，换上一个新的。通过这样做，他创造了一群奶牛，比如111222333或333222111，奶牛的就餐组按他们的晚餐卡片按升序或降序排列。FJ就像任何人一样懒惰。他很好奇：怎样他才能进行适当的分组，使得他只要修改最少次数的数字？由于奶牛们已经很长时间没有吃到饭了，所以“哞哞”的声音到处都是，FJ只好更换卡号，而不能重新排列已经排好队的奶牛。Input:第1行：一个整数：n；第2~n+1行：第i-1行描述第i个奶牛目前分组。 Output:一个整数，表示必须做出的最小变化数，以便以升序或降序排序最终序列。 SampleInput: 513211SampleOutput:1Title:I-命运Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
穿过幽谷意味着离大魔王lemon已经无限接近了！
可谁能想到，yifenfei在斩杀了一些虾兵蟹将后，却再次面临命运大迷宫的考验，这是魔王lemon设下的又一个机关。要知道，不论何人，若在迷宫中被困1小时以上，则必死无疑！
可怜的yifenfei为了去救MM，义无返顾地跳进了迷宫。让我们一起帮帮执着的他吧！
命运大迷宫可以看成是一个两维的方格阵列，如下图所示：
yifenfei一开始在左上角，目的当然是到达右下角的大魔王所在地。迷宫的每一个格子都受到幸运女神眷恋或者痛苦魔王的诅咒，所以每个格子都对应一个值，走到那里便自动得到了对应的值。
现在规定yifenfei只能向右或者向下走，向下一次只能走一格。但是如果向右走，则每次可以走一格或者走到该行的列数是当前所在列数倍数的格子，即：如果当前格子是（x,y），下一步可以是（x+1,y），(x,y+1)或者(x,y\*k) 其中k>1。
为了能够最大把握的消灭魔王lemon，yifenfei希望能够在这个命运大迷宫中得到最大的幸运值。
Input:输入数据首先是一个整数C，表示测试数据的组数。
每组测试数据的第一行是两个整数n,m，分别表示行数和列数(1<=n<=20,10<=m<=1000)；
接着是n行数据，每行包含m个整数，表示n行m列的格子对应的幸运值K ( |k|<100 )。
Output:请对应每组测试数据输出一个整数，表示yifenfei可以得到的最大幸运值。SampleInput: 13 89 10 10 10 10 -10 10 1010 -11 -1 0 2 11 10 -20-11 -11 10 11 2 10 -10 -10SampleOutput:52Title:J-免费馅饼Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
都说天上不会掉馅饼，但有一天gameboy正走在回家的小径上，忽然天上掉下大把大把的馅饼。说来gameboy的人品实在是太好了，这馅饼别处都不掉，就掉落在他身旁的10米范围内。馅饼如果掉在了地上当然就不能吃了，所以gameboy马上卸下身上的背包去接。但由于小径两侧都不能站人，所以他只能在小径上接。由于gameboy平时老呆在房间里玩游戏，虽然在游戏中是个身手敏捷的高手，但在现实中运动神经特别迟钝，每秒种只有在移动不超过一米的范围内接住坠落的馅饼。现在给这条小径如图标上坐标：
为了使问题简化，假设在接下来的一段时间里，馅饼都掉落在0-10这11个位置。开始时gameboy站在5这个位置，因此在第一秒，他只能接到4,5,6这三个位置中其中一个位置上的馅饼。问gameboy最多可能接到多少个馅饼？（假设他的背包可以容纳无穷多个馅饼）
Input:输入数据有多组。每组数据的第一行为以正整数n(0<n<100000)，表示有n个馅饼掉在这条小径上。在结下来的n行中，每行有两个整数x,T(0<T<100000),表示在第T秒有一个馅饼掉在x点上。同一秒钟在同一点上可能掉下多个馅饼。n=0时输入结束。
Output:每一组输入数据对应一行输出。输出一个整数m，表示gameboy最多可能接到m个馅饼。
提示：本题的输入数据量比较大，建议用scanf读入，用cin可能会超时。
SampleInput: 65 14 16 17 27 28 30SampleOutput:4Title:K-Cow Roller CoasterDescription:The cows are building a roller coaster! They want your help to design as fun a roller coaster as possible, while keeping to the budget.The roller coaster will be built on a long linear stretch of land of length L (1 ≤ L ≤ 1,000). The roller coaster comprises a collection of some of the N (1 ≤ N ≤ 10,000) different interchangable components. Each component i has a fixed length Wi (1 ≤ Wi ≤ L). Due to varying terrain, each component i can be only built starting at location Xi (0 ≤ Xi ≤ L - Wi). The cows want to string together various roller coaster components starting at 0 and ending at L so that the end of each component (except the last) is the start of the next component.Each component i has a "fun rating" Fi (1 ≤ Fi ≤ 1,000,000) and a cost Ci (1 ≤ Ci ≤ 1000). The total fun of the roller coster is the sum of the fun from each component used; the total cost is likewise the sum of the costs of each component used. The cows' total budget is B (1 ≤ B ≤ 1000). Help the cows determine the most fun roller coaster that they can build with their budget.Input:Line 1: Three space-separated integers: L, N and B.
Lines 2..N+1: Line i+1 contains four space-separated integers, respectively: Xi, Wi, Fi, and Ci.Output:Line 1: A single integer that is the maximum fun value that a roller-coaster can have while staying within the budget and meeting all the other constraints. If it is not possible to build a roller-coaster within budget, output -1.SampleInput: 5 6 100 2 20 62 3 5 60 1 2 11 1 1 31 2 5 43 2 10 2SampleOutput:17Title:L-悼念512汶川大地震遇难同胞——珍惜现在，感恩生活Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
急！灾区的食物依然短缺！
为了挽救灾区同胞的生命，心系灾区同胞的你准备自己采购一些粮食支援灾区，现在假设你一共有资金n元，而市场有m种大米，每种大米都是袋装产品，其价格不等，并且只能整袋购买。
请问：你用有限的资金最多能采购多少公斤粮食呢？
后记：
人生是一个充满了变数的生命过程，天灾、人祸、病痛是我们生命历程中不可预知的威胁。
月有阴晴圆缺，人有旦夕祸福，未来对于我们而言是一个未知数。那么，我们要做的就应该是珍惜现在，感恩生活——
感谢父母，他们给予我们生命，抚养我们成人；
感谢老师，他们授给我们知识，教我们做人
感谢朋友，他们让我们感受到世界的温暖；
感谢对手，他们令我们不断进取、努力。
同样，我们也要感谢痛苦与艰辛带给我们的财富～
Input:输入数据首先包含一个正整数C，表示有C组测试用例，每组测试用例的第一行是两个整数n和m(1<=n<=100, 1<=m<=100),分别表示经费的金额和大米的种类，然后是m行数据，每行包含3个数p，h和c(1<=p<=20,1<=h<=200,1<=c<=20)，分别表示每袋的价格、每袋的重量以及对应种类大米的袋数。Output:对于每组测试数据，请输出能够购买大米的最多重量，你可以假设经费买不光所有的大米，并且经费你可以不用完。每个实例的输出占一行。SampleInput: 18 22 100 44 100 2SampleOutput:400

**三、程序代码（可附算法）**

Title: A-简单的整数划分问题Code: #include<iostream>#include<stack>#include<string>#include<cstring>#include<vector>using namespace std;int dp[105];int main(){ int n; dp[0] = 1; for(int i = 1; i <= 50; i++) for(int j = 1; j <= 50; j++){ if(j-i >= 0) dp[j] += dp[j-i]; } while(cin >> n){ cout << dp[n] << endl; } return 0;} Title: B-复杂的整数划分问题Code: #include <iostream>#include <cstring>using namespace std;#define N 51int dp1[N][N]; //N划分成K个正整数之和的划分数目。int dp2[N][N]; //N划分成若干个不同正整数之和的划分数目。int dp3[N][N]; //N划分成若干个可相同的正整数之和的划分数目。int f[N][N]; //N划分成K个奇正整数之和的划分数目。int g[N][N]; //N划分成K个偶正整数之和的划分数目。void initDivideInt() { memset(dp1, 0, sizeof(dp1)); //dp[n][k]=dp[n-k][k]+dp[n-1][k-1] memset(dp2, 0, sizeof(dp2)); //dp[n][m]=dp[n][m-1]+dp[n-m][m-1] memset(dp3, 0, sizeof(dp3)); //dp[n][m]=dp[n][m-1]+dp[n-m][m] for (int i = 1; i < N; i++) { for (int j = 1; j < N; j++) { if (i < j) { dp1[i][j] = 0; dp2[i][j] = dp2[i][i]; dp3[i][j] = dp3[i][i]; } else if (i == j) { dp1[i][j] = 1; dp2[i][j] = dp2[i][j - 1] + 1; dp3[i][j] = dp3[i][j - 1] + 1; } else { dp1[i][j] = dp1[i - j][j] + dp1[i - 1][j - 1]; dp2[i][j] = dp2[i][j - 1] + dp2[i - j][j - 1]; dp3[i][j] = dp3[i][j - 1] + dp3[i - j][j]; } } }}void initDivideOdd() { f[0][0] = 1; g[0][0] = 1; for (int i = 1; i < N; i++) { for (int j = 1; j <= i; j++) { g[i][j] = f[i - j][j]; f[i][j] = f[i - 1][j - 1] + g[i - j][j]; } }}int main() { int n, k; initDivideInt(); initDivideOdd(); while (cin >> n >> k) { cout << dp1[n][k] << endl; cout << dp2[n][n] << endl; int sum = 0; for (int i = 0; i <= n; i++) { sum += f[n][i]; } cout << sum << endl; } return 0;}Title: C-Coin ChangeCode: #include<stdio.h>int main(){ int a,b,c,d,e,count,n; while(scanf("%d",&n)!=EOF) { count=0; for(a=0;a<=n;a++) for(b=0;5\*b<=n-a;b++) for(c=0;10\*c<=n-a-5\*b;c++) for(d=0;25\*d<=n-a-5\*b-10\*c;d++) { e=n-a-5\*b-10\*c-25\*d; if(e%50==0&&a+b+c+d+e/50<=100)count++; } printf("%d\r
",count); }}Title: D-最大报销额Code: #include<iostream>#include<cstdio>#include<cstring>#include<algorithm>#include<cmath>#include<iomanip>using namespace std;int cost[35];int dp[3000300];int main(){ //freopen("in.txt","r",stdin); double q; int n; while(cin>>q>>n&&n){ memset(dp,0,sizeof(dp)); int index = 1; while(n--){ double t,sum = 0; double A = 0,B = 0,C = 0; bool flag = 0; int cnt; cin>>cnt; while(cnt--){ char type,pass; double money; cin>>type>>pass>>money; if(money > 600) flag = 1; if(type == 'A') A += money; else if(type == 'B' ) B += money; else if(type == 'C') C += money; else flag = 1; sum = A + B + C; if(sum > 1000 || A > 600 || B > 600 || C > 600) { flag = 1; } } if(!flag) cost[index++] = sum\*100; } // for(int i = 1; i < index; i++)// cout<<cost[i]<<" "; int qq = (int)(q\*100); for(int i = 1; i < index; i++) for(int j = qq; j >= cost[i]; j--) dp[j] = max(dp[j],dp[j-cost[i]]+cost[i]); cout<<fixed<<setprecision(2)<<dp[qq]/100.00<<endl; } return 0;} Title: E-FATECode: #include<stdio.h>#include<string.h>#include<algorithm>using namespace std;struct node{ int j,r;}mon[1010];int dp[1010][1010];int cmp(node x,node y){ return x.r<y.r;}int main(){ int n,m,a,b; while(scanf("%d%d%d%d",&n,&m,&a,&b)!=EOF)//经验 忍耐 种族 杀怪数 { memset(dp,0,sizeof(dp)); for(int i=1; i<=a; i++) scanf("%d%d",&mon[i].j,&mon[i].r); int maxx,flag=0; for(int i=1; i<=a; i++) { for(int j=1; j<=b; j++) { for(int k=mon[i].r; k<=m; k++) { dp[j][k]=max(dp[j][k],dp[j-1][k-mon[i].r]+mon[i].j); } } } for(int j=1;j<=b;j++) for(int i=0; i<=m; i++) if(dp[j][i]>=n) { flag=1; maxx=m-i; break; } if(flag==0) printf("-1\r
"); else printf("%d\r
",maxx); } return 0;}Title: F-最少拦截系统Code: #include<iostream>using namespace std;const int MAXN = 10000;int n,high[MAXN];int LIS(){ int ans = 1; int dp[MAXN]; dp[1] = 1; for(int i = 2; i <= n; i++){ int maxx = 0; for(int j = 1; j < i; j++) if(dp[j] > maxx && high[j] < high[i]) maxx = dp[j]; dp[i] = maxx + 1; if(dp[i] > ans) ans = dp[i]; } return ans;}int main(){ while(cin >> n){ for(int i = 1; i <= n; i++) cin >> high[i]; cout << LIS() << endl; } return 0;}Title: G-Common SubsequenceCode: #include<bits/stdc++.h>using namespace std;const int MAXN = 1000 + 5;char a[MAXN];char b[MAXN]; int dp[MAXN][MAXN];int main(){ while(cin >> a+1 >> b+1){ int len1 = strlen(a+1); int len2 = strlen(b+1); int maxx = max(len1, len2); for(int i = 0; i <= maxx; i++){ dp[i][0] = 0; dp[0][i] = 0; } for(int i = 1; i <= len1; i++) for(int j = 1; j <= len2; j++){ if(a[i] == b[j]) dp[i][j] = dp[i-1][j-1] + 1; else dp[i][j] = max(dp[i-1][j], dp[i][j-1]); } cout << dp[len1][len2] << endl; } return 0;}Title: H-Eating TogetherCode: #include<cstdio>#include<cstring>#include<iostream>#include<algorithm>#include<queue>#include<stack>#include<vector>#include<cstdlib> using namespace std;const int maxn = 300000;int A[maxn],f1[maxn],f2[maxn];int main(){ int n;scanf("%d",&n); for(int i = 1;i <= n;i++) scanf("%d",&A[i]); int cnt = 1;f1[cnt] = A[1]; for(int i = 2;i <= n;i++){ if(A[i] >= f1[cnt]){ f1[++cnt] = A[i]; }else \*upper\_bound(f1+1,f1+cnt+1,A[i]) = A[i]; } int ans = cnt; cnt = 1;f2[cnt] = A[1]; for(int i = 2;i <= n;i++){ if(A[i] <= f2[cnt]) f2[++cnt] = A[i]; else \*upper\_bound(f2+1,f2+1+cnt,A[i],greater<int>()) = A[i]; } printf("%d\r
",n-max(ans,cnt)); return 0;} Title: I-命运Code: #include<iostream>#include<algorithm>using namespace std;int n,m,C;const int MAXN = 1005;int dp[MAXN][MAXN];int a[MAXN][MAXN];int main(){ cin >> C; while(C--){ cin >> n >> m; for(int i = 1; i <= n; i++) for(int j = 1; j <= m; j++) cin >> a[i][j]; for(int i = 0; i <= n; i++) dp[i][0] = -1e8; for(int i = 0; i <= m; i++) dp[0][i] = -1e8; dp[1][1] = a[1][1]; for(int i = 1; i <= n; i++){ for(int j = 1; j <= m; j++){ if(i == 1 && j == 1) continue; dp[i][j] = max(dp[i-1][j], dp[i][j-1]) + a[i][j]; for(int k = 1; k < j; k ++){ if(j % k == 0) dp[i][j] = max(dp[i][j], dp[i][k] + a[i][j]); } } } cout << dp[n][m] << endl; } return 0;}Title: J-免费馅饼Code: #include<bits/stdc++.h>using namespace std;const int MAXN = 100000+5;//dp[i][j]:第i秒j位置的个数 int dp[MAXN][11];//int a[MAXN][11];int m,n;int maxx;int main(){ while(~scanf("%d",&n) && n){// for(int i = 0; i <= 10; i++)// for(int j = 1; j < MAXN; j++){// dp[j][i] = 0;// } memset(dp,0,sizeof(dp)); m = maxx = -1e3; for(int i = 1; i <= n; i++){ int x,y; scanf("%d %d",&x,&y); if(y == 1){ if(x == 4 || x == 5 || x == 6) dp[y][x]++; } else{ dp[y][x]++; } maxx = max(maxx, y); } // for(int i = 1; i <= maxx; i++){// for(int j = 4; j <=8 ;j++){// cout << i << "--" <<j<<":"<<dp[i][j]<<endl;// }// } for(int i = 2; i <= maxx; i++){ dp[i][0] = max(dp[i-1][1], dp[i-1][0]) + dp[i][0]; for(int j = 1; j <= 9; j++){ dp[i][j] = max(max(dp[i-1][j-1] ,dp[i-1][j]), dp[i-1][j+1]) + dp[i][j]; } dp[i][10] = max(dp[i-1][9], dp[i-1][10]) + dp[i][10]; } for(int i = 0; i <= 10; i++){ m = max(dp[maxx][i], m); } printf("%d\r
",m); } return 0;}Title: K-Cow Roller CoasterCode: #include<cstdio>#include<cstring>#include<iostream>#include<algorithm>using namespace std;const int maxn=10020;struct node{ int x; int w; int f; int c; bool operator <(const node &a)const { if(x==a.x) return w<a.w; return x<a.x; }}a[maxn];int l,n,b,dp[1001][1001];int main(){ while(scanf("%d%d%d",&l,&n,&b)!=EOF) { memset(dp,-1,sizeof(dp)); for(int i=0;i<n;i++) scanf("%d%d%d%d",&a[i].x,&a[i].w,&a[i].f,&a[i].c); sort(a,a+n); for(int i=0;i<=b;i++) dp[0][i]=0; for(int i=0;i<n;i++) for(int j=0;j<=b-a[i].c;j++) if(dp[a[i].x][j]!=-1) dp[a[i].x+a[i].w][j+a[i].c]=max(dp[a[i].x+a[i].w][j+a[i].c],dp[a[i].x][j]+a[i].f); printf("%d\r
",dp[l][b]); } return 0;}Title: L-悼念512汶川大地震遇难同胞——珍惜现在，感恩生活Code: #include<iostream>using namespace std;struct item{ int p; int h; int c;};item a[105];int dp[105];int main(){ int C,n,m; cin >> C; while(C--){ cin >> m >> n; for(int i = 0; i <= 104; i++) dp[i] = 0; for(int i = 0; i < n; i++) cin >> a[i].p >> a[i].h >> a[i].c; for(int i = 0; i < n; i++){ for(int j = 0; j < a[i].c; j++){ for(int k = m; k >= a[i].p; k--){ dp[k] = max(dp[k], dp[k-a[i].p]+a[i].h); } } } cout << dp[m] << endl; } return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

# 《算法设计与分析》实验报告6

实验名称：  **搜索算法**

实验日期：

实验地点： A608

**一、实验目的**

1．

2．

3．

**二、实验内容（编程题）**

Title:A-踩方格Description: 有一个方格矩阵，矩阵边界在无穷远处。我们做如下假设：a.    每走一步时，只能从当前方格移动一格，走到某个相邻的方格上；b.    走过的格子立即塌陷无法再走第二次；c.    只能向北、东、西三个方向走；请问：如果允许在方格矩阵上走n步，共有多少种不同的方案。2种走法只要有一步不一样，即被认为是不同的方案。 Input: 允许在方格上行走的步数n(n <= 20) Output: 计算出的方案数量 SampleInput: 2 SampleOutput: 7 Title:B-棋盘问题Description:在一个给定形状的棋盘（形状可能是不规则的）上面摆放棋子，棋子没有区别。要求摆放时任意的两个棋子不能放在棋盘中的同一行或者同一列，请编程求解对于给定形状和大小的棋盘，摆放k个棋子的所有可行的摆放方案C。Input:输入含有多组测试数据。
每组数据的第一行是两个正整数，n k，用一个空格隔开，表示了将在一个n\*n的矩阵内描述棋盘，以及摆放棋子的数目。 n <= 8 , k <= n
当为-1 -1时表示输入结束。
随后的n行描述了棋盘的形状：每行有n个字符，其中 # 表示棋盘区域， . 表示空白区域（数据保证不出现多余的空白行或者空白列）。
Output:对于每一组数据，给出一行输出，输出摆放的方案数目C （数据保证C<2^31）。SampleInput: 2 1#..#4 4...#..#..#..#...-1 -1SampleOutput:21Title:C-迷宫问题Description: 定义一个二维数组： int maze[5][5] = {0, 1, 0, 0, 0,0, 1, 0, 1, 0,0, 0, 0, 0, 0,0, 1, 1, 1, 0,0, 0, 0, 1, 0,}; 它表示一个迷宫，其中的1表示墙壁，0表示可以走的路，只能横着走或竖着走，不能斜着走，要求编程序找出从左上角到右下角的最短路线。 Input: 一个5 × 5的二维数组，表示一个迷宫。数据保证有唯一解。Output: 左上角到右下角的最短路径，格式如样例所示。SampleInput: 0 1 0 0 00 1 0 1 00 0 0 0 00 1 1 1 00 0 0 1 0SampleOutput: (0, 0)(1, 0)(2, 0)(2, 1)(2, 2)(2, 3)(2, 4)(3, 4)(4, 4)Title:D-Dungeon MasterDescription:(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)(https://imgchr.com/i/BuF0OO)花花被困在一个三维的空间中,现在要寻找最短路径逃生！空间由立方体单位构成。花花每次向上下前后左右移动一个单位需要一分钟，且花花不能对角线移动。空间的四周封闭。花花的目标是走到空间的出口。是否存在逃出生天的可能性？如果存在，则需要多少时间？Input:每个空间的描述的第一行为L，R和C（皆不超过30）。L表示空间的高度，R和C分别表示每层空间的行与列的大小。随后L层，每层R行，每行C个字符。每个字符表示空间的一个单元。'#'表示不可通过单元，'.'表示空白单元。花花的起始位置在'S'，出口为'E'。每层空间后都有一个空行。L，R和C均为0时输入结束。Output:每个空间对应一行输出。如果可以逃生，则输出如下Escaped in x minute(s).x为最短脱离时间。如果无法逃生，则输出如下Trapped!SampleInput: 3 4 5S.....###..##..###.#############.####...###########.#######E1 3 3S###E####0 0 0SampleOutput:Escaped in 11 minute(s).Trapped!Title:E-FliptileDescription:农夫约翰知道，一头知识上满意的母牛是一头快乐的母牛，它将提供更多的牛奶。他为母牛安排了脑力活动，他们在其中操纵M × N 网格（1≤ M ≤15； 1≤ N ≤15）个正方形瓷砖，每个瓷砖的一面都涂成黑色，另一面则涂成白色。 正如人们所猜测的那样，当翻转单个白色瓷砖时，它会改变变黑翻转单个黑色图块时，它会变为白色。母牛在翻转砖块时会得到奖励，因此每块砖块的白色侧面都朝上。但是，母牛的蹄子相当大，当他们尝试翻转特定的砖块时，它们也会翻转所有相邻的砖块（与翻转后的砖块共享整个边缘的砖块）。由于翻转很累，奶牛希望尽量减少必须做的翻转次数。 帮助母牛确定所需的最小翻转次数，并确定要达到最小翻转次数的位置。如果有多种方法可以以最少的翻转次数来完成任务，则当将其视为字符串时，以输出中词典顺序最少的方式返回。如果无法完成任务，请用“ IMPOSSIBLE”一词打印一行。Input:第1行：两个以空格分隔的整数： M 和 N 第2 .. M +1行：第 i +1行描述了网格中第i行（ N 以空格分隔的整数，黑色为1，白色为0。Output:第1 .. M 行：每行包含 N 个以空格分隔的整数，每个整数指定翻转该特定位置的次数。SampleInput: 4 41 0 0 10 1 1 00 1 1 01 0 0 1SampleOutput:0 0 0 01 0 0 11 0 0 10 0 0 0Title:F-Find The MultipleDescription: Given a positive integer n, write a program to find out a nonzero multiple m of n whose decimal representation contains only the digits 0 and 1. You may assume that n is not greater than 200 and there is a corresponding m containing no more than 100 decimal digits.Input: The input file may contain multiple test cases. Each line contains a value of n (1 <= n <= 200). A line containing a zero terminates the input.Output: For each value of n in the input print a line containing the corresponding value of m. The decimal representation of m must not contain more than 100 digits. If there are multiple solutions for a given value of n, any one of them is acceptable.SampleInput: 26190SampleOutput: 10100100100100100100111111111111111111Title:G-Prime PathDescription:正如前文所说，eroengine突然喜爱上了质数，他对质数的爱已经到了魔怔的地步。现在他想到了一个小游戏，首先随机挑选两个四位的素数a，b。游戏规则是：a可以通过改变某一位上的数字使其变成c，但只有当c也是四位的素数时才能进行这种改变。eroengine只擅长抛出问题而不擅长解决问题，但他觉得你一定可以很轻松的算出a最少经过多少次变化使其变为b。例如：1033 -> 8179 1033 1733 3733 3739 3779 8779 8179最少变换了6次。Input:第一行输入整数T，表示样例数。 (T 每个样例输入两个四位的素数a，b。(没有前导零) Output:对于每个样例，输出最少变换次数，如果无法变换成b则输出"Impossible"。SampleInput: 31033 81791373 80171033 1033SampleOutput:670Title:H-Oil DepositsDescription: GeoSurvComp地质调查公司负责探测地下石油储藏。 GeoSurvComp现在在一块矩形区域探测石油，并把这个大区域分成了很多小块。他们通过专业设备，来分析每个小块中是否蕴藏石油。如果这些蕴藏石油的小方格相邻，那么他们被认为是同一油藏的一部分。在这块矩形区域，可能有很多油藏。你的任务是确定有多少不同的油藏。Input: 输入可能有多个矩形区域（即可能有多组测试）。每个矩形区域的起始行包含m和n，表示行和列的数量，1<=n,m<=100，如果m =0表示输入的结束，接下来是m行，每行n个字符。每个字符对应一个小方格，并且要么是'\*'，代表没有油，要么是'@'，表示有油。Output: 对于每一个矩形区域，输出油藏的数量。两个小方格是相邻的，当且仅当他们水平或者垂直或者对角线相邻（即8个方向）。SampleInput:   1 1 \* 3 5 \*@\*@\* \*\*@\*\* \*@\*@\* 1 8 @@\*\*\*\*@\* 5 5  \*\*\*\*@ \*@@\*@ \*@\*\*@ @@@\*@ @@\*\*@ 0 0    SampleOutput: 0 1 2 2Title:I-非常可乐Description:MathJax.Hub.Config({tex2jax: { inlineMath: [['$','$']] } });
setTimeout(function(){MathJax.Hub.Queue(['Typeset', MathJax.Hub, 'left\_view']);}, 2000);
大家一定觉的运动以后喝可乐是一件很惬意的事情，但是seeyou却不这么认为。因为每次当seeyou买了可乐以后，阿牛就要求和seeyou一起分享这一瓶可乐，而且一定要喝的和seeyou一样多。但seeyou的手中只有两个杯子，它们的容量分别是N 毫升和M 毫升 可乐的体积为S （S<101）毫升　(正好装满一瓶) ，它们三个之间可以相互倒可乐 (都是没有刻度的，且 S==N+M，101＞S＞0，N＞0，M＞0) 。聪明的ACMER你们说他们能平分吗？如果能请输出倒可乐的最少的次数，如果不能输出"NO"。
Input:三个整数 : S 可乐的体积 , N 和 M是两个杯子的容量，以"0 0 0"结束。Output:如果能平分的话请输出最少要倒的次数，否则输出"NO"。SampleInput: 7 4 34 1 30 0 0SampleOutput:NO3Title:J-Find a wayDescription:伊菲的家在乡下，而梅基的家在市中心。于是，伊菲和梅基安排在KFC见面。西安有很多KFC，他们想选择一个花费总时间最短的KFC见面。现在给你一张西安的地图，伊菲和梅基都可以上、下、左、右移动到相邻的地点，每移动一个位置花费11分钟。Input:输入包含多个测试用例。每个测试用例包括前两个整数n，m.（2<=n，m<=200）。接下来的n行，每行包含m个字符。“Y”表示伊菲的初始位置。“M”表示梅基初始位置。“#”死路；'.'可走的路。“@” KCFOutput:对于每个测试用例，输出伊菲和梅基花费的最短总时间。总有一家KFC可以让他们见面。SampleInput: 4 4Y.#@.....#..@..M4 4Y.#@.....#..@#.M5 5Y..@..#....#...@..M.#...#SampleOutput:668866

**三、程序代码（可附算法）**

Title: A-踩方格Code: #include<iostream>#include<cstring>#include<cstdio>using namespace std;int dp[25][3]; int main(){ int n; dp[1][0] = 1; dp[1][1] = 1; dp[1][2] = 1; for(int i = 2; i <= 20; i++){ dp[i][0] += dp[i-1][0]; dp[i][0] += dp[i-1][1]; dp[i][1] += dp[i-1][0]; dp[i][1] += dp[i-1][1]; dp[i][1] += dp[i-1][2]; dp[i][2] += dp[i-1][1]; dp[i][2] += dp[i-1][2]; } cin >> n; cout << dp[n][0]+dp[n][1]+dp[n][2] << endl; return 0;}Title: B-棋盘问题Code: #include<iostream>using namespace std;const int MAXN = 15;int a[MAXN][MAXN];int vis[MAXN];int n,k,ans;void dfs(int step, int tot){ if(tot==0){ ans++; return; } //this if mast under ^ if if(step==n+1) return; for(int i = 1; i <= n; i++){ if(a[step][i] == 1 && !vis[i]){ vis[i] = 1; dfs(step+1,tot-1); vis[i] = 0; } } //line step no put dfs(step+1,tot);}int main(){ while(cin >> n >> k){ if(n == -1 || k == -1) break; ans = 0; for(int i = 1; i <= n; i++) for(int j = 1; j <= n; j++){ char t; cin >> t; t=='#'?a[i][j]=1:a[i][j]=0; } dfs(1,k); cout << ans << endl; } return 0;}Title: C-迷宫问题Code: #include<iostream>#include<cstring>#include<cstdio>using namespace std;bool map[6][6];bool book[6][6];int nextt[4][2] = {{0,1},{1,0},{0,-1},{-1,0}};typedef struct step{ int x; int y; int f; } step;step que[36];void pp(int now){ //printf("now:%d\r
",now); if(now==1) return; now = que[now].f; pp(now); printf("(%d, %d)\r
",que[now].x-1,que[now].y-1); }int main(){ int head = 1,tail = 1; for(int i = 1; i <= 5; i++) for(int j = 1; j <= 5; j++) scanf("%d",&map[i][j]); que[tail].x = 1; que[tail].y = 1; que[tail].f = 0; tail++; book[1][1] = 1; bool flag = 0; int tx,ty; while(head < tail){ for(int i = 0; i < 3; i++){ tx = que[head].x + nextt[i][0]; ty = que[head].y + nextt[i][1]; if(tx < 1||tx > 5||ty < 1||ty > 5) continue; if(map[tx][ty] == 0 && book[tx][ty] == 0){ book[tx][ty] = 1; que[tail].x = tx; que[tail].y = ty; que[tail].f = head; tail++; } if(tx == 5&&ty == 5){ flag = 1; break; } } if(flag) break; head++; } // for(int i = 1; i <= 36; i++){// printf("x:%d y:%d f:%d\r
",que[i].x,que[i].y,que[i].f);// } pp(tail-1); printf("(4, 4)\r
"); return 0;}Title: D-Dungeon MasterCode: #include<iostream>#include<cstring>#include<cstdio>using namespace std;int l,r,c;char map[35][35][35];bool book[35][35][35];int fx[6] = {-1,1,0,0,0,0};int fy[6] = {0,0,-1,1,0,0};int fz[6] = {0,0,0,0,1,-1};typedef struct step{ int x; int y; int z; int s;} step;step que[43000];step s,e;//血的教训，变量名千万不能粗心写错了，一个bug改一天；//地图不要忘记置0 //坐标不能搞错了，又改了几个小时int main(){ while(cin>>l>>r>>c&&l&&r&&c){ memset(book,0,sizeof(book)); for(int i = 1; i <= l; i++){ for(int j = 1; j <= r; j++){ for(int k = 1; k <= c; k++){ cin>>map[j][k][i]; if(map[j][k][i] == 'S'){ s.x = j; s.y = k; s.z = i; //这里的坐标气死我也 } else if(map[j][k][i] == 'E'){ e.x = j; e.y = k; e.z = i; } map[j][k][i] = map[j][k][i]==35; } } } // // for(int i = 1; i <= l; i++){// for(int j = 1; j <= r; j++){// for(int k = 1; k <= c; k++){// cout<<(int)map[j][k][i]<<" ";// }// cout<<endl;// }// } int head = 1,tail = 1; que[tail].x = s.x; que[tail].y = s.y; que[tail].z = s.z; que[tail].s = 0; tail++; book[s.x][s.y][s.z] = 1; bool flag = 0; int tx,ty,tz; while(head < tail){ for(int i = 0; i < 6; i++){ tx = que[head].x + fx[i]; ty = que[head].y + fy[i]; tz = que[head].z + fz[i]; if(tx < 1||tx > r||ty < 1||ty > c||tz < 1||tz >l) continue; if(map[tx][ty][tz] == 0 && book[tx][ty][tz] == 0){ book[tx][ty][tz] = 1; //这里的坐标气我+1 que[tail].x = tx; que[tail].y = ty; que[tail].z = tz; que[tail].s = que[head].s + 1; tail++; } if(tx == e.x&&ty == e.y&&tz == e.z){ cout<<"Escaped in "<<que[tail-1].s<<" minute(s)."<<endl; flag = 1; break; } } if(flag) break; head++; } if(!flag) cout<<"Trapped!"<<endl; } return 0;}Title: E-FliptileCode: #include<iostream>#include<cstring>#include<cstdio>using namespace std;int l,r,c;char map[35][35][35];bool book[35][35][35];int fx[6] = {-1,1,0,0,0,0};int fy[6] = {0,0,-1,1,0,0};int fz[6] = {0,0,0,0,1,-1};typedef struct step{ int x; int y; int z; int s;} step;step que[43000];step s,e;//血的教训，变量名千万不能粗心写错了，一个bug改一天；//地图不要忘记置0 //坐标不能搞错了，又改了几个小时int main(){ while(cin>>l>>r>>c&&l&&r&&c){ memset(book,0,sizeof(book)); for(int i = 1; i <= l; i++){ for(int j = 1; j <= r; j++){ for(int k = 1; k <= c; k++){ cin>>map[j][k][i]; if(map[j][k][i] == 'S'){ s.x = j; s.y = k; s.z = i; //这里的坐标气死我也 } else if(map[j][k][i] == 'E'){ e.x = j; e.y = k; e.z = i; } map[j][k][i] = map[j][k][i]==35; } } } // // for(int i = 1; i <= l; i++){// for(int j = 1; j <= r; j++){// for(int k = 1; k <= c; k++){// cout<<(int)map[j][k][i]<<" ";// }// cout<<endl;// }// } int head = 1,tail = 1; que[tail].x = s.x; que[tail].y = s.y; que[tail].z = s.z; que[tail].s = 0; tail++; book[s.x][s.y][s.z] = 1; bool flag = 0; int tx,ty,tz; while(head < tail){ for(int i = 0; i < 6; i++){ tx = que[head].x + fx[i]; ty = que[head].y + fy[i]; tz = que[head].z + fz[i]; if(tx < 1||tx > r||ty < 1||ty > c||tz < 1||tz >l) continue; if(map[tx][ty][tz] == 0 && book[tx][ty][tz] == 0){ book[tx][ty][tz] = 1; //这里的坐标气我+1 que[tail].x = tx; que[tail].y = ty; que[tail].z = tz; que[tail].s = que[head].s + 1; tail++; } if(tx == e.x&&ty == e.y&&tz == e.z){ cout<<"Escaped in "<<que[tail-1].s<<" minute(s)."<<endl; flag = 1; break; } } if(flag) break; head++; } if(!flag) cout<<"Trapped!"<<endl; } return 0;}Title: F-Find The MultipleCode: #include<iostream>#include<stdio.h>#include<queue>#include<string.h>#include<algorithm>#include<string>#include<stack>using namespace std;typedef long long ll;int n;void bfs(ll x){ queue<ll> Q; Q.push(1) ; while(!Q.empty()) { ll u = Q.front() ; Q.pop() ; if(u%n == 0) { cout<<u<<endl; return ; } Q.push(u\*10); Q.push(u\*10+1); } return ;} int main(){ while(cin>>n) { if(n==0) { break; } bfs(1); } }Title: G-Prime PathCode: #include <iostream>#include <cstdio>#include <cmath>#include <cstring>#include <algorithm>#include <string>#include <map>#include <set>#include <vector>#include <queue>#include <stack>#define inf 100000000using namespace std;typedef long long ll;int n, z, t; // n表示输入的第一个数，z表示输入的第二个数，t表示案例数；bool prime[10000]; //存素数表；int p[10000]; //记录变化次数，bfs过程用；int bfs(){ queue<int> q; q.push(n); p[n]=0; //标记n; while(!q.empty()) { int pa=q.front(); q.pop(); if(pa==z) //判断是否已经搜索到 z 了； return p[z]; int d[4]; d[0]=(pa/10)\*10; //pa去个位数的值； d[1]=pa%10+(pa/100)\*100; //pa去十位数的值； d[2]=(pa/1000)\*1000+pa%100; //pa去百位数的值； d[3]=pa%1000; //pa去千位数的值； int d2=1; for(int i=0;i<4;i++) { int j=0; if(i==3) //千位不为0处理； j=1; for(;j<10;j++) { int w=d[i]+j\*d2; if(w>=1000&&w<10000&&prime[w]==0&&p[w]==-1) { p[w]=p[pa]+1; if(w==z) return p[z]; q.push(w); } } d2=d2\*10; } } return -1;}int main(){ prime[0]=1; prime[1]=1; for(int i=2;i\*i<10000;i++) //打素数表，表中0表示素数； { if(!prime[i]) { for(int j=i\*2;j<10000;j=j+i) { prime[j]=1; } } } scanf("%d",&t); for(int o=0;o<t;o++) { memset(p,-1,sizeof(p)); //对p初始化； int za=0; //表示变化次数，输出用； scanf("%d %d",&n,&z); za=bfs(); if(za==-1) //没搜索到； { printf("Impossible\r
"); } else //搜索到了； { printf("%d\r
",za); } } return 0;}Title: H-Oil DepositsCode: #include<iostream>#include<cstring>#include<cstdio>using namespace std;bool map[101][101];bool book[101][101];int nextt[8][2] = {{0,1},{1,1},{1,0},{1,-1},{0,-1},{-1,-1},{-1,0},{-1,1}};int m,n,count;void dfs(int x, int y){ book[x][y] = 1; for(int i = 0; i < 8; i++){ int tx = x + nextt[i][0]; int ty = y + nextt[i][1]; if(tx<1 || tx>m || ty<1 || ty > n) continue; if(book[tx][ty] == 1 || map[tx][ty] == 1) continue; book[tx][ty] = 1; dfs(tx,ty); } return; }int main(){ while(scanf("%d %d",&m,&n)!=EOF&&m){ memset(book,0,sizeof(book)); char c;getchar();count = 0; for(int i = 1; i <= m; i++){ for(int j = 1; j <= n; j++){ c = getchar(); map[i][j] = c!=64; } getchar(); } for(int i = 1; i <= m; i++){ for(int j = 1; j <= n; j++){ if(map[i][j]==0&&!book[i][j]){ count++;dfs(i,j); } } } printf("%d\r
",count); } return 0;}Title: I-非常可乐Code: #include <bits/stdc++.h>using namespace std;int main(){ int a,b,c; while(cin>>a>>b>>c&&(a&&b&&c)) { a /= \_\_gcd(b,c); if(a&1) cout<<"NO"<<endl; else cout<<a-1<<endl; } return 0;}Title: J-Find a wayCode: #include<bits/stdc++.h>using namespace std;const int MAXN = 205;int n,m;int a[MAXN][MAXN];int vis[MAXN][MAXN];int vis\_t[MAXN][MAXN];int ans[MAXN][MAXN];int nxy[4][2] = {{-1, 0}, {1, 0}, {0, -1}, {0, 1}};struct node{ int x; int y; int step;};int isIn(int x, int y){ if(x > n || x < 1) return 0; if(y > m || y < 1) return 0; return 1;}int bfs(node dot, int flag){ for(int i = 1; i <= n; i++) for(int j = 1; j <= m; j++) vis\_t[i][j] = vis[i][j]; queue<node> que; dot.step = 0; que.push(dot); int step = 0; while(!que.empty()){ node t = que.front();// cout << "t:" << t.x <<" "<<t.y<<endl; if(a[t.x][t.y] == '@'){ if(flag == 1) ans[t.x][t.y] = 0; ans[t.x][t.y] += t.step;// ma[make\_pair(t.x, t.y)] += t.step;// cout <<"string:"<<t.step<<endl;// ma[ans] += step; } int flag = 0; for(int i = 0; i < 4; i++){ int tx = t.x + nxy[i][0]; int ty = t.y + nxy[i][1]; if(isIn(tx,ty)&& !vis\_t[tx][ty]){// cout <<"tx:"<<tx<<" ty:"<<ty<<" "<<step<<endl; step++; vis\_t[tx][ty] = 1; node needIn; needIn.x = tx; needIn.y = ty; needIn.step = t.step+1; que.push(needIn); } } que.pop(); }}int main(){ node yf, mj; int len = 1; while(cin >> n >> m){ memset(vis, 0, sizeof(vis)); for(int i = 1; i <= n; i++) for(int j = 1; j <= m; j++) ans[i][j] = 1e8; len = 1; for(int i = 1; i <= n; i++){ for(int j = 1; j <= m; j++){ char t; cin >> t; if(t == 'Y'){ yf.x = i; yf.y = j; vis[i][j] = 1; }else if(t == 'M'){ mj.x = i; mj.y = j; vis[i][j] = 1; }else if(t == '#'){ vis[i][j] = 1; } a[i][j] = t; } } a[yf.x][yf.y] = '#'; a[mj.x][mj.y] = '#';// cout << endl;// for(int i = 1; i <= n; i++){// for(int j = 1; j <= m; j++){// cout << ans[i][j] << " ";// }// cout << endl;// } bfs(yf,1);// cout << endl;// for(int i = 1; i <= n; i++){// for(int j = 1; j <= m; j++){// cout << ans[i][j] << " ";// }// cout << endl;// } bfs(mj,0);// cout << endl;// for(int i = 1; i <= n; i++){// for(int j = 1; j <= m; j++){// cout << ans[i][j] << " ";// }// cout << endl;// } int minn = 1e8; for(int i = 1; i <= n; i++) for(int j = 1; j <= m; j++){ minn = min(minn, ans[i][j]); } cout << minn\*11 << endl; } return 0;}

**四、实验结果（抓上机运行结果图）及心得体会**

**要求：运行结果图上的标题包含个人学号姓名信息。**

**五、教师评价**

**成 绩：（必填）**

**指导教师（签名）：（必填）**

**（封底页）**