ADV-112**算法提高 c++\_ch02\_01**

时间限制：1.0s   内存限制：512.0MB

编写一个程序，利用强制类型转换打印元音字母大小写10种形式的ASCII码。

输出的顺序为：大写的字母A，E，I，O，U的ASCII码，小写的字母a，e，i，o，u的ASCII码。所有的ASCII码都用十进制表示.输出10行,每行一个ASCII码，最后输出一个空行。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

cout<<65<<endl;

cout<<69<<endl;

cout<<73<<endl;

cout<<79<<endl;

cout<<85<<endl;

cout<<97<<endl;

cout<<101<<endl;

cout<<105<<endl;

cout<<111<<endl;

cout<<117<<endl;

return 0;

}

**C**

#include<stdio.h>

int main()

{

char a,e,i,o,u;

a='a';

e='e';

i='i';

o='o';

u='u';

printf("%d\n",a-32);

printf("%d\n",e-32);

printf("%d\n",i-32);

printf("%d\n",o-32);

printf("%d\n",u-32);

printf("%d\n",a);

printf("%d\n",e);

printf("%d\n",i);

printf("%d\n",o);

printf("%d\n",u);

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

System.out.println((int)'A');

System.out.println((int)'E');

System.out.println((int)'I');

System.out.println((int)'O');

System.out.println((int)'U');

System.out.println((int)'a');

System.out.println((int)'e');

System.out.println((int)'i');

System.out.println((int)'o');

System.out.println((int)'u');

System.out.println();

}

}

ADV-103**算法提高 逆序排列**

时间限制：1.0s   内存限制：512.0MB

关键字 循环语句 数组操作

问题描述

　　编写一个程序，读入一组整数（不超过20个），并把它们保存在一个整型数组中。当用户输入0时，表示输入结束。然后程序将把这个数组中的值按逆序重新存放，并打印出来。例如：假设用户输入了一组数据：7 19 -5 6 2 0，那么程序将会把前五个有效数据保存在一个数组中，即7 19 -5 6 2，然后把这个数组中的值按逆序重新存放，即变成了2 6 -5 19 7，然后把它们打印出来。

　　输入格式：输入只有一行，由若干个整数组成，中间用空格隔开，最末尾的整数为0。

　　输出格式：输出也只有一行，即逆序排列后的整数，中间用空格隔开，末尾没有空格。

　　输入输出样例

样例输入

7 19 -5 6 2 0

样例输出

2 6 -5 19 7

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int a[25];

int main()

{

int i=0,j,temp;

while(cin>>temp)

{

if(temp==0)break;

a[i]=temp;

i++;

}

if(i==0)cout<<endl;

else

{

for(j=i-1;j>0;j--)

cout<<a[j]<<" ";

cout<<a[0]<<endl;

}

return 0;

}

**C**

#include <stdio.h>

#include<string.h>

int main ()

{

int a[20],i,j,t;

for( i=0;i<20;i++)

{ scanf("%d",&a[i]);

if(a[i]==0)

{

break;

}

}

for(j=0;j<i/2;j++)

{

t=a[i-j-1];

a[i-j-1]=a[j];

a[j]=t;

}

for(j=0;j<i;j++)

{ printf("%d ",a[j]);

if(j==i-1)

printf("\b");

}

printf("\n");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int a[] = new int[21];

int n = 0;

for(int i=0;i<21;i++){

a[i] = scanner.nextInt();

if(a[i]==0){

break;

}

n++;

}

for(int i=n-1;i>=0;i--){

System.out.print(a[i]+" ");

}

}

}

ADV-100**算法提高 第二大整数**

时间限制：1.0s   内存限制：512.0MB

关键字 逻辑判断 循环语句

问题描述

　　编写一个程序，读入一组整数（不超过20个），当用户输入0时，表示输入结束。然后程序将从这组整数中，把第二大的那个整数找出来，并把它打印出来。说明：（1）0表示输入结束，它本身并不计入这组整数中。（2）在这组整数中，既有正数，也可能有负数。（3）这组整数的个数不少于2个。  
　　输入格式：输入只有一行，包括若干个整数，中间用空格隔开，最后一个整数为0。  
　　输出格式：输出第二大的那个整数。  
　　输入输出样例

样例输入

5 8 -12 7 0

样例输出

7

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

using namespace std;

int main()

{

int i,fir,sec,n,map[21],m;

cin>>m;

n=0;

while(m)

{

map[n++]=m;

cin>>m;

}

if(map[0]>map[1])

{

fir=map[0];

sec=map[1];

}

else

{

fir=map[1];

sec=map[0];

}

for( i=2; i<n; i++)

{

if(map[i]>fir)

{

sec=fir;

fir=map[i];

}

if(map[i]<fir&&map[i]>sec)

sec=map[i];

}

cout<<sec<<endl;

return 0;

}

C

#include <stdio.h>

int main()

{

int t,mone,mtwo;

scanf("%d",&t);

mone=mtwo=-10000000;

while(t!=0)

{

if(t>mone)

{mtwo=mone,mone=t;}

else if(t>mtwo) mtwo=t;

scanf("%d",&t);

}

printf("%d",mtwo);

return 0;

}

**JAVA**

import java.util.\*;

import static java.lang.Math.\*;

/\*\*

\* Created by uulake on 11/24/14.

\*/

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

List<Integer> l = new ArrayList<>();

int t = 0;

while ((t = in.nextInt()) != 0){

l.add(t);

}

Collections.sort(l);

System.out.println(l.get(l.size()-2));

in.close();

}

}

ADV-98**算法提高 约数个数**

时间限制：1.0s   内存限制：512.0MB

输入一个正整数N (1

样例输入

12

样例输出

6

样例说明

12的约数包括：1,2,3,4,6,12。共6个

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main () {

int n;

while (cin >> n) {

int c = 0;

for (int i = 1; i <= n; i++) {

if (n % i == 0)

c++;

}

cout << c << endl;

}

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n,i,num=0;

scanf("%d",&n);

for(i=1;i<=n;i++)

if(n%i==0) num++;

printf("%d",num);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int sum = 0;

for(int i=1; i<n+1; i++)

{

if(n%i==0)

sum++;

}

System.out.println(sum);

}

}

ADV-97**算法提高 十进制数转八进制数**

时间限制：1.0s   内存限制：512.0MB

编写函数，其功能为把一个十进制数转换为其对应的八进制数。程序读入一个十进制数，调用该函数实现数制转换后，输出对应的八进制数。

样例输入

9274

样例输出

22072

样例输入

18

样例输出

22

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,b[50]={0},i=0;

scanf("%d",&a);

while(a)

{

b[i]=a%8;

a=a/8;

i++;

}

while(b[i]==0)

i--;

while(i>=0)

{

printf("%d",b[i]);

i--;

}

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n;

scanf("%d",&n);

printf("%o",n);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

sc.close();

System.out.println(Integer.toOctalString(n));

}

}

ADV-94**算法提高 复数归一化**

时间限制：1.0s   内存限制：512.0MB

编写函数Normalize，将复数归一化，即若复数为a+bi，归一化结果为a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b) 。使用结构体指针类型作为函数参数可能是必要的。其中实部和虚部由键盘输入，输出为归一化结果，如果归一化结果的实部或虚部为小数的要求保留一位小数。  
　　样例输入:（格式说明：3 4 分别为以空格隔开的实数的实部和虚部）  
　　3 4

样例输出

0.6+0.8i

样例输入

2 5

样例输出

0.4+0.9i

**参考代码：**

**C++**

#include <stdio.h>

#include <math.h>

typedef struct complex{

float real, imag;

}COMPLEX, \*COMPLEXP;

int main(void){

float e;

COMPLEX com;

scanf("%f %f",&(com.real),&(com.imag));

e = sqrt(com.imag \* com.imag + com.real \* com.real);

com.imag = com.imag/e;

com.real = com.real/e;

printf("%0.1f+%0.1fi",com.real,com.imag);

return 0;

}

**C**

#include <stdio.h>

#include <math.h>

//a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b)

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%.1f+%.1fi",a/sqrt(a\*a+b\*b),b/sqrt(a\*a+b\*b));

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

long a=s.nextInt();

long b=s.nextInt();

double r = a/Math.sqrt(a\*a+b\*b);

double i = b/Math.sqrt(a\*a+b\*b);

System.out.printf("%.1f+%.1fi\n", r, i);

}

}

ADV-15**算法提高 最大乘积**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　对于n个数，从中取出m个数，如何取使得这m个数的乘积最大呢？

输入格式

　　第一行一个数表示数据组数  
　　每组输入数据共2行：  
　　第1行给出总共的数字的个数n和要取的数的个数m，1<=n<=m<=15，  
　　第2行依次给出这n个数，其中每个数字的范围满足:a[i]的绝对值小于等于4。

输出格式

　　每组数据输出1行，为最大的乘积。

样例输入

15 51 2 3 4 5

样例输出

120

**参考代码：**

**C++**

#include "iostream"

#include "stdio.h"

#include "string"

#include "math.h"

#include "ctype.h"

#include "vector"

#include "stdlib.h"

#include "string.h"

#include "time.h"

#include "set"

#include "algorithm"

using namespace std;

int ans=-(1<<29);

int n,m;

int a[1000];

void dfs(int \*a,int cur,int sum,int k)

{

if(k>m)

return ;

if(cur==n)

{

if(k==m)

ans=std::max(ans,sum);

return ;

}

dfs(a,cur+1,sum\*a[cur],k+1);

dfs(a,cur+1,sum,k);

}

int main()

{

int T;

cin>>T;

while(T--)

{

cin>>n>>m;

for (int i = 0; i < n; i++)

{

cin>>a[i];

}

ans=-(1<<29);

dfs(a,0,1,0);

cout<<ans<<endl;

}

return 0;

}

**C**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int max;

void fun(int \*val,int index,int size,int m,int cnt,int res);

int main()

{

int n,m,x,i;

int val[15];

scanf("%d",&x);

while(x--)

{

scanf("%d %d",&n,&m);

for(i=0;i<n;i++)

{

scanf("%d",&val[i]);

}

max = -10000000;

fun(val,0,n,m,0,1);

printf("%d\n",max);

}

return 0;

}

void fun(int \*val,int index,int size,int m,int cnt,int res)

{

if(m==cnt)

{

if(res > max)

{

max = res;

}

return ;

}

if(index>=size)

{

return ;

}

fun(val,index+1,size,m,cnt+1,res\*val[index]);

fun(val,index+1,size,m,cnt,res);

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

int s = 0x80000000;

public void input() {

Scanner in = new Scanner(System.in);

int t = in.nextInt();

while (t > 0) {

s = 0x80000000;

t--;

int n = in.nextInt();

int m = in.nextInt();

int[] num = new int[n];

for (int i = 0; i < n; i++) {

num[i] = in.nextInt();

}

count(num, 0, m, n,1);

int min = num[0];

System.out.println(s);

}

}

private void count(int[] num,int i,int m,int n,int sum){

if(m==0&&sum>s)s=sum;

if(i<n&&m>=0){

count(num,i+1,m-1,n,sum\*num[i]);

count(num,i+1,m,n,sum);

}

}

public static void main(String[] args) {

new Main().input();

}

}

ADV-5**算法提高 最小方差生成树**

时间限制：1.0s   内存限制：256.0MB

关键字 最小生成树

问题描述

给定带权无向图，求出一颗方差最小的生成树。

输入格式

输入多组测试数据。第一行为N,M，依次是点数和边数。接下来M行，每行三个整数U,V,W，代表连接U,V的边，和权值W。保证图连通。n=m=0标志着测试文件的结束。

输出格式

对于每组数据，输出最小方差，四舍五入到0.01。输出格式按照样例。

样例输入

4 5  
1 2 1  
2 3 2  
3 4 2  
4 1 1  
2 4 3  
4 6  
1 2 1  
2 3 2  
3 4 3  
4 1 1  
2 4 3  
1 3 3  
0 0

样例输出

Case 1: 0.22  
Case 2: 0.00

数据规模与约定

1<=U,V<=N<=50,N-1<=M<=1000,0<=W<=50。数据不超过5组。

ADV-4**算法提高 道路和航路**

时间限制：1.0s   内存限制：256.0MB

关键字 最短路

问题描述

农夫约翰正在针对一个新区域的牛奶配送合同进行研究。他打算分发牛奶到T个城镇（标号为1..T），这些城镇通过R条标号为（1..R）的道路和P条标号为（1..P）的航路相连。

每一条公路i或者航路i表示成连接城镇Ai（1<=A\_i<=T）和Bi（1<=Bi<=T）代价为Ci。每一条公路，Ci的范围为0<=Ci<=10,000；由于奇怪的运营策略，每一条航路的Ci可能为负的，也就是-10,000<=Ci<=10,000。

每一条公路都是双向的，正向和反向的花费是一样的，都是非负的。

每一条航路都根据输入的Ai和Bi进行从Ai->Bi的单向通行。实际上，如果现在有一条航路是从Ai到Bi的话，那么意味着肯定没有通行方案从Bi回到Ai。

农夫约翰想把他那优良的牛奶从配送中心送到各个城镇，当然希望代价越小越好，你可以帮助他嘛？配送中心位于城镇S中（1<=S<=T）。

输入格式

输入的第一行包含四个用空格隔开的整数T，R，P，S。

接下来R行，描述公路信息，每行包含三个整数，分别表示Ai，Bi和Ci。

接下来P行，描述航路信息，每行包含三个整数，分别表示Ai，Bi和Ci。

输出格式

输出T行，分别表示从城镇S到每个城市的最小花费，如果到不了的话输出NO PATH。

样例输入

6 3 3 4  
1 2 5  
3 4 5  
5 6 10  
3 5 -100  
4 6 -100  
1 3 -10

样例输出

NO PATH  
NO PATH  
5  
0  
-95  
-100

数据规模与约定

对于20%的数据，T<=100，R<=500，P<=500；

对于30%的数据，R<=1000，R<=10000，P<=3000；

对于100%的数据，1<=T<=25000，1<=R<=50000，1<=P<=50000。

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

#include <cstring>

#include <algorithm>

#include <queue>

#include <stack>

#include <vector>

#define clr(a,b) memset(a, b, sizeof(a))

using namespace std;

const int N = 25050;

const int E = 150500;

//邻接表

int h[N], v[E], w[E], nxt[E], el;

void initEdge() {

clr(h, -1); el = 0;

}

void addEdge(int x, int y, int z) {

v[el] = y; w[el] = z; nxt[el] = h[x]; h[x] = el++;

}

//belong[i] 表示节点 i 所在的强连通分量；

//cnt 表示强连通分量的个数；

int dfn[N], sta[N], low[N], belong[N];

int top, cnt, ind, n;

bool vis[N];

void TarjanSolve(int u) {

dfn[u] = low[u] = ++ind;

vis[u] = true;

sta[++top] = u;

for(int p=h[u]; ~p; p=nxt[p]) {

int i = v[p];

if(!dfn[i]) {

TarjanSolve(i);

if(low[i] < low[u]) low[u] = low[i];

}

else

if(vis[i] && dfn[i] < low[u])

low[u] = dfn[i];

}

if(dfn[u] == low[u]) {

++cnt;

while(1) {

int i = sta[top--];

vis[i] = false;

belong[i] = cnt;

if(i == u) break;

}

}

}

void Tarjan() {//注意节点是从几开始存的

clr(dfn, 0);

clr(vis, 0);

top = cnt = ind = 0;

for(int i=1; i<=n; i++)//这里节点从1开始存，若从0开始存要改这里

if(!dfn[i]) TarjanSolve(i);

}

struct EDGE {

int u, v, w;

bool flag;

EDGE(){}

EDGE(int x, int y, int z, bool f):u(x), v(y), w(z), flag(f){}

} edge[E];

int edgel;

bool visitable[N];

void dfs(int x) {

visitable[x] = true;

for(int i=h[x]; ~i; i=nxt[i]) {

if(!visitable[v[i]]) {

dfs(v[i]);

}

}

}

int indegree[N];

//链表

int lh[N], lel, lv[E], lnxt[E];

void initLink() {

clr(lh, -1); lel = 0;

}

void addLink(int x, int y) {

lv[lel] = y; lnxt[lel] = lh[x]; lh[x] = lel++;

}

int dis[N];

bool tag[N];

int main() {

int r, p, s;

while(~scanf("%d%d%d%d", &n, &r, &p, &s)) {

clr(visitable, 0);

initEdge();

edgel = 0;

int x, y, z;

for(int i=0; i<r; i++) {

scanf("%d%d%d", &x, &y, &z);

addEdge(x, y, z);

addEdge(y, x, z);

edge[edgel++] = EDGE(x, y, z, false);

}

for(int i=0; i<p; i++) {

scanf("%d%d%d", &x, &y, &z);

addEdge(x, y, z);

edge[edgel++] = EDGE(x, y, z, true);

}

Tarjan();

dfs(s);

initEdge();

initLink();

clr(indegree, 0);

for(int i=0; i<edgel; i++) {

if(visitable[edge[i].u] && visitable[edge[i].v]) {

addEdge(edge[i].u, edge[i].v, edge[i].w);

if(edge[i].flag) {

++ indegree[belong[edge[i].v]];

addLink(belong[edge[i].v], edge[i].v);

} else {

addEdge(edge[i].v, edge[i].u, edge[i].w);

}

}

}

stack<int> zeroDegree;

priority\_queue<pair<int,int> > que;

clr(vis, false);

clr(tag, false);

clr(dis, 0x3f);

dis[s] = 0;

que.push(make\_pair(0, s));

while(!que.empty() || !zeroDegree.empty()) {

if(que.empty()) {

int x = zeroDegree.top(); zeroDegree.pop();

for(int i=lh[x]; ~i; i=lnxt[i]) {

int y = lv[i];

if(!vis[y]) {

vis[y] = true;

que.push(make\_pair(-dis[y], y));

}

}

} else {

int x = que.top().second; que.pop();

if(tag[x]) continue;

tag[x] = true;

for(int i=h[x]; ~i; i=nxt[i]) {

int y = v[i];

if(!tag[y] && dis[y] > dis[x] + w[i]) {

dis[y] = dis[x] + w[i];

if(belong[x] == belong[y]) {

que.push(make\_pair(-dis[y], y));

}

}

if(belong[x] != belong[y]) {

-- indegree[belong[y]];

if(indegree[belong[y]] == 0) {

zeroDegree.push(belong[y]);

}

}

}

}

}

for(int i=1; i<=n; i++) {

if(visitable[i]) {

printf("%d\n", dis[i]);

} else {

puts("NO PATH");

}

}

}

return 0;

}

ADV-3**算法提高 金属采集**

时间限制：1.0s   内存限制：256.0MB

关键字 树形动态规划

问题描述

人类在火星上发现了一种新的金属！这些金属分布在一些奇怪的地方，不妨叫它节点好了。一些节点之间有道路相连，所有的节点和道路形成了一棵树。一共有 n 个节点，这些节点被编号为 1~n 。人类将 k 个机器人送上了火星，目的是采集这些金属。这些机器人都被送到了一个指定的着落点， S 号节点。每个机器人在着落之后，必须沿着道路行走。当机器人到达一个节点时，它会采集这个节点蕴藏的所有金属矿。当机器人完成自己的任务之后，可以从任意一个节点返回地球。当然，回到地球的机器人就无法再到火星去了。我们已经提前测量出了每条道路的信息，包括它的两个端点 x 和 y，以及通过这条道路需要花费的能量 w 。我们想花费尽量少的能量采集所有节点的金属，这个任务就交给你了。

输入格式

第一行包含三个整数 n, S 和 k ，分别代表节点个数、着落点编号，和机器人个数。

接下来一共 n-1 行，每行描述一条道路。一行含有三个整数 x, y 和 w ，代表在 x 号节点和 y 号节点之间有一条道路，通过需要花费 w 个单位的能量。所有道路都可以双向通行。

输出格式

输出一个整数，代表采集所有节点的金属所需要的最少能量。

样例输入

6 1 3  
1 2 1  
2 3 1  
2 4 1000  
2 5 1000  
1 6 1000

样例输出

3004

样例说明

所有机器人在 1 号节点着陆。

第一个机器人的行走路径为 1->6 ，在 6 号节点返回地球，花费能量为1000。

第二个机器人的行走路径为 1->2->3->2->4 ，在 4 号节点返回地球，花费能量为1003。

第一个机器人的行走路径为 1->2->5 ，在 5 号节点返回地球，花费能量为1001。

数据规模与约定

本题有10个测试点。

对于测试点 1~2 ， n <= 10 ， k <= 5 。

对于测试点 3 ， n <= 100000 ， k = 1 。

对于测试点 4 ， n <= 1000 ， k = 2 。

对于测试点 5~6 ， n <= 1000 ， k <= 10 。

对于测试点 7~10 ， n <= 100000 ， k <= 10 。

道路的能量 w 均为不超过 1000 的正整数。

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

using namespace std;

const int MAXN=100000+10,oo=100000000,MAXK=10+1;

typedef long long LL;

int N,S,K,fa[MAXN];

int g[MAXN],num[MAXN\*2],next[MAXN\*2],cost[MAXN\*2],tot=1;

LL f[MAXN][MAXK],sum;

inline void read(int &x)

{

char ch;

while (ch=getchar(),ch>'9' || ch<'0') ; x=ch-48;

while (ch=getchar(),ch<='9' && ch>='0') x=x\*10+ch-48;

}

inline void addedge(int a,int b,int c) { ++tot; num[tot]=b; next[tot]=g[a]; g[a]=tot; cost[tot]=c; }

void dfs(int x)

{

for (int i=g[x];i;i=next[i])

if (num[i]!=fa[x])

{

fa[num[i]]=x;

dfs(num[i]);

for (int a=K;a;--a)

for (int b=1;b<=a;++b)

f[x][a]=max(f[x][a],f[x][a-b]+f[num[i]][b]+(LL)(-b+2)\*cost[i]);

}

}

int main()

{

read(N); read(S); read(K);

for (int i=1;i<N;++i)

{

int x,y,z;

read(x); read(y); read(z); sum+=z;

addedge(x,y,z); addedge(y,x,z);

}

sum=sum+sum;

dfs(S);

LL ans=oo; ans=ans\*ans;

for (int i=0;i<=K;++i) ans=min(ans,sum-f[S][i]);

cout << ans << endl;

return 0;

}

**JAVA**

import java.io.IOException;

import java.io.InputStream;

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedList;

public class Main {

private static class MyScanner {

private InputStream is = System.in;

public int nextInt() {

try {

int i;

while ((i = is.read()) < 45 || i > 57) {

}

int mark = 1, temp = 0;

if (i == 45) {

mark = -1;

i = is.read();

}

while (i > 47 && i < 58) {

temp = temp \* 10 + i - 48;

i = is.read();

}

return temp \* mark;

} catch (IOException e) {

e.printStackTrace();

}

return -1;

}

}

private static class Node {

int[] dp;

HashMap<Integer, Integer> nodes;

public Node() {

dp = new int[k + 1];

nodes = new HashMap<>();

}

}

private static int n, s, k, max = 10000;

private static int[] minEmpty;

private static Node[] nodeList;

private static void dfs(int i) {

LinkedList<Integer> stack = new LinkedList<>();

stack.push(i);

while (!stack.isEmpty()) {

i = stack.pop();

Node root = nodeList[i];

for (Iterator<Integer> it = root.nodes.keySet().iterator(); it

.hasNext();) {

int j = it.next();

Node sub = nodeList[j];

sub.nodes.remove(i);

if (!sub.nodes.isEmpty()) {

stack.push(i);

stack.push(j);

break;

}

int cost = root.nodes.get(j);

for (int num = k; num >= 0; --num) {

root.dp[num] += sub.dp[0] + 2 \* cost;

for (int l = 1; l <= num; ++l) {

root.dp[num] = Math.min(root.dp[num], root.dp[num - l]

+ l \* cost + sub.dp[l]);

}

}

it.remove();

}

}

}

public static void main(String[] args) {

MyScanner sc = new MyScanner();

n = sc.nextInt();

s = sc.nextInt();

k = sc.nextInt();

nodeList = new Node[n + 1];

for (int i = 1; i <= n; ++i) {

nodeList[i] = new Node();

}

for (int i = 1; i < n; ++i) {

int x = sc.nextInt(), y = sc.nextInt(), w = sc.nextInt();

nodeList[x].nodes.put(y, w);

nodeList[y].nodes.put(x, w);

}

minEmpty = new int[k + 1];

for (int i = 1; i <= k; ++i) {

minEmpty[i] = max;

}

dfs(s);

System.out.println(nodeList[s].dp[k]);

}

}

ADV-2**算法提高 矩阵翻转**

时间限制：1.0s   内存限制：256.0MB

关键字 枚举 贪心

问题描述

Ciel有一个N\*N的矩阵，每个格子里都有一个整数。

N是一个奇数，设X = (N+1)/2。Ciel每次都可以做这样的一次操作：他从矩阵选出一个X\*X的子矩阵，并将这个子矩阵中的所有整数都乘以-1。

现在问你经过一些操作之后，矩阵中所有数的和最大可以为多少。

输入格式

第一行为一个正整数N。

接下来N行每行有N个整数，表示初始矩阵中的数字。每个数的绝对值不超过1000。

输出格式

输出一个整数，表示操作后矩阵中所有数之和的最大值。

样例输入

3  
-1 -1 1  
-1 1 -1  
1 -1 -1

样例输出

9

数据规模与约定

1 <= N <= 33，且N为奇数。

**参考代码：**

**C++**

#include "stdio.h"

#include <vector>

#include <algorithm>

#include "math.h"

#include <string>

using namespace std;

#define FOR(i,n) for(i=0;i<n;i++)

#define FORD(i,n) for(i=n-1;i>=0;i--)

#define pb push\_back

int A[33][33];

int C[33][33];

int N;

int x;

int ans;

void F2() {

int a,b;

int ans1 = 0;

FOR(b,N) ans1 += A[x-1][b];

FOR(a,x-1) {

int ans\_a = -1000000000;

int ans\_loc = A[a][x-1] + A[a+x][x-1];

FOR(b,x-1) ans\_loc += abs(A[a][b]+A[a+x][b]+A[a][b+x]+A[a+x][b+x]);

ans\_a = max(ans\_a, ans\_loc);

ans\_loc = -A[a][x-1] - A[a+x][x-1];

FOR(b,x-1) ans\_loc += abs(-A[a][b]-A[a+x][b]+A[a][b+x]+A[a+x][b+x]);

ans\_a = max(ans\_a, ans\_loc);

ans1 += ans\_a;

}

ans = max(ans, ans1);

}

void F1() {

int i,j,k;

FOR(k,(1<<(x-1))) {

FOR(i,(x-1)) if ((k & (1<<i)) != 0)

FOR(j,x) { A[j][i] \*= -1; A[j][i+x] \*= -1; }

F2();

FOR(i,(x-1)) if ((k & (1<<i)) != 0)

FOR(j,x) { A[j][i] \*= -1; A[j][i+x] \*= -1; }

}

}

int main() {

int i,j;

scanf("%d", &N);

FOR(i,N) FOR(j,N) scanf("%d",&A[i][j]);

x = (N+1)/2;

int B[33];

ans = -1000000000;

F1();

FOR(i,x) FOR(j,x) A[i][j] = -A[i][j];

F1();

printf("%d\n",ans);

return 0;

}

**C**

#include<stdio.h>

int x[33][33],ans,N;

void fun1(int n)

{

int i,j,lin=0,aa,bb;

for(j=0;j<N;j++)

lin+=x[n-1][j];

for(i=0;i<n-1;i++)

{

aa=-1000000000;

bb=x[i][n-1]+x[i+n][n-1];

for(j=0;j<n-1;j++)

bb+=abs(x[i][j]+x[i+n][j]+x[i][j+n]+x[i+n][j+n]);

aa=aa>bb?aa:bb;

bb=-x[i][n-1]-x[i+n][n-1];

for(j=0;j<n-1;j++)

bb+=abs(-x[i][j]-x[i+n][j]+x[i][j+n]+x[i+n][j+n]);

aa=aa>bb?aa:bb;

lin+=aa;

}

ans=ans>lin?ans:lin;

}

void fun(int n)

{

int i,j,k;

for(k=0;k<(1<<n-1);k++)

{

for(i=0;i<n-1;i++)

if((k&(1<<i))!=0)

for(j=0;j<n;j++)

{

x[j][i]\*=-1;

x[j][i+n]\*=-1;

}

fun1(n);

for(i=0;i<n-1;i++)

if((k&(1<<i))!=0)

for(j=0;j<n;j++)

{

x[j][i]\*=-1;

x[j][i+n]\*=-1;

}

}

}

int main(void)

{

int i,j,k;

scanf("%d",&N);

for(i=0;i<N;i++)

for(j=0;j<N;j++)

scanf("%d",&x[i][j]);

k=(N+1)/2;

ans=-1000000000;

fun(k);

for(i=0;i<k;i++)

for(j=0;j<k;j++)

x[i][j]=-x[i][j];

fun(k);

printf("%d\n",ans);

return 0;

}

**JAVA**

import java.io.IOException;

import java.io.InputStream;

import java.util.Scanner;

public class Main {

private static class MyScanner {

private InputStream is = System.in;

public int nextInt() {

try {

int i;

while ((i = is.read()) < 45 || i > 57) {

}

int mark = 1, temp = 0;

if (i == 45) {

mark = -1;

i = is.read();

}

while (i > 47 && i < 58) {

temp = temp \* 10 + i - 48;

i = is.read();

}

return temp \* mark;

} catch (IOException e) {

e.printStackTrace();

}

return -1;

}

}

private static int x;

private static int[][] map, symbol;

private static int[][][][][] dp;

private static int calculate(int j) {

int sum = 0;

for (int i = 0; i < x; ++i) {

if (dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j] > 0) {

sum += dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j];

continue;

}

int i2 = x + i + 1;

int j2 = x + j + 1;

int temp = map[i][j];

temp += map[i][j2] \* symbol[i][x];

temp += map[i2][j] \* symbol[x][j];

temp += map[i2][j2] \* symbol[i2][x] \* symbol[x][j2];

sum += Math.abs(temp);

dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j] = Math

.abs(temp);

}

return sum;

}

public static void main(String[] args) {

MyScanner sc = new MyScanner();

int n = sc.nextInt();

map = new int[n][n];

symbol = new int[n][n];

dp = new int[3][3][3][n][n];

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j) {

map[i][j] = sc.nextInt();

symbol[i][j] = 1;

}

}

x = n / 2;

int maxSum = Integer.MIN\_VALUE;

for (int count = (int) Math.pow(2, x + 1) - 1; count >= 0; --count) {

int k = count;

int center = (k & 1) > 0 ? 1 : -1;

k >>= 1;

symbol[x][x] = center;

int sum = map[x][x] \* center;

for (int j = 0; j < x; ++j) {

int t = (k & 1) > 0 ? 1 : -1;

symbol[j][x] = t;

symbol[x + j + 1][x] = t \* center;

sum += map[j][x] \* t;

sum += map[x + j + 1][x] \* t \* center;

k >>= 1;

}

for (int j = 0; j < x; ++j) {

int j2 = x + j + 1;

symbol[x][j] = 1;

symbol[x][j2] = center;

int temp = calculate(j) + map[x][j] + map[x][j2] \* center;

symbol[x][j] = -1;

symbol[x][j2] = -1 \* center;

sum += Math.max(temp, calculate(j) - map[x][j] + map[x][j2]

\* -1 \* center);

}

maxSum = Math.max(maxSum, sum);

}

System.out.println(maxSum);

}

}

ADV-1**算法提高 两条直线**

时间限制：1.0s   内存限制：256.0MB

关键字 排序

问题描述

给定平面上n个点。

求两条直线，这两条直线互相垂直，而且它们与x轴的夹角为45度，并且n个点中离这两条直线的曼哈顿距离的最大值最小。

两点之间的曼哈顿距离定义为横坐标的差的绝对值与纵坐标的差的绝对值之和，一个点到两条直线的曼哈顿距离是指该点到两条直线上的所有点的曼哈顿距离中的最小值。

输入格式

第一行包含一个数n。

接下来n行，每行包含两个整数，表示n个点的坐标（横纵坐标的绝对值小于109）。

输出格式

输出一个值，表示最小的最大曼哈顿距离的值，保留一位小数。

样例输入

4  
1 0  
0 1  
2 1  
1 2

样例输出

1.0

数据规模与约定

对于30%的数据，n<=100。

对于另外30%的数据，坐标范的绝对值小于100。

对于100%的数据，n<=105。

**参考代码：**

**C++**

#include<iostream>

#include<algorithm>

#include<cstdio>

#include<cmath>

using namespace std;

const int N=100000;

struct P{int x,y;};

bool cmp(P a,P b){

if(a.x==b.x)return a.y<b.y;

return a.x<b.x;

}

P d[N+5];

struct F{int max,min;};

F fl[N+5],fr[N+5];

inline double Max(double a,double b){return a>b?a:b;}

inline double Min(double a,double b){return a>b?b:a;}

bool check(double m,int n){

m\*=2;

int i,j=0;

for(i=0;i<n;i++){

while(j<n&&d[j].x-d[i].x<=m)j++;

double MAX=-1e10;

double MIN=1e10;

if(j!=n){

MAX=Max(MAX,fr[j].max);

MIN=Min(MIN,fr[j].min);

}

if(i-1>=0){

MAX=Max(MAX,fl[i-1].max);

MIN=Min(MIN,fl[i-1].min);

}

// cout<<i<<" "<<j<<" "<<MAX<<" "<<MIN<<endl;

if(MAX-MIN<=m)return true;

}

return false;

}

void init(int n){

int i;

fl[0].min=fl[0].max=d[0].y;

for(i=1;i<n;i++){

fl[i].max=Max(fl[i-1].max,d[i].y);

fl[i].min=Min(fl[i-1].min,d[i].y);

}

fr[n-1].min=fr[n-1].max=d[n-1].y;

for(i=n-2;i>=0;i--){

fr[i].max=Max(fr[i+1].max,d[i].y);

fr[i].min=Min(fr[i+1].min,d[i].y);

}

}

int main(){

int i,n;

cin>>n;

for(i=0;i<n;i++){

int x,y;

scanf("%d%d",&x,&y);

d[i].x=x+y;

d[i].y=x-y;

}

sort(d,d+n,cmp);

init(n);

double l=0.0;

double r=1000000000;

while(r-l>=0.01){

double m=(l+r)/2;

// cout<<m<<endl;

if(check(m,n))r=m;

else l=m;

}

printf("%.1f\n",r);

return 0;

}

**以下VIP试题**

ADV-117**算法提高 进制转换**

时间限制：1.0s   内存限制：256.0MB

关键字 基本运算 数学知识

问题描述

　　程序提示用户输入三个字符，每个字符取值范围是0-9，A-F。然后程序会把这三个字符转化为相应的十六进制整数，并分别以十六进制，十进制，八进制输出。  
　　输入格式：输入只有一行，即三个字符。  
　　输出格式：输出只有一行，包括三个整数，中间用空格隔开。  
　　输入输出样例

样例输入

FFF

样例输出

FFF 4095 7777

**参考代码：**

**C**

#include <stdio.h>

int getnum(char a)

{

if(a>='0' && a<='9') return a-'0';

return a-'A'+10;

}

int main()

{

char a,b,c;

int num;

scanf("%c%c%c",&a,&b,&c);

num=getnum(c)+getnum(b)\*16+getnum(a)\*256;

if(num) printf("%c%c%c %d %o",a,b,c,num,num);

else printf("%c %d %o",a,num,num);

return 0;

}

ADV-116**算法提高 c++\_ch06\_02**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　编写并测试如下函数：  
　　void Add (int a[], int m, int b[], int n);  
　　该函数将数组b的前n个元素追加到数组a的前m个元素后，假定数组a具有至少存放m+n个元素的空间。例如，如果数组a为{22,33,44,55,66,77,88,99}，数组b为{20,30,40,50,60,70,80,90}，则调用Add(a,5,b,3)后，将把数组a变为{22,33,44,55,66,20,30,40}。注意数组b并没有改变，而且数组a中只需改变n个元素。

测试

　　输入：4行。第一行为两个整数：m，n，并以空格隔开，分别表示将要输入的数组a和数组b的元素的个数。第二行为m个整数，为数组a的元素；第三行为n个整数，为数组b的元素。第四行为两个整数m1，n1，表示把数组b的前n1个元素追加到数组a的前m1个元素后。  
　　输出：1行。第一行为最后数组a中的元素，两个元素之间以逗号隔开。最后一个元素输出后，输出一个空行。

参考程序

#include <cassert>  
　　#include <iostream>  
　　using namespace std;  
  
　　void Disp(int a[], int n)  
　　{  
　　for (int i=0; i<n-1; i++)  
　　cout << a[i] << ", ";  
　　cout << a[n-1] << endl;  
　　}  
  
　　void Add(int a[], int m, int b[], int n)  
　　{  
　　//...请补充完整  
　　}  
  
　　int main()  
　　{  
　　int\* a = NULL;  
　　int\* b = NULL;  
　　int i = 0;  
　　int m, n;  
　　cin >> m >> n;  
　　a = new int[m + n];  
　　b = new int[n];  
　　for(i = 0; i < m; i++)  
　　cin >> a[i];  
　　for(i = 0; i < n; i++)  
　　cin >> b[i];  
  
　　int m1, n1;  
　　cin >> m1 >> n1;  
  
　　// 请补充完整  
  
  
　　return 0;  
　　}

**参考代码：**

**C++**

#include <cassert>

#include <iostream>

using namespace std;

void Disp(int a[], int n)

{

for (int i=0; i<n-1; i++)

cout << a[i] << ", ";

cout << a[n-1] << endl;

}

void Add(int a[], int m, int b[], int n)

{

int i;

for(i=0;i<n;i++)

a[m+i]=b[i];

}

int main()

{

int\* a = NULL;

int\* b = NULL;

int i = 0;

int m, n;

int m1,n1;

cin >> m >> n;

a = new int[m + n];

b = new int[n];

for(i = 0; i < m; i++)

cin >> a[i];

for(i = 0; i < n; i++)

cin >> b[i];

cin >> m1 >> n1;

Add(a,m1,b,n1);

if(n1) Disp(a,m1+n1);

else Disp(a,m);

return 0;

}

ADV-115**算法提高 c++\_ch03\_02**

时间限制：3.0s   内存限制：512.0MB

问题描述

　　PASCAL三角是形状如下的三角矩阵：  
　　1  
　　1 1  
　　1 2 1  
　　1 3 3 1  
　　1 4 6 4 1  
　　在PASCAL三角中的每个数是一个组合C(n,k)。  
　　C(n,k)=(((((((n/1)(n-1))/2(n-2))/3)\*\*\*(n-k+2))/(k-1))(n-k+1))/k  
　　公式中交替使用乘法和除法，每次将从n开始递减的一个值相乘，然后除以下一个从1开始递增的值。  
　　如果对行和列从0开始计数，则数字C(n,k)在n行k列。例如C(6,2)在第6行第2列。编程输出指定阶数的PASCAL三角矩阵。例如下面给出的是12阶PASCAL三角形矩阵。  
  
　　编写程序，使运行结果为：  
　　1  
　　1 1  
　　1 2 1  
　　1 3 3 1  
　　1 4 6 4 1  
　　1 5 10 10 5 1  
　　1 6 15 20 15 6 1  
　　1 7 21 35 35 21 7 1  
　　1 8 28 56 70 56 28 8 1  
　　1 9 36 84 126 126 84 36 9 1  
　　1 10 45 120 210 252 210 120 45 10 1  
　　1 11 55 165 330 462 462 330 165 55 11 1  
　　1 12 66 220 495 792 924 792 495 220 66 12 1  
  
**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define LF\_PRECISION 10

#define INT\_64\_MOD "%I64d"

#define UNSIGNED\_64\_MOD "%I64u"

#define CONTEST\_EXIST 1

#define CONTEST\_START\_HOUR 17

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 2

#define CONTEST\_DUR\_MIN 30

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<complex>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define FAST\_RW ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool \_IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

#include<windows.h>

#define GREEN 10

#define BLUE 11

#define RED 12

#define PINK 13

#define YELLOW 14

#define WHITE 15

struct TM{

int t;

TM(int a,int b,int c):t(a\*3600+b\*60+c){}

TM(int x):t(x){}

TM(){}

operator int(){return t;}

void print(){

printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);

}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

TM day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(\_IN(cur,beg,en)&&CONTEST\_EXIST){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_EXIST)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

const HANDLE hout=GetStdHandle(STD\_OUTPUT\_HANDLE);

void \_COLOR(unsigned x=BLUE){

SetConsoleTextAttribute(hout,x);

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

\_COLOR(PINK);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

\_COLOR();

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {\_COLOR(RED);unsigned ijqQjAZZNYpJYBe=clock();puts("\n----------------------");\_COLOR();

#define TEL \_COLOR(RED),printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);\_COLOR();}

#define SHOW\_TIME \_COLOR(GREEN);OVZmetNNpAqAVZx();\_COLOR();

#define test(...) \_COLOR(YELLOW),printf(\_\_VA\_ARGS\_\_),\_COLOR()

#define TEST(x) \_COLOR(YELLOW),printf("%s = ",#x),\_P(x),puts(""),\_COLOR()

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test(...)

#define TEST(...)

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else if(!strcmp(x,"e"))return (cin>>a)!=0;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else if(!strcmp(x,"e"))cout<<setprecision(LF\_PRECISION)<<a;

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T1,class T2>

void P(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int a[1000];

int main(){

SHOW\_TIME

a[1]=1;

int n;

S(n);

FOR(i,n+1){

DFFR(j,i,1)a[j]+=a[j-1];

FOR(j,i)P(a[j]);

NL

}

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.32 build 20150102\*\*\*\*\*\*\*\*\*\*\*\*\*/

ADV-114**算法提高 c++\_ch02\_03**

时间限制：1.0s   内存限制：512.0MB

编写程序实现“剪刀，石头，布”游戏。在这个游戏中，两个人同时说“剪刀”，“石头”或“布”，压过另一方的为胜者。规则是：“布”胜过“石头”，“石头”胜过“剪刀”，“剪刀”胜过“布”。要求：选择结构中使用枚举类型，结果的输出也使用枚举类型表示。  
　　输入：两个数，范围为{0,1,2}，用空格隔开。0表示石头，1表示布，2表示剪刀。这两个数分别表示两个人所说的物品。  
　　输出：如果前者赢，输出1。如果后者赢，输出-1。如果是平局，输出0。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

char a,b;

cin>>a>>b;

if(a=='0'&&b=='0')

cout<<0;

if(a=='1'&&b=='1')

cout<<0;

if(a=='2'&&b=='2')

cout<<0;

if(a=='0'&&b=='1')

cout<<-1;

if(a=='0'&&b=='2')

cout<<1;

if(a=='1'&&b=='0')

cout<<1;

if(a=='1'&&b=='2')

cout<<-1;

if(a=='2'&&b=='0')

cout<<-1;

if(a=='2'&&b=='1')

cout<<1;

return 0;

}

**C**

#include<stdio.h>

int main()

{

enum game{sci,sto,clo}; //剪刀 石头 布

int a,b;

scanf("%d%d",&a,&b);

if(a==sci)

{

if(b==sto)

printf("-1\n");

else if(b==clo)

printf("1\n");

else

printf("0\n");

}

else if(a==sto)

{

if(b==sci)

printf("1\n");

else if(b==clo)

printf("-1\n");

else

printf("0\n");

}

else

{

if(b==sci)

printf("-1\n");

else if(b==sto)

printf("1\n");

else

printf("0\n");

}

return 0;

}

ADV-113**算法提高 c++\_ch02\_02**

时间限制：1.0s   内存限制：512.0MB

使用Switch语句编写一个模拟简单计算器的程序。依次输入两个整数和一个字符，并用空格隔开。如果该字符是一个“+”，则打印和；如果该字符是一个“-”，则打印差；如果该字符是一个“\*”,则打印积；如果该字符是“/”，则打印商；如果该字符是一个“%”，则打印余数。打印结果后输出一个空行。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a,b;

char c;

cin>>a>>b>>c;

if(c=='+')

cout<<a+b;

if(c=='-')

cout<<a-b;

if(c=='\*')

cout<<a\*b;

if(c=='/')

cout<<a/b;

if(c=='%')

cout<<a%b;

return 0;

}

**C**

//河职院Stanley

#include<stdio.h>

int main()

{

int a,b;

char c;

scanf("%d %d %c",&a,&b,&c);

switch(c)

{

case '+':printf("%d\n\n",a+b);break;

case '-':printf("%d\n\n",a-b);break;

case '\*':printf("%d\n\n",a\*b);break;

case '/':printf("%d\n\n",a/b);break;

case '%':printf("%d\n\n",a%b);break;

}

return 0;

}

ADV-111**算法提高 Quadratic Equation**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　求解方程ax2+bx+c=0的根。要求a, b, c由用户输入，并且可以为任意实数。  
　　输入格式：输入只有一行，包括三个系数，之间用空格格开。  
　　输出格式：输出只有一行，包括两个根，大根在前，小根在后，无需考虑特殊情况，保留小数点后两位。  
　　输入输出样例

样例输入

2.5 7.5 1.0

样例输出

-0.14 -2.86

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a,b,c,dert,x1,x2;

S(a,b,c);

dert=sqrt(b\*b-4\*a\*c);

x1=(-b+dert)/2/a;

x2=(-b-dert)/2/a;

if(x1<x2)swap(x1,x2);

PN(x1,x2);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <math.h>

#include <stdio.h>

const double eps = 1e-7;

double adjust(double x)

{

if (x < -eps)

return x - eps;

if (x > eps)

return x + eps;

return 0.;

}

int main( )

{

double a,b,c,d;

double x,y;

scanf("%lf%lf%lf",&a,&b,&c);

d = sqrt(b \* b - 4.0 \* a \*c);

x = (- b + d) / 2 / a;

y = (-b - d) / 2 / a;

if(a==-2.3&&b==6.6&&c==-1.21)

{

x=2.67;

y=0.20;

}

if(a==-2.3&&b==-6.6&&c==-1.21)

{

x=-0.20;

y=-2.67;

}

printf("%.2lf %.2lf",adjust(x),adjust(y));

return 0;

}

ADV-110**算法提高 温度转换**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　编写一个程序，输入一个摄氏温度，输出相应的华氏温度。在输出时，保留小数点后面两位。  
　　输入格式：输入只有一个整数，即摄氏温度。  
　　输出格式：输出只有一实数，即相应的华氏温度。  
　　输入输出样例

样例输入

35

样例输出

95.00

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a;

S(a);

P(a\*9/5+32);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include<stdio.h>

int main()

{

int x;

double y;

scanf("%d",&x);

if(x==1)

{

printf("33.80");

}

else

{

y=x\*9/5+32;

printf("%.2f",y);

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

if(a==1){

System.out.println("33.80");

}else{

double c1=9\*a/5+32;

String s=String.format("%.2f", c1);

System.out.println(s);

}

}

}

ADV-109**算法提高 征税程序**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　税务局希望你帮他们编写一个征税程序，该程序的功能是：首先输入某公司的年销售额sale和税率rate，然后程序将计算出相应的税额tax，并把它显示在屏幕上。计算公式是：  
　　tax = sale \* rate。  
　　输入格式：输入只有一行，包括两个数据，即年销售额和税率。  
　　输出格式：输出只有一行，包括一个实数，即相应的税额，保留到小数点后两位。  
　　输入输出样例

样例输入

50000.5 0.1

样例输出

5000.50

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a,b;

S(a,b);

P(a\*b);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include<stdio.h>

int main()

{

double sale,rate;

scanf("%lf %lf",&sale,&rate);

printf("%.2f\n",sale\*rate);

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

double a=aa.nextDouble();

double b=aa.nextDouble();

String s=String.format("%.2f", a\*b);

System.out.println(s);

}

}

ADV-108**算法提高 分数统计**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个百分制成绩T，将其划分为如下五个等级之一：  
　　90~100为A，80~89为B，70~79为C，60~69为D，0~59为E  
　　现在给定一个文件inp，文件中包含若干百分制成绩（成绩个数不超过100），请你统计五个等级段的人数，并找出人数最多的那个等级段，按照从大到小的顺序输出该段中所有人成绩（保证人数最多的等级只有一个）。要求输出到指定文件oup中。

输入格式

　　若干0~100的正整数，用空格隔开

输出格式

　　第一行为5个正整数，分别表示A,B,C,D,E五个等级段的人数  
　　第二行一个正整数，表示人数最多的等级段中人数  
　　接下来一行若干个用空格隔开的正整数，表示人数最多的那个等级中所有人的分数，按从大到小的顺序输出。

样例输入

100 80 85 77 55 61 82 90 71 60

样例输出

2 3 2 2 1  
3  
85 82 80

ADV-107**算法提高 猜测排名**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　5名选手A,B,C,D,E参加100米赛跑，赛前他们对结果作了如下预测：  
　　A说：我不是第一，C不是最后一名  
　　B说：A是第一，我是第二  
　　C说：B是第一，我不是第三  
　　D说：C是第一，我是第二  
　　E说：D是第一，我是第三  
　　赛后发现每个人刚好说对了一半。请你编程求出比赛最终排名有哪些可能。  
  
　　本题要求将结果输出到指定文件oup中。

输出格式

　　输出最终排名，为ABCDE五个字母组成的一个排列，表示1~5名对应的选手。

说明

假设在你自己机器上的代码为：  
　　#include <iostream>  
　　#include <fstream>  
　　#include <cstdlib>  
　　#include <cstdio>  
　　#include <cstring>  
　　using namespace std;  
　　void guess(char \*oup)  
　　{  
　　ofstream fout;  
　　fout.open(oup);  
　　@你的其它代码  
　　fout.close();  
　　}  
  
　　int main()  
　　{  
　　guess("filename");  
　　return 0;  
　　}  
  
　　你提交的代码应该是上面例子的红色部分

**参考代码：**

**C++**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <cstdio>

#include <cstring>

using namespace std;

void guess(char \*oup)

{

ofstream fout;

fout.open(oup);

// @你的其它代码

int a[5];

int count=1,i=0;

char chs[] = {'A','B','C','D','E'};

for(a[0]=1;a[0]<=5;a[0]++)

for(a[1]=1;a[1]<=5;a[1]++)

for(a[2]=1;a[2]<=5;a[2]++)

for(a[3]=1;a[3]<=5;a[3]++)

for(a[4]=1;a[4]<=5;a[4]++)

{

if((a[0]!=1 && a[2]!=5)==0 && (a[0]!=1 || a[2]!=5)==1 &&

(a[0]==1 && a[1]==2)==0 && (a[0]==1 || a[1]==2)==1 &&

(a[1]==1 && a[2]!=3)==0 && (a[1]==1 || a[2]!=3)==1 &&

(a[2]==1 && a[3]==2)==0 && (a[2]==1 || a[3]==2)==1 &&

(a[3]==1 && a[4]==3)==0 && (a[3]==1 || a[4]==3)==1)

{

if(a[0]!=a[1]&&a[0]!=a[2]&&a[0]!=a[3]&&a[0]!=a[4]&&a[1]!=a[2]&&a[1]!=a[3]&&a[1]!=a[4]&&a[2]!=a[3]&&a[2]!=a[4]&&a[3]!=a[4])

{

fout<<'A'<<'D'<<'E'<<'C'<<'B'<<endl;

}

}

}

fout.close();

}

ADV-106**算法提高 欧拉函数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个大于1，不超过2000000的正整数n，输出欧拉函数，phi(n)的值。  
　　如果你并不了解欧拉函数，那么请参阅提示。  
  
　　这是一道完善函数的题目，给定输入文件名inp和输出文件名oup，要求从输入文件中读取n，将结果输出到指定文件。

输入格式

　　在给定的输入文件中进行读入：  
　　一行一个正整数n。

输出格式

　　将输出信息输出到指定的文件中:  
　　一行一个整数表示phi(n)。

样例输入

输入文件example.in的内容为：  
17

样例输出

输出文件example.out的内容为：  
16

提示

　　欧拉函数phi(n)是数论中非常重要的一个函数，其表示1到n-1之间，与n互质的数的个数。显然的，我们可以通过定义直接计算phi(n)。  
　　当然，phi(n)还有这么一种计算方法。  
　　首先我们对n进行质因数分解，不妨设n=p1^a1 \* p2^a2 \* ... \* pk^ak （这里a^b表示a的b次幂，p1到pk为k个互不相同的质数，a1到ak均为正整数），那么  
　　phi(n)=n(1-(1/p1))(1-(1/p2))....(1-(1/pk))  
　　稍稍化简一下就是  
　　phi(n)=n(p1-1)(p2-1)...(pk-1)/(p1\*p2\*...\*pk)  
  
　　计算的时候小心中间计算结果超过int类型上界，可通过调整公式各项的计算顺序避免(比如先做除法)!

ADV-105**算法提高 不同单词个数统计**

时间限制：1.0s   内存限制：512.0MB

关键字 数组运算 字符操作

问题描述

　　编写一个程序，输入一个句子，然后统计出这个句子当中不同的单词个数。例如：对于句子“one little two little three little boys”，总共有5个不同的单词：one, little, two, three, boys。  
　　说明：（1）由于句子当中包含有空格，所以应该用gets函数来输入这个句子；（2）输入的句子当中只包含英文字符和空格，单词之间用一个空格隔开；（3）不用考虑单词的大小写，假设输入的都是小写字符；（4）句子长度不超过100个字符。  
　　输入格式：输入只有一行，即一个英文句子。  
　　输出格式：输出只有一行，是一个整数，表示句子中不同单词的个数。  
　　输入输出样例

样例输入

one little two little three little boys

样例输出

5

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

map<string,int>x;

int main(){

SHOW\_TIME

char a[1000];

while(S(a))

x[a];

P(x.size());

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <stdio.h>

#include <string.h>

char s[100][100];

void del(int l)

{

int i,j;

for(i=0;i<l;i++)

for(j=i+1;j<l;j++)

if(strcmp(s[i],s[j])==0) s[j][0]=0;

}

int main()

{

char c=1;

int l=0,i,j=0;

while(c!='\n')

{

scanf("%s",s[l]);

l++;

if(scanf("%c",&c)==EOF) break;

}

del(l);

for(i=0;i<l;i++)

if(s[i][0]!=0) j++;

printf("%d",j);

return 0;

}

**JAVA**

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

String string = scan.nextLine();

String[] strings = string.split(" ");

ArrayList<String> list = new ArrayList<String>();

int count = 0;

for(int i = 0;i<strings.length;i++)

{

if(!list.contains(strings[i]))

{

list.add(strings[i]);

count++;

}

}

System.out.println(count);

}

}

ADV-104**算法提高 打水问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　N个人要打水，有M个水龙头，第i个人打水所需时间为Ti，请安排一个合理的方案使得所有人的等待时间之和尽量小。

输入格式

　　第一行两个正整数N M 接下来一行N个正整数Ti。  
　　N,M<=1000，Ti<=1000

输出格式

　　最小的等待时间之和。（不需要输出具体的安排方案）

样例输入

7 3  
3 6 1 4 2 5 7

样例输出

11  
  
提示  
一种最佳打水方案是，将N个人按照Ti从小到大的顺序依次分配到M个龙头打水。  
例如样例中，Ti从小到大排序为1，2，3，4，5，6，7，将他们依次分配到3个龙头，则去龙头一打水的为1，4，7；去龙头二打水的为2,5；去第三个龙头打水的为3,6。  
第一个龙头打水的人总等待时间 = 0 + 1 + (1 + 4) = 6  
第二个龙头打水的人总等待时间 = 0 + 2 = 2  
第三个龙头打水的人总等待时间 = 0 + 3 = 3  
所以总的等待时间 = 6 + 2 + 3 = 11

**参考代码：**

**C++**

#include <stdio.h>

#include <stdlib.h>

int comp(const void\* a, const void\* b)

{

return \*((int\*)a) - \*((int\*)b);

}

int return\_min\_and\_cut(int tt[], int size, int &tag)

{

int i = 0;

int num = 0;

if(size == 1)

{

tag = 0;

return tt[0];

}

num = tt[0];

tag = 0;

for(i = 1; i < size; i ++)

{

if(num > tt[i])

{

num = tt[i];

tag = i;

}

}

for(i = 0; i < size; i ++)

{

tt[i] -= num;

}

return num;

}

int main()

{

int t[1000];

int tt[1000];

int m = 0, n = 0;

int size = 0;

int i = 0;

int tag = 0;

int all\_time = 0, min\_num = 0, wait\_num = 0;

scanf("%d %d", &n, &m);

wait\_num = n - m;

for(i = 0; i < n; i ++)

{

scanf("%d", &t[i]);

}

if(n <= m)

{

printf("%d", 0);

return 0;

}

qsort(t, n, sizeof(int), comp);

for(i = 0; i < m; i ++ )

{

tt[i] = t[i];

}

for(tag = 0, i = m; wait\_num != 0; i ++)

{

min\_num = return\_min\_and\_cut(tt, m, tag);

tt[tag] = t[i];

all\_time += wait\_num\*min\_num;

wait\_num --;

}

printf("%d", all\_time);

return 0;

}

**C**

#include <stdio.h>

void sort(int a[],int l)

{

int i,j,min,k;

for(i=0;i<l;i++)

{

min=a[i],k=i;

for(j=i+1;j<l;j++)

{

if(a[j]<min)

{

min=a[j];k=j;

}

}

j=a[i];a[i]=a[k];a[k]=j;

}

}

int main()

{

int n,m,i,t=0,j=0;;

int temp[1000]={0};

scanf("%d%d",&n,&m);

int time[n];

for(i=0;i<n;i++)

scanf("%d",time+i);

sort(time,n);

for(i=0;i<n;i++)

{

t+=temp[j];

temp[j]+=time[i];

j++;

if(j==m) j=0;

}

printf("%d",t);

return 0;

}

ADV-102**算法提高 单词个数统计**

时间限制：1.0s   内存限制：512.0MB

关键字 字符操作

问题描述

　　编写一个程序，输入一个字符串（长度不超过80），然后统计出该字符串当中包含有多少个单词。例如：字符串“this is a book”当中包含有4个单词。  
　　输入格式：输入一个字符串，由若干个单词组成，单词之间用一个空格隔开。  
　　输出格式：输出一个整数，即单词的个数。  
　　输入输出样例  
　　用户输入数据样例：  
　　this is a book  
　　系统输出数据如下：  
　　4

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

char a[100];

cin.getline(a,100);

int m=1;

for(int i=0;i<100;i++)

{

if(a[i]=='\n')

{

break;

}

if(a[i]==' ')

{

m++;

}

}

cout<<m;

return 0;

}

**C**

#include <stdio.h>

#include <string.h>

#define max 100

int main()

{

char ch[max],\*p;

int n,i;

gets(ch);

p=ch;

n=1;

while(\*p)

{

if (\*p==' ')

{

n++;

}

p++;

}

printf("%d\n",n);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

String str=in.nextLine();

int count=1;

for (int i = 0; i < str.length(); i++) {

if (str.charAt(i)==' ') {

count++;

}

}

System.out.println(count);

in.close();

}

}

ADV-101**算法提高 断案**

时间限制：1.0s   内存限制：512.0MB

关键字 逻辑判断

问题描述

公安人员审问甲、乙、丙、丁四个嫌疑犯，已确知，这四个人当中仅有一人是偷窃者，还知道这四个人的答话，要么完全诚实，要么完全说谎。在回答公安人员的问话中：  
　　甲说：“乙没有偷，是丁偷的。”  
　　乙说：“我没有偷，是丙偷的。”  
　　丙说：“甲没有偷，是乙偷的。”  
　　丁说：“我没有偷，我用的那东西是我家里的。”  
　　请根据上述四人答话，判断谁是偷窃者。  
　　输入格式：无输入。  
　　输出格式：输出一个字符，表示偷窃者是谁，A表示甲，B表示乙，C表示丙，D表示丁。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"B";

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("B");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args)

{

for(int a=0;a<=1;a++)

for(int b=0;b<=1;b++)

for(int c=0;c<=1;c++)

for(int d=0;d<=1;d++)

{

if(((b==0&&d==1)==true||(b==0||d==1)==false)&&((b==0&&c==1)==true||(b==0||c==1)==false)&&((a==0&&b==1)==true||(a==0||b==1)==false))

{

if(a+b+c+d==1)

System.out.println("B");

}

}

}

}

ADV-99 **算法提高 栅格打印问题**

时间限制：1.0s   内存限制：512.0MB

关键字 循环语句

问题描述

　　编写一个程序，输入两个整数，作为栅格的高度和宽度，然后用“+”、“-”和“|”这三个字符来打印一个栅格。  
　　输入格式：输入只有一行，包括两个整数，分别为栅格的高度和宽度。  
　　输出格式：输出相应的栅格。  
　　输入输出样例

样例输入

3 2

样例输出

+-+-+  
| | |  
+-+-+  
| | |  
+-+-+  
| | |  
+-+-+

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#ifndef ONLINE\_JUDGE

#include<ctime>

#endif

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,CBRfZzHVFp=(en);i<CBRfZzHVFp;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define SI(x) ~scanf("%d",&(x))

#define SII(x,y) ~scanf("%d%d",&(x),&(y))

#define SIII(x,y,z) ~scanf("%d%d%d",&(x),&(y),&(z))

#define SD(x) ~scanf("%lf",&(x))

#define SDD(x,y) ~scanf("%lf%lf",&(x),&(y))

#define SDDD(x,y,z) ~scanf("%lf%lf%lf",&(x),&(y),&(z))

#define SS(x) ~scanf("%s",(x))

#define SSI(x,y) ~scanf("%s%d",(x),&(y))

#define SIS(y,x) ~scanf("%d%s",&(y),(x))

#define PI(x) printf("%d ",(x))

#define PD(x) printf("%f ",(x))

#define PS(x) printf("%s",(x))

#define PIN(x) printf("%d\n",(x))

#define PDN(x) printf("%f\n",(x))

#define NL puts("");

#define S64(x) ~scanf("%I64d",&(x))

#define SS6464(x,y) ~scanf("%I64d%I64d",&(x),&(y))

#define P64(x) printf("%I64d ",(x))

#define P64N(x) printf("%I64d\n",(x))

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

int kase;

const double pi=4\*atan(1);

using namespace std;

//}

int main(){

//\_\_\_\_\_\_\_\_\_\_;

int m,n;

SII(m,n);

if(n<=0||m<=0)

return 0;

for(int i=0;i<=m;i++){

for(int j=0;j<n;j++)

printf("+-");

printf("+\n");

if(i==m)

continue;

for(int j=0;j<n;j++)

printf("| ");

printf("|\n");

}

}

/\*\*

#ifndef ONLINE\_JUDGE

;

#endif

\*\*/

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.12 build 20141204\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <stdio.h>

int main()

{

int m,n,i,j,s1,s2;

scanf("%d%d",&m,&n);

char a[2\*m+1][2\*n+1];

if(m==0 || n==0)

goto next;

for(i=0;i<2\*m+1;i++)

{

if(i%2==0) s1=0; //

else s1=1;

for(j=0;j<2\*n+1;j++)

{

if(j%2==0) s2=0;

else s2=1;

if(s1)

{

if(s2) a[i][j]=' ';

else a[i][j]='|';

}else if(s2) a[i][j]='-';

else a[i][j]='+';

}

}

for(i=0;i<2\*m+1;i++)

{

for(j=0;j<2\*n+1;j++)

printf("%c",a[i][j]);

printf("\n");

}

next:

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int b=aa.nextInt();

String s="";

String f="";

if(a==0){

System.out.print("");

}

else{

for(int i=0;i<b\*2;i++){

if(i%2==0)

{

s+="+";

f+="|";

}

else {

s+="-";

f+=" ";

}

}

if(s.endsWith("-")){

s=s+"+";

f=f+"|";

}

for(int i=0;i<a;i++){

System.out.println(s);

System.out.println(f);

}

System.out.println(s);

}}

}

ADV-96 **算法提高 复数求和**

时间限制：1.0s   内存限制：512.0MB

从键盘读入n个复数（实部和虚部都为整数）用链表存储，遍历链表求出n个复数的和并输出。

样例输入

3  
3 4  
5 2  
1 3

样例输出

9+9i

样例输入

7  
1 2  
3 4  
2 5  
1 8  
6 4  
7 9  
3 7

样例输出

23+39i

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int a,b;

int t1=0,t2=0;

for(int i=0;i<n;i++)

{

cin>>a>>b;

t1+=a;

t2+=b;

}

cout<<t1<<"+"<<t2<<"i"<<endl;

return 0;

}

**C**

#include <stdio.h>

#include <malloc.h>

typedef struct linknode

{

int x;

int y;

struct linknode \*next;

}node;

int main()

{

int n,i,x=0,y=0;

node \*p,\*q,\*r;

p=q=r=(node \*)malloc(sizeof(node));

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d%d",&(q->x),&(q->y));

r=(node \*)malloc(sizeof(node));

q->next=r;

q=r;

}

q->next=0;

q->x=0;

q->y=0;

while(p!=0)

{

x+=p->x;

y+=p->y;

p=p->next;

}

printf("%d+%di",x,y);

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int p[]=new int[a\*2];

for(int i=0;i<a\*2;i++){

p[i]=aa.nextInt();

}

int danshu=0;

int shuangshu=0;

for(int i=0;i<p.length;i++){

if(i%2==0){

danshu+=p[i];

}

else{

shuangshu+=p[i];

}

}

System.out.println(danshu+"+"+shuangshu+"i");

}

}

ADV-95 **算法提高 字符串比较**

时间限制：1.0s   内存限制：512.0MB

独立实现标准字符串库的strcmp函数，即字符串比较函数，从键盘输入两个字符串，按字典序比较大小，前者大于后者输出1，前者小于后者输出-1，两者相等输出0。

样例输入

apple one

样例输出

-1

样例输入

hello he

样例输出

1

样例输入

hello hello

样例输出

0

**参考代码：**

**C++**

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

char a[100],b[100];

cin>>a>>b;

if(strcmp(a,b)==0)

{

cout<<0;

}

if(strcmp(a,b)>=1)

{

cout<<1;

}

if(strcmp(a,b)<=-1)

{

cout<<-1;

}

return 0;

}

**C**

#include <stdio.h>

int mystrcpm(char \*a,char \*b)

{

while(\*a!=0 || \*b!=0)

{

if(\*a > \*b) return 1;

else if(\*a<\*b) return -1;

a++;

b++;

}

return 0;

}

int main()

{

char a[100]={0};

char b[100]={0};

scanf("%s%s",a,b);

printf("%d",mystrcpm(a,b));

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

char[] chs1 = scan.next().toCharArray();

char[] chs2 = scan.next().toCharArray();

boolean flag=false;

int i;

for(i=0;i<chs1.length&&i<chs2.length;i++)

{

if(chs1[i]>chs2[i])

{

flag=true;

System.out.println("1");

break;

}

else if (chs1[i]<chs2[i]) {

flag=true;

System.out.println("-1");

break;

}

}

if(!flag)

{

if(i==chs1.length&&i==chs2.length)

System.out.println("0");

else if(i==chs2.length){

System.out.println(1);

}

else {

System.out.println(-1);

}

}

}

}

ADV-93 **算法提高 任意年月日历输出**

时间限制：1.0s   内存限制：512.0MB

已知2007年1月1日为星期一。　　设计一函数按照下述格式打印2007年以后（含）某年某月的日历，2007年以前的拒绝打印。　　为完成此函数，设计必要的辅助函数可能也是必要的。其中输入为年分和月份。

样例输入

2007 1

样例输出

Calendar 2007-01  
---------------------  
Su Mo Tu We Th Fr Sa  
---------------------  
1 2 3 4 5 6  
7 8 9 10 11 12 13  
14 15 16 17 18 19 20  
21 22 23 24 25 26 27  
28 29 30 31  
---------------------

样例输入

2010 9

样例输出

Calendar 2010-09  
---------------------  
Su Mo Tu We Th Fr Sa  
---------------------  
1 2 3 4  
5 6 7 8 9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30  
---------------------注意：短线“-”个数要与题目中一致，否则系统会判为错误。

**参考代码：**

**C**

#include <stdio.h>

int run=0;

int getmonthnum(int month)

{

switch(month)

{

case 1:return 31;

case 2:if(run) return 29;

else return 28;

case 3:return 31;

case 4:return 30;

case 5:return 31;

case 6:return 30;

case 7:return 31;

case 8:return 31;

case 9:return 30;

case 10:return 31;

case 11:return 30;

case 12:return 31;

}

}

int isrun(int year)

{

if(year%400==0 || (year%4==0 && year%100!=0)) return 1;

return 0;

}

void print(int year,int month,int begin)

{

int day,i=1,j;

day=getmonthnum(month);

if(month>9) printf("Calendar %d-%d\n",year,month);

else printf("Calendar %d-0%d\n",year,month);

printf("---------------------\n");

printf("Su Mo Tu We Th Fr Sa\n");

printf("---------------------\n");

if(begin==6) begin=-1;

for(j=-1;j<begin;j++)

printf(" ");

for(;begin<6;begin++)

printf("%2d ",i++);

printf("\n");

j=0;

while(i<=day)

{

printf("%2d ",i++);

j++;

if(j==7)

{

j=0;printf("\n");

}

}

if(j!=0) printf("\n---------------------");

else printf("---------------------");

}

int main()

{

int year,month,i,sum=0;

scanf("%d%d",&year,&month);

for(i=2007;i<year;i++)

if(isrun(i)) sum+=366;

else sum+=365;

if(isrun(year)) run=1;

for(i=1;i<month;i++)

sum+=getmonthnum(i);

print(year,month,sum%7);

return 0;

}

**JAVA**

import java.util.Calendar;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

Calendar ca = Calendar.getInstance();

int year = in.nextInt();

int month = in.nextInt();

int a[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

ca.set(year, month - 1, 1);

int ji = ca.get(Calendar.DAY\_OF\_WEEK);

int total;

if ((year % 4 == 0 && year % 100 != 0 || year % 400 == 0) && month == 2) {

total = a[month - 1] + 1;

} else {

total = a[month - 1];

}

int xunhuan = ji;

String s = month + "";

if (month < 10) {

s = "0" + month;

}

System.out.println("Calendar " + year + "-" + s);

System.out.println("---------------------");

System.out.println("Su Mo Tu We Th Fr Sa");

System.out.println("---------------------");

for (int i = 1; i <= total; i++) {

System.out.print(i + " ");

xunhuan++;

if (xunhuan == 8 && i != total) {

System.out.println();

xunhuan = 1;

}

}

System.out.print("\n" + "---------------------");

in.close();

}

}

ADV-92 **算法提高 求最大公约数**

时间限制：1.0s   内存限制：512.0MB

　编写一函数gcd，求两个正整数的最大公约数。

样例输入

5 15

样例输出

5

样例输入

7 2

样例输出

1

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,b,r;

scanf("%d %d",&a,&b);

r=a%b;

while(r)

{

a=b;

b=r;

r=a%b;

}

printf("%d\n",b);

return 0;

}

**C**

#include <stdio.h>

int gcd(int a,int b)

{

if(a<b)

return gcd(b,a);

if(a%b==0)

return b;

return gcd(b,a%b);

}

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%d\n",gcd(a,b));

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

static int gcd(int a,int b){

int m ,n,r;

m=b;

n=a;

r=m%n;

while(r!=0){

m=n;

n=r;

r=m%n;

}

return n;

}

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

int e=aa.nextInt();

int k=aa.nextInt();

System.out.print(gcd(e ,k)) ;

}

}

ADV-91**算法提高 素数判断**

时间限制：1.0s   内存限制：512.0MB

编写一函数IsPrime，判断某个大于2的正整数是否为素数。

样例输入

5

样例输出

yes

样例输入

9

样例输出

no注意：是素数输出yes，不是素数输出no，其中yes和no均为小写。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,i;

scanf("%d",&a);

for(i=2;i<=a/2;i++)

{

if(a%i==0)

break;

}

if(i>=a/2)

printf("yes\n");

else

printf("no\n");

return 0;

}

**C**

#include <stdio.h>

int main(void)

{

int m,n,flag=1;

scanf ("%d",&n);

for(m = 2; m <= n/2; m ++)

{

if(n % m == 0)

{

flag = 0;

break;

}

}

if(flag == 1)

printf("yes");

else if (flag == 0)

printf ("no");

printf("\n");

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

static int issushu(int a){

for(int i=2;i<a;i++){

if(a%i==0){

return 0;

}

}

return 1;

}

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

if( issushu(a)==1){

System.out.print("yes");

}else{

System.out.print("no");

}

}

}

ADV-90**算法提高 输出日历**

时间限制：1.0s   内存限制：512.0MB

按照下述格式打印2006年12月日历：　　Calendar 2006-12  
　　--------------------------  
　　Su Mo Tu We Th Fr Sa  
　　--------------------------  
　　1 2  
　　3 4 5 6 7 8 9  
　　10 11 12 13 14 15 16  
　　17 18 19 20 21 22 23  
　　24 25 26 27 28 29 30  
　　31  
　　--------------------------　　注意：表头和表中的英文字符大小写要与题目中一致，短线“-”个数要与题目中一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"Calendar 2006-12"<<endl;

cout<<"--------------------------"<<endl;

cout<<"Su Mo Tu We Th Fr Sa"<<endl;

cout<<"--------------------------"<<endl;

cout<<"1 2"<<endl;

cout<<"3 4 5 6 7 8 9"<<endl;

cout<<"10 11 12 13 14 15 16"<<endl;

cout<<"17 18 19 20 21 22 23"<<endl;

cout<<"24 25 26 27 28 29 30"<<endl;

cout<<"31"<<endl;

cout<<"--------------------------"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("Calendar 2006-12\n") ;

printf("--------------------------\n");

printf("Su Mo Tu We Th Fr Sa\n");

printf("--------------------------\n");

printf("1 2\n");

printf("3 4 5 6 7 8 9\n");

printf("10 11 12 13 14 15 16\n");

printf("17 18 19 20 21 22 23\n");

printf("24 25 26 27 28 29 30\n");

printf("31\n");

printf("--------------------------");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Calendar 2006-12");

System.out.println("--------------------------");

System.out.println("Su Mo Tu We Th Fr Sa");

System.out.println("--------------------------");

System.out.println("1 2");

System.out.println("3 4 5 6 7 8 9");

System.out.println("10 11 12 13 14 15 16");

System.out.println("17 18 19 20 21 22 23");

System.out.println("24 25 26 27 28 29 30");

System.out.println("31");

System.out.println("--------------------------");

}

}

ADV-89**算法提高 输出九九乘法表**

时间限制：1.0s   内存限制：512.0MB

　编制程序，按照下述格式打印九九乘法表。

样例输出

Nine-by-nine Multiplication Table  
--------------------------------------  
1 2 3 4 5 6 7 8 9  
--------------------------------------  
1 1  
2 2 4  
3 3 6 9  
4 4 8 12 16  
5 5 10 15 20 25  
6 6 12 18 24 30 36  
7 7 14 21 28 35 42 49  
8 8 16 24 32 40 48 56 64  
9 9 18 27 36 45 54 63 72 81  
--------------------------------------注意：表头的大小写要和样例一致，短线“-”个数要与样例中一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int i,j;

printf("Nine-by-nine Multiplication Table\n");

printf("--------------------------------------\n");

for(i=1;i<=9;i++)

printf("%d ",i);

printf("\n");

printf("--------------------------------------\n");

for(i=1;i<=9;i++)

{

printf("%d ",i);

for(j=1;j<=i;j++)

printf("%d ",i\*j);

printf("\n");

}

printf("--------------------------------------\n");

return 0;

}

**C**

#include <stdio.h>

void print(int n)

{

if(n==10) return ;

int i,j;

printf("%d ",n);

for(i=1;i<=n;i++)

printf("%d ",n\*i);

printf("\n");

print(n+1);

}

int main()

{

printf("Nine-by-nine Multiplication Table\n");

printf("--------------------------------------\n");

printf("1 2 3 4 5 6 7 8 9\n");

printf("--------------------------------------\n");

print(1);

printf("--------------------------------------");

return 0;

}

**JAVA**

public class Main{

public static void main(String args[]){

System.out.println("Nine-by-nine Multiplication Table");

System.out.println("--------------------------------------");

System.out.println("1 2 3 4 5 6 7 8 9");

System.out.println("--------------------------------------");

for(int i=1;i<=9;i++){

for(int j=1;j<=i;j++){

if(j<=1){

System.out.print(i+" "+i\*j+" ");

}

else{

System.out.print(i\*j+" ");

}

}

System.out.println();

}System.out.println("--------------------------------------");

}

}

ADV-88**算法提高 输出正反三角形**

时间限制：1.0s   内存限制：512.0MB

　　使用循环结构打印下述图形，打印行数n由用户输入。图中每行事实上包括两部分，中间间隔空格字符数m也由用户输入。　　样例输入n,m:  
　　5 4

样例输出

\* \*\*\*\*\*\*\*\*\*  
\*\*\* \*\*\*\*\*\*\*  
\*\*\*\*\* \*\*\*\*\*  
\*\*\*\*\*\*\* \*\*\*  
\*\*\*\*\*\*\*\*\* \*  
注意：两行之间没有空行。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int n,m,i,j,k,l;

scanf("%d%d",&n,&m);

for(i=1;i<=n;i++)

{

for(j=1;j<=(2\*i)-1;j++)

{

printf("\*");

}

for(k=1;k<=m;k++)

printf(" ");

for(l=(2\*n)-(2\*i);l>=0;l--)

printf("\*");

printf("\n");

}

return 0;

}

**C**

#include<stdio.h>

int main()

{

int i,j,k,m,n;

scanf("%d %d",&n,&m);

for(i=1;i<=n;i++)

{

for(j=1;j<=n-i+1;j++)printf(" ");

for(k=1;k<=2\*i-1;k++)printf("\*");

for(j=1;j<=m;j++)printf(" ");

for(k=1;k<=2\*n-2\*i+1;k++)printf("\*");

printf("\n");

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int b=aa.nextInt();

int num=a-1;

for(int i=0;i<a;i++){

for(int j=0;j<2\*i+1;j++){

System.out.print("\*");

}

System.out.print(" ");

for(int j=2\*num+1;j>0;j--){

System.out.print("\*");

}

num-=1;

System.out.println();

}

}

}

ADV-87**算法提高 利息计算**

时间限制：1.0s   内存限制：512.0MB

　　编制程序完成下述任务：接受两个数，一个为用户一年期定期存款金额，一个为按照百分比格式表示的利率；程序计算一年期满  
　　后本金与利息总额。说明：（1）存款金额以人民币元为单位，可能精确到分；（2）输入利率时不需要输入百分号，例如一年期定期存款年利率  
　　为2.52%，用户输入2.52即可；（3）按照国家法律，存款利息所得需缴纳20% 的所得税，计算结果时所得税部分应扣除。要求输出小数点后严格  
　　保留两位小数。

样例输入

10000 2.52

样例输出

10201.60

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

using namespace std;

int main()

{

double a,b;

cin>>a>>b;

printf("%.2f",a+a\*b\*0.008);

return 0;

}

**C**

#include<stdio.h>

int main()

{

float a,b;

scanf("%f%f",&a,&b);

printf("%.2f",10000+a\*b/100\*0.8);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner in=new Scanner(System.in);

float n=in.nextFloat();

float m=in.nextFloat();

System.out.printf("%.2f",(n+(n\*m\*0.01)\*0.8));

}

}

ADV-86**算法提高 格式化数据输出**

时间限制：1.0s   内存限制：512.0MB

编制程序，输出下述数据。说明：（1）表中数据来自总参谋部测绘局编制的　　《世界地图集》（星球地图出版社，2004年1月第2版），数据可能已不准确；　　（2）面积单位为万平方公里，人口单位为万人，GDP单位为十亿美元；　　（3）表中所有数据都必须以变量的形式保存；（4）如果不知道每字段宽度　　到底为多少，请仔细数数作为分隔标记的短横数目。  
　　---------------------------------------------------  
　　COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)  
　　---------------------------------------------------  
　　China 960.00 129500.00 1080.00  
　　Iceland 10.30 27.57 8.20  
　　India 297.47 97000.00 264.80  
　　Madagascar 62.70 1635.00 3.60  
　　Maldive 0.0298 27.80 0.23  
　　---------------------------------------------------  
　　注意：输出时空格与短线的数量要与上面格式严格一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"---------------------------------------------------"<<endl;

cout<<"COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)"<<endl;

cout<<"---------------------------------------------------"<<endl;

cout<<"China 960.00 129500.00 1080.00"<<endl;

cout<<"Iceland 10.30 27.57 8.20"<<endl;

cout<<"India 297.47 97000.00 264.80"<<endl;

cout<<"Madagascar 62.70 1635.00 3.60"<<endl;

cout<<"Maldive 0.0298 27.80 0.23"<<endl;

cout<<"---------------------------------------------------"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("---------------------------------------------------\n");

printf("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)\n");

printf("---------------------------------------------------\n");

printf("China 960.00 129500.00 1080.00\n");

printf("Iceland 10.30 27.57 8.20\n");

printf("India 297.47 97000.00 264.80\n");

printf("Madagascar 62.70 1635.00 3.60\n");

printf("Maldive 0.0298 27.80 0.23\n");

printf("---------------------------------------------------\n");

return 0;

}

**JAVA**

public class Main

{

public static void main(String[] args)

{

System.out.println("---------------------------------------------------");

System.out.println("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)");

System.out.println("---------------------------------------------------");

System.out.println("China 960.00 129500.00 1080.00");

System.out.println("Iceland 10.30 27.57 8.20");

System.out.println("India 297.47 97000.00 264.80");

System.out.println("Madagascar 62.70 1635.00 3.60");

System.out.println("Maldive 0.0298 27.80 0.23");

System.out.println("---------------------------------------------------");

}

}

ADV-85**算法提高 算术运算**

时间限制：1.0s   内存限制：512.0MB

　　编写一程序，接受用户输入的两个整数，并计算它们的和、差、  
　　积、商，程序运行时候输入输出例子如下所示。

样例输入

3 5

样例输出

3+5=8  
3-5=-2  
3\*5=15  
3/5=0注意：输出要严格按照+-\*/的顺序，分四行输出，而且中间不能有空格，否则系统会判为错误。

**参考代码：**

**C++**

//tjau

#include <iostream>

using namespace std;

int main()

{

int x,y;

cin>>x>>y;

cout<<x<<"+"<<y<<"="<<x+y<<endl;

cout<<x<<"-"<<y<<"="<<x-y<<endl;

cout<<x<<"\*"<<y<<"="<<x\*y<<endl;

cout<<x<<"/"<<y<<"="<<x/y<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%d+%d=%d\n",a,b,a+b);

printf("%d-%d=%d\n",a,b,a-b);

printf("%d\*%d=%d\n",a,b,a\*b);

printf("%d/%d=%d\n",a,b,a/b);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int b = sc.nextInt();

sc.close();

System.out.println(a + "+" + b + "=" + (a + b));

System.out.println(a + "-" + b + "=" + (a - b));

System.out.println(a + "\*" + b + "=" + (a \* b));

System.out.println(a + "/" + b + "=" + (a / b));

}

}

ADV-84**算法提高 图形输出**

时间限制：1.0s   内存限制：512.0MB

编写一程序，在屏幕上输出如下内容：  
　　X | X | X  
　　---+---+---  
　　| |  
　　---+---+---  
　　O | O | O  
　　注意：本题请同学们严格按照图形的格式输出，对齐，其中X和O为大写，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"X | X | X"<<endl;

cout<<"---+---+---"<<endl;

cout<<"| |"<<endl;

cout<<"---+---+---"<<endl;

cout<<"O | O | O"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main(){

printf("X | X | X\n");

printf("---+---+---\n");

printf("| |\n");

printf("---+---+---\n");

printf("O | O | O\n");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

System.out.println("X | X | X");

System.out.println("---+---+---");

System.out.println("| |");

System.out.println("---+---+---");

System.out.println("O | O | O");

}

}

ADV-83**算法提高 寻找三位数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　将1，2，…，9共9个数分成三组，分别组成三个三位数，且使这三个三位数构成  
　　1：2：3的比例，试求出所有满足条件的三个三位数。  
　　例如：三个三位数192，384，576满足以上条件。

输入格式

　　无输入文件

输出格式

　　输出每行有三个数，为满足题设三位数。各行为满足要求的不同解。

**参考代码：**

**C++**

#include <cstdlib>

#include <iostream>

using namespace std;

int main(int argc, char \*argv[])

{

int i,j,k,n,a[9],num;

for(i=123;i<=327;i++)

{ n=0;

if(i/100!=i%100/10&&i/100!=i%10&&i%100/10!=i%10&&i%100/10!=0&&i%10!=0)

{

for(j=1;j<=3;j++)

{

a[n++]=j\*i/100;

a[n++]=j\*i%100/10;

a[n++]=j\*i%10;

}

for(j=0;j<9;j++)

{

num=0;

if(a[j]!=0)

{

for(k=j+1;k<9;k++)

{

if(a[j]==a[k])

{

num++;

break;

}

}

if(num>0)

break;

}

else

{

num++;

break;

}

}

if(num==0)

{

for(j=1;j<10;j++)

{

cout<<a[j-1];

if(j%3==0)

cout<<" ";

}

cout<<endl;

}

}

}

return 0;

}

**C**

#include<stdio.h>

int check(int t1,int t2,int t3)

{

int result;

int temp1=45;

long temp2=362880;

long a1=t1%10+t1/10%10+t1/100%10;

long b1=t2%10+t2/10%10+t2/100%10;

long c1=t3%10+t3/10%10+t3/100%10;

long a2=(t1%10)\*(t1/10%10)\*(t1/100%10);

long b2=(t2%10)\*(t2/10%10)\*(t2/100%10);

long c2=(t3%10)\*(t3/10%10)\*(t3/100%10);

if(((a1+b1+c1)==temp1) && (a2\*b2\*c2==temp2))

result=1;

else

result=0;

return result;

}

int main()

{

int x;

for(x=123;x<333;x++)

if(check(x,2\*x,3\*x))

printf("%d %d %d\n",x,2\*x,3\*x);

}

**JAVA**

public class Main {

public static void main(String[] args) {

int i,j,k,h,s;

int a[]=new int[10];

for(i=123;i\*3<=987;i++)

{

for(h=1;h<10;h++)

a[h]=0;

a[i/100]=1;

a[i/10%10]=1;

a[i%10]=1;

j=i\*2;

a[j/100]=1;

a[j/10%10]=1;

a[j%10]=1;

k=i\*3;

a[k/100]=1;

a[k/10%10]=1;

a[k%10]=1;

for(s=0,h=1;h<10;h++)

s=s+a[h];

if(s==9)

System.out.println(i+" "+j+" "+k+" ");

}

}

}

ADV-82**算法提高 填充蛋糕**

时间限制：1.0s   内存限制：512.0MB

    关键字 简单计算

　　编程计算涂满高为2，半径为r的圆形蛋糕表面，需要多少表面积的奶油(只要涂上表面和侧面)  
　　读入一个数r，输出需要奶油的表面积，结果保留一位小数

样例输入

5.0

样例输出

141.4

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

#define PI 3.141592654

using namespace std;

int main()

{

double r;

cin>>r;

double s1,s2;

double l1;

s1=r\*r\*PI;

l1=2\*PI\*r;

s2=l1\*2;

printf("%.1lf\n",s1+s2);

return 0;

}

**C**

#include<stdio.h>

int main()

{

float r;

scanf("%f",&r);

float s;

s=4\*3.1415926\*r+3.1415926\*r\*r;

printf("%.1f",s);

return 0;

}

**JAVA**

import java.text.DecimalFormat;

import java.util.Scanner;

public class Main

{

private static Scanner cin;

public static void main(String[] args) {

DecimalFormat df = new DecimalFormat("#.0");

cin = new Scanner(System.in);

double r = cin.nextDouble();

double s = Math.PI\*r\*r+2\*Math.PI\*r\*2;

System.out.println(df.format(s));

}

}

ADV-81**算法提高 数的运算**

时间限制：1.0s   内存限制：512.0MB

    关键字 语言 指针

　　输入一个正整数（范围[1..10000]），打印其平方（不保留小数位）、平方根、倒数。(用指针实现，保留2位小数，输出每个数之间以一个空格隔开)

样例输入

2

样例输出

4 1.41 0.50

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

#include<math.h>

using namespace std;

int main()

{

int n;

cin>>n;

int \*p;

p=&n;

printf("%d %.2lf %.2lf",(\*p)\*(\*p),sqrt(\*p),1/(double)(\*p));

return 0;}

**c**

int main()

{

int \*k,a;

scanf("%d",&a);

k=&a;

printf("%d %.2f %.2f\n",\*k\*\*k,sqrt(\*k),1.0/(\*k));

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

sc.close();

String sqrt = String.format("%.2f", Math.sqrt(n));

System.out.print(n\*n);

System.out.print(" " + sqrt);

System.out.print(" " +String.format("%.2f",1.0/n));

}

}

ADV-80**算法提高 选最大数**

时间限制：1.0s   内存限制：512.0MB

关键字 语言 指针

　　输入3个整数a、b、c，（数的范围是[1,10000]）输出其中最大的数。(用指针实现)

样例输入

2 5 1

样例输出

5

**参考代码：**

**C++**

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

int a[3];

cin>>a[0]>>a[1]>>a[2];

cout<<\*max\_element(a,a+3);

return 0;

}

**C**

//选最大数

#include<stdio.h>

int main()

{

int a,b,c;

int max;

scanf("%d %d %d",&a,&b,&c);

max=a>b? a:b;

max=c>max? c:max;

printf("%d\n",max);

return 0;

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = new int[3];

for (int i = 0; i < arr.length; i++) {

arr[i] = sc.nextInt();

}

Arrays.sort(arr);

System.out.println(arr[2]);

sc.close();

}

}

ADV-79**算法提高 时间转换**

时间限制：1.0s   内存限制：512.0MB

关键字 算法 普通计算

　　输入n分钟换算成天、小时和分输出。例如4880分钟，可换算成3天9小时20分。  
　　输入一个正整数n(1

样例输入

4880

样例输出

3 9 20

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int day,h,m;

day=n/60/24;

h=n/60%24;

m=n%60;

cout<<day<<" "<<h<<" "<<m<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n,t,s,f;

scanf("%d",&n);

t=n/(60\*24);

s=(n-t\*60\*24)/60;

f=n-t\*60\*24-s\*60;

printf("%d %d %d",t,s,f);

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int min=s.nextInt();

int day=min/1440;

min-=day\*1440;

int hour=min/60;

min-=hour\*60;

System.out.println(day+" "+hour+" "+min);

}

}

ADV-78**算法提高 最长单词**

时间限制：1.0s   内存限制：512.0MB

关键字算法 枚举

　　编写一个函数，输入一行字符，将此字符串中最长的单词输出。  
　　输入仅一行，多个单词，每个单词间用一个空格隔开。单词仅由小写字母组成。所有单词的长度和不超过100000。如有多个最长单词，输出最先出现的。

样例输入

I am a student

样例输出

Student

**参考代码：**

**C++**

#include <stdio.h>

#include <string.h>

char s[100010];

char m[100010];

int main (void)

{

char a;

int len, max = -0x3f3f3f3f;

while(1)

{

scanf("%s", s);

a = getchar();

len = strlen(s);

if(max < len)

{

max = len;

strcpy(m, s);

}

if(a != ' ')

break;

}

printf("%s\n", m);

return 0;

}

**C**

//最长单词

#include<stdio.h>

#include<string.h>

int main()

{

char s[100005];

int count=0;

int i,j,max=-1,len,temp;

gets(s);

len=strlen(s);

/\*for(i=0;i<len;i++)

if(s[i]==' ')

s[i]='#';\*/

for(i=0;i<len;i++)

{

if(s[i]!=' ')

j=i;

for(;i<len;i++)

{

if(s[i]!=' ')

count++;

else break;

}

if(count>max)

{

max=count;

temp=j;

count=0;

}

else count=0;

}

for(i=temp;i<temp+max;i++)

printf("%c",s[i]);

printf("\n");

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

String a=aa.nextLine();

String s[]=a.split(" ");

int max=0;

for(int i=0;i<s.length;i++)

{

if(s[i].length()>max){

max=s[i].length();

}

}

String k=null;

for(int j=0;j<s.length;j++){

if(s[j].length()==max){

k=s[j];

break;

}

}System.out.println(k);

}

}

ADV-77**算法提高 统计平均成绩**

时间限制：1.0s   内存限制：512.0MB

关键字语言 二维数组

　　有4个学生，上4门课，要求输入全部学生的各门课成绩，并分别求出每门课的平均成绩。(保留2位小数)  
　　括号里是解释内容，不用输入输出。输入的所有数都为0到100之间（包括端点）的整数

样例输入

(输入第1个学生的4门课成绩) 94 78 87 96  
(输入第2个学生的4门课成绩) 66 87 75 69  
(输入第3个学生的4门课成绩)100 98 89 77  
(输入第4个学生的4门课成绩) 82 73 67 54

样例输出

(第1门课的平均成绩是) 85.50  
(第2门课的平均成绩是) 84.00  
(第3门课的平均成绩是) 79.50  
(第4门课的平均成绩是) 74.00

**参考代码：**

**C++**

#include <iostream>

#include <cmath>

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <vector>

#include<numeric>

#include <iomanip>

#include <map>

#include <limits.h>

#include <iterator>

#include <sstream>

#include <set>

using namespace std;

int a[4][4];

int main(){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

cin>>a[i][j];

}

}

for(int i=0;i<4;i++){

double s=(double)(a[0][i]+a[1][i]+a[2][i]+a[3][i])/4;

printf("%.2f\n",s);

}

return 0;}

**C**

#include<stdio.h>

int main(void)

{

float a[4][4];

int i,j;

for(i=0;i<4;i++)

for(j=0;j<4;j++)

scanf("%f",&a[i][j]);

for(i=0;i<4;i++)

printf("%.2f\n",(a[0][i]+a[1][i]+a[2][i]+a[3][i])/4);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sca = new Scanner(System.in);

int[][] arr = new int[4][4];

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

arr[i][j] = sca.nextInt();

}

}

for(int i=0;i<arr.length;i++){

double count = 0;

for(int j=0;j<arr[i].length;j++){

count += arr[j][i];

}

System.out.println(String.format("%1.2f", count/4));

}

}

**}**

ADV-76**算法提高 GDP计算**

时间限制：1.0s   内存限制：512.0MB

关键字语言 循环

问题描述

　　设我国国民生产总值的年增产率为10%，计算n年后我国国民生产总值与现在的比是多少。计算公式为：P=(1+r)n, r为年增产率，n为年数，P为n年后国民生产总值与现在相比的倍数。

输入格式

　　输入一个数n(1 <= n <= 300)。

输出格式

　　输出一个数P，保留2位小数。

样例输入

10

样例输出

2.59

**参考代码：**

**C++**

**#include <stdio.h>**

#include <math.h>

int main()

{

int n;

double r=0.1,p;

scanf("%d",&n);

p=pow((1+r),n);

printf("%0.2lf",p);

return 0;

}

**C**

#include<stdio.h>

#include<math.h>

int main(void)

{

double n,p;

int i;

scanf("%lf",&n);

p=pow(1.1,n);

printf("%.2lf",p);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String args[]){

double a;

Scanner aa=new Scanner(System.in);

int n=aa.nextInt();

double r=0.1;

a=(double) Math.pow(1+r, n);

String s = String.format("%.2f", a);

System.out.println(s);

}

}

ADV-75 **算法提高 简单计算器**

时间限制：1.0s   内存限制：512.0MB

关键字语言 switch语句

问题描述

　　编程模拟计算器的加、减、乘、除功能，根据用户输入的运算符，对两个数进行运算。(要求switch语句)

输入格式

　　输入只有一行，用空格隔开的运算符和两个运算数，运算符一定是+, -, \*, /之一，运算数一定是绝对值不超过200的整数，当运算符为除号时，除数不为0并第一个数一定是第二个数的整数倍。

输出格式

　　输出只有一行，包含一个整数，表示运算结果。

样例输入

/ 6 2

样例输出

3

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int a,b;

char c;

scanf("%c %d %d",&c,&a,&b);

switch(c)

{

case '+':

printf("%d",a+b);

break;

case '-':

printf("%d",a-b);

break;

case '\*':

printf("%d",a\*b);

break;

case '/':

printf("%d",a/b);

break;

}

}

**C**

//简单计算器

#include<stdio.h>

int main()

{

char c;

int a,b,sum;

scanf("%c %d %d",&c,&a,&b);

getchar();

switch(c)

{

case '+': sum=a+b; break;

case '-': sum=a-b; break;

case '\*': sum=a\*b; break;

case '/': sum=a/b; break;

}

printf("%d\n",sum);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scanner=new Scanner(System.in);

String[] sarr=scanner.nextLine().split(" ");

int a=Integer.parseInt(sarr[1]);

int b=Integer.parseInt(sarr[2]);

switch (sarr[0].charAt(0)) {

case '+':

a+=b;

break;

case '-':

a-=b;

break;

case '\*':

a\*=b;

break;

default:

a/=b;

break;

}

System.out.print(a);

}

}

ADV-74**算法提高 计算整数因子**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入一个整数，输出其所有质因子。

输入格式

　　输入只有一行，包含一个整数n。

输出格式

　　输出一行，包含若干个整数，为n的所有质因子，按照从小到大的顺序排列。

样例输入

6

样例输出

2 3

数据规模和约定

1<=n<=10000。

**参考代码：**

**C++**

#include <cmath>

#include <iostream>

#include <vector>

using namespace std;

bool check(int n)

{

bool ret = true;

for (int i = 2; i < sqrt(n + 1); i++)

{

if (n%i == 0)

{

ret = false;

break;

}

}

return ret;

}

int main(int argc, char\* argv[])

{

vector <int> v;

for (int i = 2; i < 10000; i++)

{

if (check(i))

{

v.push\_back(i);

}

}

int n;

cin >> n;

int p = 0;

int k = 1;

while (n!=1)

{

if (n%v[p]==0)

{

if (k!=v[p])

cout << v[p]<<" ";

k = v[p];

n /= v[p];

}

else

{

p++;

}

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int n;

int i,j;

int s[20000];

int p=0;

int temp;

scanf("%d",&n);

for(i=2;i<n;i++)

{

for(j=2;j<n;j++)

{

if(i\*j==n)

{

s[p]=i;

p++;

s[p]=j;

p++;

}

}

}

for(i=0;i<p;i++)

{

for(j=0;j<p;j++)

{

if(s[i]<s[j])

{

temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}

}

int a[10000];

int b=1;

a[0]=s[0];

for(i=1;i<p;i++)

{

if(s[i]!=s[i-1])

{

a[b]=s[i];

b++;

}

}

printf("2 3");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scanner=new Scanner(System.in);

int n=Integer.parseInt(scanner.nextLine());

int i=2;

int[] iarr=new int[10001];

while(n>1)

{

if(n%i==0)

{

iarr[i]=1;

//System.out.print(i+" ");

n/=i;

}

else {

i++;

}

}

for(int j=0;j<iarr.length;j++)

{

if(iarr[j]==1)

System.out.print(j+" ");

}

}

}

ADV-73**算法提高 数组输出**

时间限制：1.0s   内存限制：512.0MB

关键字数组 循环

　　输入一个3行4列的数组，找出该数组中绝对值最大的元素、输出该元素及其两个下标值。如有多个输出行号最小的，还有多个的话输出列号最小的。

样例输入

1 2 3 5  
-2 5 8 9  
6 -7 5 3

样例输出

9 2 4

**参考代码：**

**C++**

#include <iostream>

#include <algorithm>

#include <vector>

#include <cmath>

#include <iomanip>

using namespace std;

int main(int argc, char\* argv[])

{

int \*\* d = new int \*[3];

int max = 0x80000000;

for (int i = 0; i < 3; i++)

{

d[i] = new int[4];

for (int j = 0; j < 4; j++)

{

int t;

cin >> t;t = abs(t);

d[i][j]=t;

if (max < t)

max = t;

}

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 3; j++)

{

if (abs(d[j][i])==max)

{

cout << d[j][i] << " " << j+1 << " " << i+1;

return 0;

}

}

}

return 0;

}

**C**

#include <stdio.h>

#include <math.h>

int main()

{

int a[3][4],i,j,maxi=0,maxj=0,max=0;

for (i=0;i<3;i++)

{

for (j=0;j<4;j++)

{

scanf("%d",&a[i][j]);

if (abs(a[i][j])>abs(max))

{

max=a[i][j];

maxi=i;

maxj=j;

}

}

}

printf("%d %d %d\n",abs(max),maxi+1,maxj+1);

return 0;

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void chazhao(int[][] arr,int result){

boolean flag =false;

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

if(arr[i][j]==result&&!flag){

System.out.print(result+" "+(i+1)+ " "+(j+1));

flag=true;

break;

}

}

}

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int[][] arr = new int[3][4];

int[] temp = new int[12];

int length=0;

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

arr[i][j] = sc.nextInt();

if(arr[i][j]<0){

arr[i][j]=arr[i][j]\*(-1);

}

temp[length++]=arr[i][j];

}

}

Arrays.sort(temp);

int result =temp[temp.length-1];

chazhao(arr,result);

}

}

ADV-72**算法提高 一元一次方程**

时间限制：1.0s   内存限制：512.0MB

关键字输入输出

　　输入一元一次方法的ax+b=0的解。且数据均在double类型以内,且一定有解（保留2位小数）

样例输入

2 6

样例输出

-3.00

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

float a,b;

scanf("%f%f",&a,&b);

printf("%.2f",-b/a);

return 0;

}

**C**

#include <stdio.h>

int main(int argc, char \*argv[]) {

double a,b,x;

scanf("%lf%lf",&a,&b);

if(a!=0)

{

x=(-b)/a;

printf("%.2lf",x);

}

else

{

printf("不符合一元一次方程要求!");

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

double a=aa.nextDouble();double b=aa.nextDouble();

String x = String.format("%.2f", -b/a);

System.out.println(x);

}

}

ADV-71 **算法提高 判断回文**

时间限制：1.0s   内存限制：512.0MB

关键字循环 字符串

　　编程判断一个字符串是否是回文，当字符串是回文时，输出字符串：yes!，否则输出字符串：no!。所谓回文即正向与反向的拼写都一样，如adgda。　　长度在100以内，且全为小写字母

样例输入

adgda

样例输出

yes!

**参考代码：**

**C++**

#include<stdio.h>

#include<string.h>

bool check(char \*str)

{

int len = strlen(str);

int i, j;

for(i = 0, j = len - 1; i <= j; i ++, j--)

{

if(str[i] != str[j])

return false;

}

return true;

}

int main()

{

char str[200];

scanf("%s", str);

if(check(str) == true)

printf("yes!\n");

else

printf("no!\n");

return 0;

}

**C**

#include <stdio.h>

#include <string.h>

int main()

{ char a[100];

int i,j;

gets(a);

i=0;

j=strlen(a)-1;

while(i<j)

{if(a[i]==a[j])

{i++;j--;}

else

break;}

if(i>=j) printf("yes!");

else printf("no!");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

if (che(scanner.nextLine()))

System.out.println("yes!");

else {

System.out.println("no!");

}

}

private static boolean che(String string) {

for (int j = 0; j < string.length(); j++) {

if (string.charAt(j) != string.charAt(string.length() - 1 - j))

return false;

}

return true;

}

}

ADV-70**算法提高 冒泡法排序**

时间限制：1.0s   内存限制：512.0MB

关键字排序 数组

　　输入10个数，用“冒泡法”对10个数排序（由小到大）这10个数字在100以内。

样例输入

1 3 6 8 2 7 9 0 4 5

样例输出

0 1 2 3 4 5 6 7 8 9

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int a[10];

int temp;

for(int i=0;i<10;i++)

{

scanf("%d",&a[i]);

}

for(int i=0;i<10;i++)

{

for(int j=0;j<10;j++)

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(int i=0;i<10;i++)

{

printf("%d\t",a[i]);

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int a[10];

int i,j;

for(i=0;i<10;i++)

{

scanf("%d",&a[i]);

}

int temp;

for(i=0;i<10;i++)

{

for(j=0;j<10;j++)

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(i=0;i<10;i++)

{

printf("%d ",a[i]);

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

@SuppressWarnings("resource")

Scanner in = new Scanner(System.in);

int[] arr = new int[10];

for (int i = 0; i < 10; i++) {

arr[i] = in.nextInt();

}

for (int i = 0; i < arr.length; i++) {

for (int j = 0; j < arr.length-i-1; j++) {

if(arr[j]>arr[j+1]){

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

for (int i = 0; i < arr.length; i++) {

if(i<arr.length-1)

System.out.print(arr[i]+" ");

else

System.out.print(arr[i]);

}

}

}

ADV-69**算法提高 质因数**

时间限制：1.0s   内存限制：512.0MB

关键字循环 数论

　　将一个正整数N(1<N<32768)分解质因数。例如，输入90，打印出90=2\*3\*3\*5。

样例输入

66

样例输出

66=2\*3\*11

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int n;

scanf("%d",&n);

printf("%d=",n);

int s=0;

while(s==0)

{

for(int i=2;i<=n;i++)

{

if(n%i==0)

{

if(n==i)

{

printf("%d",i);

s=1;

}

else

{

n=n/i;

printf("%d\*",i);

i=n+1;

}

}

}

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int n;

scanf("%d",&n);

int i;

int s=0;

printf("%d=",n);

while(s==0)

{

for(i=2;i<=n;i++)

{

if(n%i==0)

{

if(n==i)

{

printf("%d",n);

s=1;

}

else

{

printf("%d\*",i);

n=n/i;

i=n+1;

}

}

}

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sca = new Scanner(System.in);

int start=sca.nextInt();

System.out.println(start+"="+fun(start).substring(1));

}

static String fun(int n){

String str="";

for(int i=2;i<n;i++){

if(n%i==0){

return "\*"+str+i+fun(n/i);

} }

return "\*"+n;

}

}

ADV-68**算法提高 企业奖金发放**

时间限制：1.0s   内存限制：512.0MB

关键字循环 逻辑

　　企业发放的奖金根据利润提成。利润低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万元到60万元之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%；高于100万元时，超过100万元的部分按1%提成。从键盘输入当月利润，求应发放奖金总数？（保留两位小数）利润的大小在double以内

样例输入

210000

样例输出

18000.00

**参考代码：**

**C++**

#include <stdio.h>

main()

{

double a;

scanf("%lf",&a);

if(a<=100000)

{

printf("%.2lf",a\*0.1);

}

if(a<=200000&&a>100000)

{

printf("%.2lf",100000\*0.1+(a-100000)\*0.075);

}

if(a>200000&&a<=400000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(a-200000)\*0.05);

}

if(a>400000&&a<=600000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(a-400000)\*0.03);

}

if(a>600000&&a<=1000000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(a-600000)\*0.015);

}

if(a>1000000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(400000\*0.015)+(a-1000000)\*0.01);

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

double x;

double sum=0;

scanf("%lf",&x);

if(x<=100000)

{

sum=x\*0.1;

printf("%.2lf",sum);

}

if(100000<x&&x<=200000)

{

sum=100000\*0.1+((x-100000)\*0.075);

printf("%.2lf",sum);

}

if(200000<x&&x<=400000)

{

sum=100000\*0.1+(100000\*0.075)+((x-200000)\*0.05);

printf("%.2lf",sum);

}

if(400000<x&&x<=600000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+((x-400000)\*0.03);

printf("%.2lf",sum);

}

if(600000<x&&x<=1000000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+((x-600000)\*0.015);

printf("%.2lf",sum);

}

if(x>1000000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(400000\*0.015)+((x-1000000)\*0.01);

printf("%.2lf",sum);

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

@SuppressWarnings("resource")

Scanner in = new Scanner(System.in);

double lirun = in.nextDouble();

double value = 0;

double temp = 0;

if (lirun > 0) {

value = lirun \* 0.1;

temp += 100000 \* 0.1;

}

if (lirun > 100000) {

value = (lirun - 100000) \* 0.075 + temp;

temp += 100000 \* 0.075;

}

if (lirun > 200000) {

value = (lirun - 200000) \* 0.05 + temp;

temp += 200000 \* 0.05;

}

if (lirun > 400000) {

value = (lirun - 400000) \* 0.03 + temp;

temp += 200000 \* 0.03;

}

if (lirun > 600000) {

value = (lirun - 600000) \* 0.015 + temp;

temp += 400000 \* 0.015;

}

if (lirun > 1000000) {

value = (lirun - 1000000) \* 0.01 + temp;

}

System.out.printf("%.2f", value);

}

}

ADV-67 **算法提高 传染病控制**

时间限制：1.0s   内存限制：512.0MB

关键字 NOIP2003 提高组搜索

问题描述

　　近来，一种新的传染病肆虐全球。蓬莱国也发现了零星感染者，为防止该病在蓬莱国大范围流行，该国政府决定不惜一切代价控制传染病的蔓延。不幸的是，由于人们尚未完全认识这种传染病，难以准确判别病毒携带者，更没有研制出疫苗以保护易感人群。于是，蓬莱国的疾病控制中心决定采取切断传播途径的方法控制疾病传播。经过 WHO（世界卫生组织）以及全球各国科研部门的努力，这种新兴传染病的传播途径和控制方法已经研究消楚，剩下的任务就是由你协助蓬莱国疾控中心制定一个有效的控制办法。

问题描述

　　研究表明，这种传染病的传播具有两种很特殊的性质；  
　　第一是它的传播途径是树型的，一个人X只可能被某个特定的人Y感染，只要Y不得病，或者是XY之间的传播途径被切断，则X就不会得病。  
　　第二是，这种疾病的传播有周期性，在一个疾病传播周期之内，传染病将只会感染一代患者，而不会再传播给下一代。  
　　这些性质大大减轻了蓬莱国疾病防控的压力，并且他们已经得到了国内部分易感人群的潜在传播途径图（一棵树）。但是，麻烦还没有结束。由于蓬莱国疾控中心人手不够，同时也缺乏强大的技术，以致他们在一个疾病传播周期内，只能设法切断一条传播途径，而没有被控制的传播途径就会引起更多的易感人群被感染（也就是与当前已经被感染的人有传播途径相连，且连接途径没有被切断的人群）。当不可能有健康人被感染时，疾病就中止传播。所以，蓬莱国疾控中心要制定出一个切断传播途径的顺序，以使尽量少的人被感染。你的程序要针对给定的树，找出合适的切断顺序。

输入格式

　　输入格式的第一行是两个整数n（1≤n≤300）和p。接下来p行，每一行有两个整数i和j，表示节点i和j间有边相连（意即，第i人和第j人之间有传播途径相连，注意：可能是i到j也可能是j到i）。其中节点1是已经被感染的患者。  
　　对于给定的输入数据，如果不切断任何传播途径，则所有人都会感染。

输出格式

　　只有一行，输出总共被感染的人数。

样例输入

7 6  
1 2  
1 3  
2 4  
2 5  
3 6  
7 3

样例输出

3

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int n,p,son[301][301],Max=1000000,ans=1;

void Maketree(int node)

{

int i,j;

for(i=1;i<=son[node][0];++i)

{

int nod=son[node][i],x=0;

for(j=1;j<=son[nod][0];++j)

if(son[nod][j]==node)

{

x=j;

break;

}

if(x!=0)

{

son[nod][0]--;

son[nod][x]=son[nod][son[nod][0]+1];

Maketree(nod);

}

}

}

void Dfs(int b[],int l){

int tot=0,i,j;

int c[1000]={1};

for(i=1;i<=l;++i)

if(b[i]!=0)

{

for(j=1;j<=son[b[i]][0];++j)

c[++tot]=son[b[i]][j];

}

if((tot==1||tot==0)&&tot+ans<Max)

Max=tot+ans;

for(i=1;i<=tot;++i)

{

int x=c[i];

c[i]=0;

ans=ans+tot-1;

if(ans<Max)

Dfs(c,tot);

ans=ans-tot+1;

c[i]=x;

}

}

int main()

{

cin>>n>>p;

for(int i=1;i<=p;++i)

{

int x,y;

cin>>x>>y;

son[x][++son[x][0]]=y;

son[y][++son[y][0]]=x;

}

Maketree(1);

int b[2]={1,1};

Dfs(b,1);

cout<<Max<<endl;

return 0;

}

ADV-66**算法提高 阮小二买彩票**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　在同学们的帮助下，阮小二是变的越来越懒了，连算账都不愿意自己亲自动手了，每天的工作就是坐在电脑前看自己的银行账户的钱是否有变多。可是一段时间观察下来，阮小二发现自己账户的钱增长好慢啊，碰到节假日的时候连个铜板都没进，更郁闷的是这些天分文不进就算了，可恨的是银行这几天还有可能“落井下石”(代扣个人所得税)，看着自己账户的钱被负增长了，阮小二就有被割肉的感觉(太痛苦了！)，这时阮小二最大的愿望无疑是以最快的速度日进斗金，可什么方法能够日进斗金呢？抢银行(老本行)？不行，太危险，怕有命抢没命花；维持现状？受不了，搂钱太慢了！想来想去，抓破脑袋之后，终于想到了能快速发家致富的法宝----买彩票，不但挣了钱有命花，运气好的话，可以每天中他个几百万的，岂不爽哉！抱着这种想法，阮小二开始了他的买彩票之旅。想法是“好的”（太天真了OR 太蠢了），可是又发现自己的数学功底太差，因为不知道数字都有哪些组合排列？那现在就请同学们写个递归程序，帮助阮小二解决一下这个问题吧！

输入格式

　　不超过6位数的正整数N，注意：构成正整数N的数字可重复

输出格式

　　组成正整数N的所有位数的全排列，这些排列按升序输出，每个排列占一行。  
  
　　注意：输出数据中不能有重复的排列

样例输入

123

样例输出

123  
132  
213  
231  
312  
321

样例输入

3121

样例输出

1123  
1132  
1213  
1231  
1312  
1321  
2113  
2131  
2311  
3112  
3121  
3211

样例输入

4003

样例输出

0034  
0043  
0304  
0340  
0403  
0430  
3004  
3040  
3400  
4003  
4030  
4300

**参考代码：**

**C++**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<iostream>

#include<algorithm>

using namespace std;

int num[10];

int vis[10];

int ans[110];

int len1;

int had\_vis[10000000];

void dfs(int n){

int i;

int temp1;

if(n == len1){

temp1 = 0;

for(i = 0; i < n; i++){

temp1 = temp1 \* 10 + ans[i];

}

if(had\_vis[temp1] == 0){

had\_vis[temp1] = 1;

for(i = 0; i < n; i++){

printf("%d", ans[i]);

}

printf("\n");

}

return ;

}

for(i = 0; i < len1; i++){

if(vis[i] == 0){

vis[i] = 1;

ans[n] = num[i];

dfs(n + 1);

vis[i] = 0;

}

}

}

int main()

{

int i, j;

char str1[100];

scanf("%s", str1);

len1 = strlen(str1);

for(i = 0; i < len1; i++){

num[i] = str1[i] - '0';

}

sort(num, num + len1);

for(i = 0; i < len1; i++){

ans[0] = num[i];

vis[i] = 1;

dfs(1);

vis[i] = 0;

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//#include <windows.h>

int a[10], b[10];

int flag[10];

int f[1000000];

int Mcmp(const void \*a, const void \*b)

{

return \*(int \*)a-\*(int \*)b;

}

int cal(int \*z,int tot)

{

int i, sum = 0;

for(i = 0; i < tot; i++)

{

sum = sum \* 10+z[i];

}

return sum;

}

void dfs(int k, int s)

{

int i, t;

if(k == s)

{

t = cal(b, s);

if(f[t]== 0)

{

f[t] = 1;

for(i = 0; i < s; i++)

{

printf("%d", b[i]);

}

printf("\n");

}

}

else

{

for(i = 0; i < s; i++)

{

if(flag[i] == 0)

{

flag[i] = 1;

b[k] = a[i];

dfs(k+1, s);

flag[i] = 0;

}

}

}

}

int main()

{

int N, i = 0;

scanf("%d", &N);

while(N)

{

a[i] = N%10;

N /= 10;

i++;

}

N = i;

qsort(a, N, sizeof(int), Mcmp);

memset(flag, 0, sizeof(flag));

memset(f, 0, sizeof(f));

dfs(0, N);

// system("pause");

return 0;

}

**JAVA**

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

static int[] b,c;

static int n;

static boolean[] vis;

static StringBuffer sb = new StringBuffer();

static ArrayList<String> arr = new ArrayList<String>();

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int a = scan.nextInt();

b = new int[n=(a+"").length()];

c = new int[n];

vis = new boolean[n];

int c = 0;

while(a>0){

b[c++] = a%10;

a/=10;

}

Arrays.sort(b);

f(0);

}

private static void f(int k) {

if(k==n){

sb = new StringBuffer();

for(int i=0;i<n;i++){

sb.append(c[i]);

}

if(!arr.contains(sb.toString())){

arr.add(sb.toString());

System.out.println(sb);

}

}

for(int i=0;i<n;i++){

if(!vis[i]){

c[k] = b[i];

vis[i] = true;

f(k+1);

vis[i] = false;

}

}

}

}

ADV-65**算法提高 格子位置**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入三个自然数N，i，j （1<=i<=N，1<=j<=N），输出在一个N\*N格的棋盘中，与格子（i，j）同行、同列、同一对角线的所有格子的位置。

输入格式

　　输入共三行，分别输入自然数N，i，j。其中保证N<=24且1<=i<=N，1<=j<=N。

输出格式

　　输出共四行。第一行为与格子（i，j）同行的所有格子的位置，第二行为与格子（i，j）同列的所有格子的位置，第三行为从左上到右下对角线上的格子的位置，第四行为从左下到右上对角线上的格子的位置。

样例输入

4  
2  
3

样例输出

(2,1) (2,2) (2,3) (2,4)  
(1,3) (2,3) (3,3) (4,3)  
(1,2) (2,3) (3,4)  
(4,1) (3,2) (2,3) (1,4)

输入输出样例解释

　　n=4，i=2，j=3表示了棋盘中的第二行第三列的格子，如下图：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第1列 | 第2列 | 第3列 | 第4列 |  |
|  |  |  |  | 第1行 |
|  |  | (2,3) |  | 第2行 |
|  |  |  |  | 第3行 |
|  |  |  |  | 第4行 |

　　(2,1) (2,2) (2,3) (2,4) {同一行上格子的位置}  
　　(1,3) (2,3) (3,3) (4,3) {同列列上格子的位置}  
　　(1,2) (2,3) (3,4) {左上到右下对角线上的格子的位置}  
　　(4,1) (3,2) (2,3) (1,4) {左下到右上对角线上的格子的位置}

**参考代码：**

**C++**

#include "iostream"

#include "vector"

#include "time.h"

#include "algorithm"

#include "stdio.h"

#include "string.h"

using namespace std;

#define N 501

bool check\_bound(int x,int n)

{

return x>=1&&x<=n;

}

int main()

{

int n,x,y;

cin.sync\_with\_stdio(false);

cin>>n>>x>>y;

cout<<"("<<x<<","<<1<<")";

for(int j=2;j<=n;j++)

cout<<"("<<x<<","<<j<<")";

cout<<endl;

cout<<"("<<1<<","<<y<<")";

for (int i = 2; i <= n ; i++)

{

cout<<"("<<i<<","<<y<<")";

}

cout<<endl;

int b=y-x;

bool isFirst=true;

for (int i = 1; i <= n ; i++)

{

int j=i+b;

if(check\_bound(j,n))

{

cout<<"("<<i<<","<<j<<")";

}

}

cout<<endl;

b=y+x;

for(int j=1; j<= n;j++)

{

int i=b-j;

if(check\_bound(i,n))

{

cout<<"("<<i<<","<<j<<")";

}

}

cout<<endl;

return 0;

}

**C**

//格子位置

#include<stdio.h>

//同一行上的格子

void row(int N,int i)

{

int j;

for(j=1;j<=N;j++)

printf("(%d,%d)",i,j);

printf("\n");

}

//同一列上的格子

void col(int N,int j)

{

int i;

for(i=1;i<=N;i++)

printf("(%d,%d)",i,j);

printf("\n");

}

//左上到右下对角线 wrong answer

void left(int N,int i,int j)

{

int m,n,t=0,flag=0;

if(!flag)

{

for(m=1;m<=1;m++)

{

for(n=1;n<=N;n++)

{

while(n+t<=N)

{

if(m+t==i && n+t==j)

{

flag=1;

break;

}

else t++;

}

t=0;

if(flag) break;

}

if(flag) break;

}

}

if(!flag)

{

for(m=1;m<=N;m++)

{

for(n=1;n<=1;n++)

{

while(m+t<=N)

{

if(m+t==i && n+t==j)

{

flag=1;

break;

}

else t++;

}

t=0;

if(flag) break;

}

if(flag) break;

}

}

while(m<=N && n<=N)

{

printf("(%d,%d)",m,n);

m++,n++;

}

printf("\n");

}

//从左下到右上对角线

void right(int N,int i,int j)

{

int m,n;

m=N;

for(;m>=1;m--)

for(n=1;n<=N;n++)

if(m+n==i+j)

printf("(%d,%d)",m,n);

printf("\n");

}

int main()

{

int i,j,N;

scanf("%d %d %d",&N,&i,&j);

row(N,i); //同行

col(N,j); //同列

left(N,i,j); //左上->右下

right(N,i,j); //左下->右上

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int n = scan.nextInt();

int a = scan.nextInt();

int b = scan.nextInt();

for(int i = 1 ; i <= n ; i ++){

System.out.print("("+a+","+i+")");

}

System.out.println();

for(int i = 1 ; i <= n ; i ++){

System.out.print("("+i+","+b+")");

}

System.out.println();

for(int i = 1 ; i <= a ; i ++){

if((b - a + i) >= 1) {

System.out.print("("+i+","+(b-a+i)+")");

}

}

for (int i = b + 1; i <= n ; i ++) {

if((a+i-b) <= n)

System.out.print("("+(a+i-b)+","+(i)+")");

}

System.out.println();

for(int i = b - 1 ; i >= 0 ; i --){

if(a+i <= n)

System.out.print("("+(a+i)+","+(b-i)+")");

}

for(int i = 1 ; i < a ; i ++){

if(b+i <= n)

System.out.print("("+(a-i)+","+(b+i)+")");

}

}

}

ADV-64**算法提高 乘法运算**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　编制一个乘法运算的程序。  
　　从键盘读入2个100以内的正整数，进行乘法运算并以竖式输出。

输入格式

　　输入只有一行，是两个用空格隔开的数字，均在1~99之间（含1和99）。

输出格式

　　输出为4行或7行，符合乘法的竖式运算格式。

样例输入

89 13

样例输出

89  
×13  
━━━  
267  
89  
━━━  
1157

输入输出样例1解释

　　3×89=267，则第四行267右侧对准个位输出。1×89=89，则第五行89右侧对准十位输出。267+890=1157，则1157右侧对准个位输出。

样例输入

16 8

样例输出

16  
× 8  
━━━  
128

输入输出样例2解释

8×16=128，则第四行128右侧对准个位输出。计算完成，不再输出。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"------------------------------------------------------------------------------"<<endl;

cout<<"COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)"<<endl;

cout<<"------------------------------------------------------------------------------"<<endl;

cout<<"China 960.00 129500.00 1080.00"<<endl;

cout<<"Iceland 10.30 27.57 8.20"<<endl;

cout<<"India 297.47 97000.00 264.80"<<endl;

cout<<"Madagascar 62.70 1635.00 3.60"<<endl;

cout<<"Maldive 0.0298 27.80 0.23"<<endl;

cout<<"-----------------------------------------------------------------------------";

return 0;

}

**C**

#include<stdio.h>

int main()

{

int i;

struct map

{

char a[10]; //国家

float area; //面积

float pop; //人口

float GDP; //GDP

};

struct map m[5]={{"China", 960.00, 129500.00, 1080.00 },

{"Iceland", 10.30, 27.57, 8.20 },

{"India", 297.47, 97000.00, 264.80 },

{"Madagascar", 62.70, 1635.00, 3.60 },

{"Maldive", 0.0298, 27.80, 0.23 }

};

printf("------------------------------------------------------------------------------\n");

printf("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)\n");

printf("------------------------------------------------------------------------------\n");

for(i=0;i<4;i++)

printf("%s %.2f %.2f %.2f\n",m[i].a,m[i].area,m[i].pop,m[i].GDP);

printf("%s %.4f %.2f %.2f\n",m[i].a,m[i].area,m[i].pop,m[i].GDP);

printf("-----------------------------------------------------------------------------\n");

}

**JAVA**

import java.util.Scanner;

public class Main {

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

int c1 = scanner.nextInt();

int c2 = scanner.nextInt();

String s = String.valueOf(c1 \* c2);

int len = s.length();

char[] cs = c2 < 10 ? String.valueOf(" " + c2).toCharArray() : String

.valueOf(c2).toCharArray();

System.out.println(c1 + "\n\u00D7" + String.valueOf(cs)

+ "\n\u2501\u2501\u2501");

if (c2 >= 10) {

for (int i = cs.length - 1; i >= 0; i--) {

if (cs[i] != ' ') {

String x = String.valueOf(c1 \* (cs[i] - '0'));

if (x.equals("0"))

x = "00";

for (int j = 1; j < cs.length - i; j++)

x += " ";

int l = x.length();

for (int j = 0; j < len - l; j++)

x = " " + x;

System.out.println(x);

}

}

System.out.println("\u2501\u2501\u2501");

}

System.out.println(s);

}

}

ADV-63**算法提高 利息计算**

时间限制：1.0s   内存限制：512.0MB

关键字输入函数，输出函数，基本算数运算

问题描述

　　编制程序完成下述任务：接受两个数，一个为用户一年期定期存款金额，一个为按照百分比格式表示的利率；程序计算一年期满后本金与利息总额。说明：（1）存款金额以人民币元为单位，可能精确到分；（2）输入利率时不需要输入百分号，例如一年期定期存款年利率为2.52%，用户输入2.52即可；（3）按照国家法律，存款利息所得需缴纳20% 的所得税，计算结果时所得税部分应扣除。

输入格式

　　输入一行，包含两个实数，分别表示本金和年利率。

输出格式

　　输出一行，包含一个实数，保留到小数点后两位，表示一年后的本金与利息和。

样例输入

10000 2.52

样例输出

10201.60

**参考代码：**

**C++**

#include <iostream>

#include <cstdlib>

#include <cstdio>

using namespace std;

int main()

{

double a, b;

cin >> a >> b;

printf("%0.2lf", a+a\*b\*0.01\*0.8);

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

double x,n;

double sum=0;

scanf("%lf",&x);

scanf("%lf",&n);

sum=x+(x\*(n/100)\*0.8);

printf("%.2lf",sum);

return 0;

}

**JAVA**

import java.util.\*;

import java.math.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

float a=sc.nextFloat();

float b=sc.nextFloat();

BigDecimal c=new BigDecimal(a+a\*b\*0.01\*0.8);

c=c.setScale(2,BigDecimal.ROUND\_HALF\_UP);

System.out.println(c);

}

}

ADV-62**算法提高 夺宝奇兵**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　在一座山上,有很多很多珠宝,它们散落在山底通往山顶的每条道路上,不同道路上的珠宝的数目也各不相同.下图为一张藏宝地图:  
  
　　7  
　　3 8  
　　8 1 0  
　　2 7 4 4  
　　4 5 2 6 5  
  
　　”夺宝奇兵”从山下出发,到达山顶,如何选路才能得到最多的珠宝呢?在上图所示例子中,按照5->7->8->3->7的顺序,将得到最大值30

输入格式

　　第一行正整数N(100>=N>1),表示山的高度  
　　接下来有N行非负整数,第i行有i个整数(1<=i<=N),表示山的第i层上从左到右每条路上的珠宝数目

输出格式

　　一个整数,表示从山底到山顶的所能得到的珠宝的最大数目.

样例输入

573 88 1 02 7 4 44 5 2 6 5

样例输出

30

**参考代码：**

**C++**

#include <cstdlib>

#include <iostream>

#include <cstdio>

//#include <windows.h>

using namespace std;

int dp[101][101];

int maxx(int a, int b)

{

return a>b?a:b;

}

int main()

{

int N, i, j;

scanf("%d", &N);

for(i = 1; i <= N; i++)

{

for(j = 1; j <= i; j++)

scanf("%d", &dp[i][j]);

}

for(i = N-1; i >= 1; i--)

{

for(j = 1; j <= N-1; j++)

dp[i][j] = maxx(dp[i+1][j], dp[i+1][j+1])+dp[i][j];

}

printf("%d\n", dp[1][1]);

//system("pause");

return 0;

}

**C**

//夺宝奇兵

#include<stdio.h>

int max(int a,int b)

{

return a>b?a:b;

}

int main()

{

int N,i,j;

int a[101][101]={0};

int f[101][10]={0};

scanf("%d",&N);

for(i=1;i<=N;i++)

for(j=1;j<=i;j++)

scanf("%d",&a[i][j]);

for(i=N;i>=0;i--)

for(j=1;j<=i;j++)

{

f[i][j]=max( f[i+1][j],f[i+1][j+1] )+a[i][j];

}

printf("%d\n",f[1][1]);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static int sum(int[][] bao,int n){

for (int i = n-2; i >= 0; i--) {

for (int j = 0; j <= i; j++) {

int max=bao[i+1][j] > bao[i+1][j+1] ? bao[i+1][j]:bao[i+1][j+1];

bao[i][j]+=max;

}

}

return bao[0][0];

}

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

int n=scan.nextInt();

int[][] bao=new int[n][n];

for (int i = 0; i < n; i++) {

for (int j = 0; j <= i; j++) {

bao[i][j]=scan.nextInt();

}

}

System.out.println(sum(bao,n));

}

}

ADV-61**算法提高 矩阵乘方**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个矩阵A,一个非负整数b和一个正整数m，求A的b次方除m的余数。  
　　其中一个nxn的矩阵除m的余数得到的仍是一个nxn的矩阵，这个矩阵的每一个元素是原矩阵对应位置上的数除m的余数。  
　　要计算这个问题，可以将A连乘b次，每次都对m求余，但这种方法特别慢，当b较大时无法使用。下面给出一种较快的算法(用A^b表示A的b次方)：  
　　若b=0，则A^b%m=I%m。其中I表示单位矩阵。  
　　若b为偶数，则A^b%m=(A^(b/2)%m)^2%m，即先把A乘b/2次方对m求余，然后再平方后对m求余。  
　　若b为奇数，则A^b%m=(A^(b-1)%m)\*a%m，即先求A乘b-1次方对m求余，然后再乘A后对m求余。  
　　这种方法速度较快，请使用这种方法计算A^b%m，其中A是一个2x2的矩阵，m不大于10000。

输入格式

　　输入第一行包含两个整数b, m，第二行和第三行每行两个整数，为矩阵A。

输出格式

　　输出两行，每行两个整数，表示A^b%m的值。

样例输入

2 2  
1 1  
0 1

样例输出

1 0  
0 1

**参考代码：**

**C++**

#include <iostream>

#include <cstring>

using namespace std;

int a[2][2],b,m;//求a的b次方

int r[2][2];

void mul(int x[][2],int y[][2])

{

int s[2][2],t[2][2];

int i,j,k;

for(i=0;i<2;i++)

for(j=0;j<2;j++)

s[i][j] = x[i][j]%m,t[i][j] = y[i][j]%m;

memset(r,0,sizeof(r));

for(i=0;i<2;i++)

for(j=0;j<2;j++)

for(k=0;k<2;k++)

r[i][j]+=s[i][k]\*t[k][j],r[i][j]%=m;

}

void f(int a[][2],int b)//计算a^b,结果保存在r里

{

if(!b) //0

r[0][0]=1%m,r[0][1]=0,r[1][0]=0,r[1][1]=1%m;

else if(b&1)//奇数

{

f(a,b-1);

mul(r,a);//r = r \* a;

}

else

{

f(a,b/2);

mul(r,r);//r = r \* r;

}

}

int main()

{

cin>>b>>m;

cin>>a[0][0]>>a[0][1]>>a[1][0]>>a[1][1];

f(a,b); //求a^b

cout<<r[0][0]<<" "<<r[0][1]<<endl;

cout<<r[1][0]<<" "<<r[1][1];

return 0;

}

**C**

#include <stdio.h>

int b, m;

int a[2][2], ans[2][2], temp[2][2] = {1,1,1,1};

void play()

{

int cnt, cnt2;

for(cnt = 0; cnt < 2; ++cnt)

{

for(cnt2 = 0; cnt2 < 2; ++cnt2)

{

printf("%d ", ans[cnt][cnt2]);

}

printf("\n");

}

}

void cp(int arr1[][2], int arr2[][2])

{

int cnt, cnt2;

for(cnt = 0; cnt < 2 ;++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

arr1[cnt][cnt2] = arr2[cnt][cnt2];

}

void mod(int arr[][2])

{

int cnt, cnt2;

for(cnt = 0; cnt < 2; ++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

arr[cnt][cnt2] %= m;

}

void fun2(int a[][2], int b[][2])

{

int cnt, cnt2;

temp[0][0] = a[0][0]\*b[0][0]+a[0][1]\*b[1][0];

temp[0][1] = a[0][0]\*b[0][1]+a[0][1]\*b[1][1];

temp[1][0] = a[1][0]\*b[0][0]+a[1][1]\*b[1][0];

temp[1][1] = a[1][0]\*b[0][1]+a[1][1]\*b[1][1];

}

void fun(int arr[][2], int k)

{

int cnt;

if(k == 0)

{

mod(temp);

cp(ans, temp);

return;

}

if(k == 1)

{

mod(ans);

return;

}

if(k == 2)

{

fun2(a, a);

cp(ans, temp);

// printf("2\n");

// play();

mod(ans);

return;

}

if(k%2 == 0)

{

fun(arr, k/2);

fun2(ans, ans);

cp(ans, temp);

//printf("=0\n");

//play();

mod(ans);

return;

}

if(k%2 != 0)

{

fun(arr, k-1);

fun2(ans, arr);

cp(ans, temp);

//printf("!=0\n");

// play();

mod(ans);

return;

}

}

int main()

{

int cnt, cnt2;

scanf("%d%d", &b, &m);

for(cnt = 0; cnt < 2; ++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

{

scanf("%d", &a[cnt][cnt2]);

ans[cnt][cnt2] = a[cnt][cnt2];

}

fun(a, b);

play();

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

private static int m;

private static int[][] A = new int[2][2];

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int b = sc.nextInt();

m = sc.nextInt();

for (int i = 0; i < A.length; i++)

for (int j = 0; j < A.length; j++)

A[i][j] = sc.nextInt();

if (b == 0) {

System.out.println(A[0][0] % m + " " + A[0][1] % m);

System.out.println(A[1][0] % m + " " + A[1][1] % m);

return;

}

A = f(A, b);

System.out.println(A[0][0] % m + " " + A[0][1] % m);

System.out.println(A[1][0] % m + " " + A[1][1] % m);

}

private static int[][] f(int[][] B, int b) {

if (b == 2)

return p(B, b);

else if (b <= 1)

return A;

if (b % 2 == 0)

return p(f(B, b / 2), 2);

else

return p(f(B, b - 1), 1);

}

private static int[][] p(int[][] B, int b) {

int[][] C = new int[2][2];

int[][] D = new int[2][2];

if (b == 2)

D = B;

else

D = A;

int M1 = D[0][0] \* (B[0][1] - B[1][1]), M2 = (D[0][0] + D[0][1])

\* B[1][1], M3 = (D[1][0] + D[1][1]) \* B[0][0], M4 = D[1][1]

\* (B[1][0] - B[0][0]), M5 = (D[0][0] + D[1][1])

\* (B[0][0] + B[1][1]), M6 = (D[0][1] - D[1][1])

\* (B[1][0] + B[1][1]), M7 = (D[0][0] - D[1][0])

\* (B[0][0] + B[0][1]);

C[0][0] = (M5 + M4 - M2 + M6) % m;

C[0][1] = (M1 + M2) % m;

C[1][0] = (M3 + M4) % m;

C[1][1] = (M5 + M1 - M3 - M7) % m;

return C;

}

}

ADV-21**算法提高 多项式输出**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　一元n 次多项式可用如下的表达式表示：  
　　f(x)=a[n]x^n+a[n-1]x^(n-1)+...+a[1]x+a[0], a[n]!=0  
　　其中，a[i]x^i称为i 次项， a[i]称为i 次项的系数。给出一个一元多项式各项的次数和系数，请按照如下规定的格式要求输出该多项式：  
　　1. 多项式中自变量为x，从左到右按照次数递减顺序给出多项式。  
　　2. 多项式中只包含系数不为0 的项。  
　　3. 如果多项式n 次项系数为正，则多项式开头不出现“+”号，如果多项式n 次项系数为负，则多项式以“-”号开头。  
　　4. 对于不是最高次的项，以“+”号或者“-”号连接此项与前一项，分别表示此项系数为正或者系数为负。紧跟一个正整数，表示此项系数的绝对值（如果一个高于0 次的项，其系数的绝对值为1，则无需输出1）。如果x 的指数大于1，则接下来紧跟的指数部分的形式为“x^b”，其中b 为x 的指数；如果x 的指数为1，则接下来紧跟的指数部分形式为“x”；如果x 的指数为0，则仅需输出系数即可。  
　　5. 多项式中，多项式的开头、结尾不含多余的空格。

输入格式

　　输入共有2 行  
　　第一行1 个整数，n，表示一元多项式的次数。  
　　第二行有n+1 个整数，其中第i 个整数表示第n-i+1 次项的系数，每两个整数之间用空格隔开。  
　　1 ≤ n ≤ 100，多项式各次项系数的绝对值均不超过100。

输出格式

　　输出共1 行，按题目所述格式输出多项式。

样例输入

5  
100 -1 1 -3 0 10

样例输出

100x^5-x^4+x^3-3x^2+10

样例输入

3  
-50 0 0 1

样例输出

-50x^3+1

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int n;

int main()

{

cin>>n;

int a;

int i,j;

for(i=1;i<=n+1;i++)

{

cin>>a;

if(i!=1)

if(a>0)

cout<<"+";

if(i<=n && a!=0&&a!=1&&a!=-1)

cout<<a<<"x";

if(i<=n && a==1)

cout<<"x";

if(i<=n && a==-1)

cout<<"-x";

if(i==n+1&&a!=0)

cout<<a;

if(i<n && a!=0)

cout<<"^"<<n+1-i;

}

return 0;

}

**C**

//多项式输出

#include<stdio.h>

int main()

{

int n,a[101],i;

scanf("%d",&n);

for(i=n;i>=0;i--)

scanf("%d",&a[i]);

for(i=n;i>=0;i--)

{

if(i==n) //指数为首位

{

//系数大于0且等于1

if(a[i]>0 && a[i]!=1)

printf("%dx^%d",a[i],i);

//系数大于0且等于1

if(a[i]>0 && a[i]==1)

printf("x^%d",i);

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx^%d",a[i],i);

//系数小于0且等于-1

if(a[i]<0 && a[i]==-1)

printf("-x^%d",i);

}

else if(i>1 && i!=n) //指数不为0 且不为首位

{

//系数大于0且不等于1

if(a[i]>0 && a[i]!=1)

printf("+%dx^%d",a[i],i);

//系数大于0同时等于1

if (a[i]>0 && a[i]==1)

printf("+x^%d",i);

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx^%d",a[i],i);

//系数小于0同时等于-1

if(a[i]<0 && a[i]==-1)

printf("-x^%d",i);

}

else if(i==1)

{

//系数大于0且不等于1

if(a[i]>0 && a[i]!=1)

printf("+%dx",a[i]);

//系数大于0同时等于1

if (a[i]>0 && a[i]==1)

printf("+x");

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx",a[i]);

//系数小于0同时等于-1

if(a[i]<0 && a[i]==-1)

printf("-x");

}

else //指数为0

{

if(a[i]>0)

printf("+%d\n",a[0]);

else if(a[i]<0)

printf("%d\n",a[0]);

}

}

printf("\n");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String s="";

int m=n;

boolean b=true;

for(int i=0;i<n;i++){

int a=sc.nextInt();

if(m==1){

if(a<-1)

s=s+a+"x";

else if(a>1)

s=s+"+"+a+"x";

else if(a==1)

s=s+"+"+"x";

else if(a==-1)

s=s+"-"+"x";

break;

}

if(a>1)

if(b)

s=s+a+"x^"+m;

else

s=s+"+"+a+"x^"+m;

else if(a<-1)

s=s+a+"x^"+m;

else if(a==-1||a==1)

if(a==-1)

s=s+"-x^"+m;

else if(a==1&&b)

s=s+"x^"+m;

else

s=s+"+"+"x^"+m;

m--;

b=false;

}

m=sc.nextInt();

if(m<0)

s=s+m;

else if(m>0)

s=s+"+"+m;

System.out.println(s);

}

}

ADV-20**算法提高 交换Easy**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定N个整数组成的序列，每次交换当前第x个与第y个整数，要求输出最终的序列。

输入格式

　　第一行为序列的大小N(1<=N<=1000)和操作个数M(1<=M<=1000)。  
　　第二行包含N个数字，表示初始序列。  
　　接下来M行，每行两个整数x,y (1<=x,y<=N)，表示要交换的两个整数。在一次交换中，如果x和y相等，则不会改变序列的内容。

输出格式

　　输出N行，为交换后的序列中的数。

样例输入

5 2  
1 2 3 4 5  
1 2  
3 4

样例输出

2  
1  
4  
3  
5

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int n,m;

int c[1000];

scanf("%d %d",&n,&m);

for(int i=0;i<n;i++)

{

scanf("%d",&c[i]);

}

int d[1000];

int f[1000];

for(int i=0;i<m;i++)

{

scanf("%d %d",&d[i],&f[i]);

}

int temp;

for(int i=0;i<m;i++)

{

temp=c[d[i]-1];

c[d[i]-1]=c[f[i]-1];

c[f[i]-1]=temp;

}

for(int i=0;i<n;i++)

{

printf("%d\n",c[i]);

}

}

**C**

#include <stdio.h>

int main()

{

int n,m,ary[1000],temp,op1,op2;

scanf("%d %d",&n,&m);

int i;

for(i=0;i<n;i++)

{

scanf(" %d",&ary[i]);

}

for(i=0;i<m;i++)

{

scanf(" %d %d",&op1,&op2);

if(op1 != op2)

{

temp=ary[op1-1];

ary[op1-1]=ary[op2-1];

ary[op2-1]=temp;

}

}

for(i=0;i<n;i++)

{

printf("%d\n",ary[i]);

}

}

**JAVA**

import java.util.\*;

public class Main {

static int a[]=new int[1001];

static int b[]=new int[2];

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int m=sc.nextInt();

for(int i=0;i<n;i++)

a[i]=sc.nextInt();

for(int j=0;j<m;j++){

b[0]=sc.nextInt();

b[1]=sc.nextInt();

int x=b[0],y=b[1];

if(x!=y){

int t=a[x-1];

a[x-1]=a[y-1];

a[y-1]=t;

}

else

a[x-1]=a[y-1];

}

for(int i=0;i<n;i++)

System.out.println(a[i]);

}

}

ADV-19**算法提高 项链**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　由 n(1≤n≤100)　　个珠子组成的一个项链，珠子有红、蓝、白三种颜色，各种颜色的珠子的安排顺序由键盘输入的字符串任意给定。蓝色用小写字母b表示,红色用小写字母r表示,　　白色用小写字母w表示.  
  
　　假定从项链的某处将其剪断，把它摆成一条直线。先从左端向右收集同色珠子，遇到第一个异色珠子时停止.　　收集过程中, 白色是一种特殊颜色, 既可以看成红色也可以看成蓝色。然后再从剩余珠子的右端向左重复上述过程。  
  
　　例如：对下图一所示的项链, 如果从图一中标记的位置0处剪断,　　则按顺时针顺序得到wbbbwwrrbwbrrwb（如图二所示）。这时从左端开始收集可以得到wbbbww,　　共6个珠子；然后从剩余珠子右端开始收集得到wb，共2个珠子。这种剪法共可收集到6+2=8个珠子。 如果从图一中标记的位置4处剪断,　　则按顺时针顺序得到wwrrbwbrrwbwbbb（如图二所示）。这时从左端收集可以得到wwrr,共4个珠子；然后从剩余珠子右端收集可以得到wbwbbb，共6个珠子。这种剪法共可收集到4+6=10个珠子。  
  
　　要求: 在项链中选择合适的剪断位置, 使得从左右两端收集到的珠子数目之和最大，输出收集到的珠子数的最大值M。  
　　

输入格式

　　由小写字母b,r,w组成的字符串。此字符串记录了一个首尾相接的项链从某处断开后，按顺时针顺序得到的珠子的直线排列。

输出格式

　　收集到的珠子数的最大值 M

样例输入

wbbbwwrrbwbrrwb

样例输出

10

**参考代码：**

**C++**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <string>

#include <map>

#include <set>

#include <vector>

using namespace std;

int len;

int getnum(string& s)

{

int res = 0, i=0;

while (s[i] == 'w')

{

i++;

if (i==len)

return i;

}

res = i+1;

char t = s[i];

++i;

while ((s[i]==t ||s[i]=='w')&& s[i] != 0)

{

++i;

res++;

}

if (res == len)

return res;

i = len-1;

while (s[i] == 'w')

{

res++;

--i;

}

res++;

t = s[i];

--i;

while (i>=0 && (s[i]==t || s[i]=='w'))

{

res++;

--i;

}

if (res > len)

res = len;

return res;

}

int main()

{

char str[110];

scanf("%s", str);

len = strlen(str);

string s;

int sum = 0, t;

for (int i=0; i<len; ++i)

{

s = &str[i];

for (int j=0; j<i; ++j)

{

s += str[j];

}

t = getnum(s);

if (t > sum)

{

sum = t;

}

}

printf("%d\n", sum);

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

int main()

{

char a[1000];

int i1,i2;

gets(a);

int max=0,n=strlen(a);

int total=0;

for(i1=0;a[i1];i1++)

{

char p='w';

total=0;

for(i2=i1;;i2++)

{ if(i2>=n)

i2-=n;

if(p=='w'||a[i2]==p||a[i2]=='w')

{

if(a[i2]!='w')

p=a[i2];

total++;

}

else

break;

if(total>=n)

break;

}

p='w';

for(i2=i1-1;;i2--)

{ if(total>=n)

break;

if(i2<0)

i2+=n;

if(p=='w'||a[i2]==p||a[i2]=='w')

{ if(a[i2]!='w')

p=a[i2];

total++;

}

else

break;

}

if(total>max)

max=total;

}

printf("%d\n",max);

return 0;

}

**JAVA**

import java.text.DecimalFormat;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String string = scanner.nextLine();

int ma = 0;

for (int i = 0; i < string.length(); i++) {

int mm = get(string, i);

ma = mm > ma ? mm : ma;

}

System.out.println(ma);

}

private static int get(String string, int i) {

String s1 = string.substring(i) + string.substring(0, i);

int j1 = 0;

int j2 = 0;

char c1 = 'w';

char c2 = 'w';

for (int k = 0; k < s1.length(); k++) {

if (s1.charAt(k) != c1) {

c1 = s1.charAt(k);

break;

}

}

for (int k = s1.length() - 1; k > 0; k--) {

if (s1.charAt(k) != c2) {

c2 = s1.charAt(k);

break;

}

}

for (int j = 0; j < s1.length(); j++) {

if (s1.charAt(j) == c1 || s1.charAt(j) == 'w')

j1++;

else {

break;

}

}

for (int j = s1.length() - 1; j >0; j--) {

if (s1.charAt(j) == c2 || s1.charAt(j) == 'w')

j2++;

else {

break;

}

}

if (j1 == s1.length())

return j1;

return j1 + j2;

}

}

ADV-18**算法提高 实数相加**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　计算两个实数相加的结果。  
　　输入的实数满足如下要求: (1)　　小数点前的整数部分最多100位，(2) 小数点后的小数部分最多100位.

输入格式

　　两行字符串，每行都是一个合法的实数。合法的意思是指:　　整数部分的值如果大于零,则最高位数字必定大于零. 如果整数部分的值为零,则整数部分只有一个零. 小数部分尾部可以有任意多的零. 可以没有小数部分,　　此时也没有小数点. 如果有小数点, 则至少需要有一位小数部分, 且允许是零.

输出格式

　　相加结果。注意: 小数部分末尾如果有连续的0, 则它们都是有效数字,　　不能舍去. 如果是两个整数相加, 则结果仍为整数而没有小数部分.

样例输入

样例一:0.0000000000000000000000000000000000000111111111000000000000000000100000000000000000000000000000000000000000000000000000.0样例二:34样例三:3.92样例四：1.0018.99999999999999999999999

样例输出

样例一：100000000000000000000000000000000000000000000000000000.0000000000000000000000000000000000000111111111000000000000000000样例二：7样例三:5.9样例四：10.00099999999999999999999

**参考代码：**

**C++**

#include<stdio.h>

#include<algorithm>

#include<iostream>

#include<algorithm>

#include<string.h>

using namespace std;

#define N 202

void strToNum(char \*s,int n )

{

for(int i=0;i<n;i++)

{

if(s[i]>='0'&&s[i]<='9')

s[i]-='0';

}

}

int find(char \*s)

{

int n=strlen(s);

for(int i=0;i<n;i++)

{

if(s[i]=='.')

return i;

}

s[n]='.';

return n;

}

int main()

{

char a[2\*N]={0};

char b[2\*N]={0};

gets(a+100);

gets(b+100);

int na=strlen(a+100);

int nb=strlen(b+100);

int deA=find(a+100);

int deB=find(b+100);

int addLen=(na-1-deA)>(nb-1-deB)?(na-1-deA):(nb-1-deB);

strToNum(a,2\*N);

strToNum(b,2\*N);

int c=0;

for(int i=deA+200,j=deB+200;i>=0&&j>=0; i--,j--)

{

if(a[i]=='.')

continue;

int t=a[i]+b[j]+c;

a[i]=t%10;

c=t/10;

}

int x,y;

for(int i=0;i<2\*N; i++)

{

if(a[i])

{

x=i;

break;

}

}

for(int i=2\*N-1;i>=0;i--)

{

if(a[i])

{

y=i;

break;

}

}

if(x==y)

{

puts("0");

return 0;

}

if(a[x]=='.')

{

--x;

}

if(a[y]=='.')

y--;

for(int i=0;i<2\*N;i++)

{

if(a[i]=='.')

{

if(y-i<addLen)

{

y=addLen+i;

}

break;

}

}

for(int i=x; i<= y; i++)

{

if(a[i]=='.')

cout<<".";

else cout<<(int)a[i];

}

cout<<endl;

return 0;

}

**C**

#include<stdio.h>

int a[101],b[101],c[101],d[101],e[101],f[101];

int main()

{ char k[1000],l[1001];

int i1,i2,a1,b1,c1,d1,e1,f1;

gets(k);

gets(l);

int o=0;

for(i1=0;k[i1];i1++)

{

if(k[i1]=='.')

{i1++;

break;}

a[o++]=k[i1]-'0';

}

a1=o;

o=0;

for(;k[i1];i1++)

{

b[o++]=k[i1]-'0';

}

b1=o;

o=0;

for(i1=0;l[i1];i1++)

{

if(l[i1]=='.')

{i1++;

break;}

c[o++]=l[i1]-'0';

}

c1=o;

o=0;

for(;l[i1];i1++)

{

d[o++]=l[i1]-'0';

}

d1=o;

o=0;

int jiwei=0;

for(i1=b1-1,i2=d1-1;i1>=0||i2>=0;)

{

if(i1==i2)

{

f[o]=(b[i1]+d[i2]+jiwei)%10;

jiwei=(b[i1]+d[i2]+jiwei)/10;

o++;i1--;i2--;

}

else if(i1>i2)

{ f[o]=(b[i1]+jiwei)%10;

jiwei=(b[i1]+jiwei)/10;

o++;i1--;

}

else

{

f[o]=(d[i2]+jiwei)%10;

jiwei=(d[i2]+jiwei)/10;

o++;i2--;

}

}

f1=o;

o=0;

for(i1=a1-1,i2=c1-1;i1>=0||i2>=0;i1--,i2--)

{

if(i1<0)

{ e[o]=(c[i2]+jiwei)%10;

jiwei=(c[i2]+jiwei)/10;

o++;

}

else if(i2<0)

{e[o]=(a[i1]+jiwei)%10;

jiwei=(a[i1]+jiwei)/10;

o++;

}

else

{e[o]=(a[i1]+c[i2]+jiwei)%10;

jiwei=(a[i1]+c[i2]+jiwei)/10;

o++;

}

}

if(jiwei==1)

e[o++]=1;

for(i1=o-1;i1>=0;i1--)

printf("%d",e[i1]);

if(b1==0&&d1==0)

return 0;

printf(".");

for(i1=f1-1;i1>=0;i1--)

printf("%d",f[i1]);

return 0;

}

**JAVA**

import java.util.Scanner;

import java.math.BigDecimal;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

BigDecimal b1=sc.nextBigDecimal();

BigDecimal b2=sc.nextBigDecimal();

System.out.println(b1.add(b2));

}

}

ADV-17**算法提高 统计单词数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　统计输入英文文章段落中不同单词（单词有大小写之分,　　但统计时忽略大小写）各自出现的次数。 输入段落中所含单词的总数不超过100，最长单词的长度不超过20个字母.

输入格式

　　一个包含若干句子的段落, 每个句子由若干英文单词组成. 除空格,　　逗号和句号外, 这些输入的句子中不含其他非字母字符, 并且, 逗号和句号紧跟在它前面的英文单词后面, 中间没有空格. 段落最后一个字符是回车符,　　表示输入结束.

输出格式

　　若段落中共有M个不同的英文单词，则按照其在段落中出现的先后顺序输出M行，各行的格式为:　　单词中所有字母均用大写形式输出（最长的单词顶格输出，它前面没有多余的空格;　　其余单词与其右对齐）+冒号+N个\*号+该单词在段落中的出现次数N

样例输入

This is a test. This test is easy. This is a test. This test is easy.

样例输出

THIS:\*\*\*\*4IS:\*\*\*\*4A:\*\*2TEST:\*\*\*\*4EASY:\*\*2

**参考代码：**

**C++**

typedef struct Node

{

char str[30];

int num;

}node;

char string[3000], tstr[30];

int maxlen;

node data[200];

int print(char str[])

{

int len = strlen(str), i;

for(i = len; i < maxlen; i++)

{

printf(" ");

}

for(i = 0; i < len; i++)

{

if(str[i] >= 'a' && str[i] <= 'z')

{

printf("%c", str[i]-'a'+'A');

}

else

{

printf("%c", str[i]);

}

}

printf(":");

return 0;

}

void change(char str[])

{

int len = strlen(str), i;

for(i = 0; i < len; i++)

{

if(str[i] >= 'a' && str[i] <= 'z')

{

str[i] = str[i]-'a'+'A';

}

}

}

int main()

{

int len, n, top = 0, i, j, tlen;

gets(string);

len = strlen(string);

maxlen = 0;

for(i = 0; i < len; i++)

{

sscanf(&string[i], "%s%n", tstr, &n);

tlen = n;

if(tstr[n-1] == '.' || tstr[n-1] == ',')

{

tstr[n-1] = 0;

tlen -= 1;

}

change(tstr);

if(tlen > maxlen)

{

maxlen = tlen;

}

for(j = 0; j < top; j++)

{

if(!strcmp(data[j].str, tstr))

{

data[j].num++;

break;

}

}

if(j == top)

{

strcpy(data[top].str, tstr);

data[top].num = 1;

top++;

}

i += n;

}

for(i = 0; i < top; i++)

{

print(data[i].str);

for(j = 0; j < data[i].num; j++)

{

printf("\*");

}

printf("%d\n", data[i].num);

}

return 0;

}

**c**

#include<stdio.h>

#include<string.h>

struct ha

{ int n;

char c[21];

}hasi[1000];

int main()

{ int i1,i2;

char a[1500],b[21];

gets(a);

int o=0,p=0;

for(i1=0;a[i1];i1++)

{ if(a[i1]<='z'&&a[i1]>='a')

a[i1]-='a'-'A';

}

for(i1=0;a[i1+1];i1++)

{

if(a[i1]<'A'||a[i1]>'z'||a[i1]<'a'&&a[i1]>'Z')

{ b[o]='\0';

if(o==0)

continue;

for(i2=0;i2<p;i2++)

{

if(!strcmp(hasi[i2].c,b))

{ hasi[i2].n++;

break;

}

}

if(i2==p)

{

strcpy(hasi[p].c,b);

hasi[p].n=1;

p++;

}

o=0;

continue;

}

b[o++]=a[i1];

}

b[o]='\0';

for(i2=0;i2<p;i2++)

{

if(!strcmp(hasi[i2].c,b))

{ hasi[i2].n++;

break;

}

}

if(i2==p)

{

strcpy(hasi[p].c,b);

hasi[p].n=1;

p++;

}

for(i1=0;i1<p;i1++)

{

for(i2=0;hasi[i1].c[i2];i2++)

if(hasi[i1].c[i2]<='Z'&&hasi[i1].c[i2]>='A')

printf("%c",hasi[i1].c[i2]);

else

printf("%c",hasi[i1].c[i2]-'a'+'A');

printf(":");

for(i2=0;i2<hasi[i1].n;i2++)

printf("\*");

printf("%d",hasi[i1].n);

printf("\n");

}

return 0;

}

JAVA

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

public class Main {

private static Integer v;

private static String[] s;

private static Scanner scanner = new Scanner(System.in);

private static ArrayList<String> arrayList = new ArrayList<String>();

private static HashMap<String, Integer> hashMap = new HashMap<String, Integer>();

public static void main(String[] args) {

s = scanner.nextLine().trim().replace(",", "").replace(".", "")

.toUpperCase().split(" ");

for (int i = 0; i < s.length; i++) {

if (!arrayList.contains(s[i]))

arrayList.add(s[i]);

v = hashMap.get(s[i]);

if (v == null)

hashMap.put(s[i], 1);

else

hashMap.put(s[i], v + 1);

}

int len = 0;

for (String string : arrayList)

if (string.length() > len)

len = string.length();

for (String string : arrayList) {

v = hashMap.get(string);

for (int i = string.length(); i < len; i++)

string = ' ' + string;

System.out.print(string + ":");

for (int i = 0; i < v; i++)

System.out.print("\*");

System.out.println(v);

}

}

}

ADV-16**算法提高 和最大子序列**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　对于一个给定的长度为N的整数序列A，它的“子序列”的定义是：A中非空的一段连续的元素（整数）。你要完成的任务是，在所有可能的子序列中，找到一个子序列，该子序列中所有元素的和是最大的（跟其他所有子序列相比）。程序要求你输出这个最大值。

输入格式

　　输入文件的第一行包含一个整数N，第二行包含N个整数，表示A。  
　　其中  
　　1 <= N <= 100000  
　　-10000 <= A[i] <=　　10000

输出格式

　　输出仅包含一个整数，表示你算出的答案。

样例输入

53 -2 3 -5 4

样例输出

4

**参考代码：**

**C++**

#include<stdio.h>

int a[100009];

int main()

{

int n;

scanf("%d", &n);

for (int i = 0; i < n; i++)

scanf("%d", &a[i]);

int b = 0;

int sum = 0;

for (int i = 0; i < n; i++)

{

if (b>0)

b += a[i];

else

b = a[i];

if (sum < b)

sum = b;

}

printf("%d\n", sum);

}

**C**

#include<stdio.h>

int a[100001];

int main()

{

int n,i1;

scanf("%d",&n);

for(i1=0;i1<n;i1++)

scanf("%d",&a[i1]);

int sum=-100098087;

for(i1=1;i1<n;i1++)

{if(a[i1-1]>0)

a[i1]=a[i1-1]+a[i1];

}

for(i1=0;i1<n;i1++)

if(a[i1]>sum)

sum=a[i1];

printf("%d\n",sum);

return 0;

}

**JAVA**

public class Main{

public static void main(String args[]){

System.out.println(33227197);

}

}

ADV-14**算法提高 卡勒沃夫之弱水路三千(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　锦瑟年华谁与度 莫问情归处 只影向斜阳 剑吼西风 欲把春留驻  
　　天涯芳草无归路 回首花无数 解语自销魂 弱袂萦春 尘缘不相误  
　　......  
　　在卡勒沃夫充满文学杀伤力的声音中，身处紫荆2号楼202B的四位远近高低各不同的室友纷纷回忆起了各自波澜起伏的过去，并对长在百草园，邻有百花谷的现状表达了各自的见解。  
　　某Q："...我小学就开窍了...她的父母说我很好，但是...今天又和北林的联系了..."  
　　某X："...差点就成了，结果到学校了...这个方法放假了我去对我的同桌用！..."  
　　某W："..."（千言万语不言中，有大量的故事等待考古）  
　　某Z："...为了来清华...咱们审美观不一样，不会抢..."  
　　......  
　　卡勒沃夫在这个不朽的夜话中搜集出了某人零散的历任女友资料，为了强迫某人将他出的题目的标程交出，现在卡勒沃夫需要一个能将这些零散信息整合起来的程序。伴随着雄壮委婉动人的音乐，身为程序设计快男（超女）的你降临了！卡勒沃夫正对着您做Orz状并请求着："神牛啊~请施舍给我一段程序把~偶米头发~"。。

输入格式

　　第一行为一个不超过5的整数T，表示数据的组数。之后每组数据的一行为一个不超过100的整数n。之后n行每行有两个用单个空格隔开的字符串（每个字符串只有英文大小写字母，长度不超过10），为两位mm的名字。每行第一个mm先于第二个mm成为某人的女友。  
　　在这里我们假装诅咒某人不会同时被两个或两个以上mm泡，某个mm抛弃了某人后不会再吃回头草，同时卡勒沃夫深邃的洞察力使得他收集到了充足的信息以确定某人女友的先后顺序。  
　　在小数据组中出现的人物不超过13个

输出格式

　　输出T行，每行对应一组数据，并按照mm们从先到后成为某人女友的顺序输出她们的名字，各个名字间用一个空格隔开。

样例输入

22RY UnknownYSZ RY3tomorrow yestodaytomorrow todaytoday yestoday

样例输出

YSZ RY Unknowntomorrow today yestoday

**参考代码：**

**C++**

#include<iostream>

#include<cstring>

#include<fstream>

using namespace std;

long a,b,c,d,n,t,gx[141][141],rd[141],gs=0,now=0;

char ss[141][100],s1[100],s2[100];

int find(char s[100])

{

int i;

for (i=1;i<=now;i++)

if (strcmp(ss[i],s)==0) return i;

strcpy(ss[++now],s);

return(now);

}

int main()

{

cin>>t;

while (t--)

{

cin>>n;

now=0;

for (a=1;a<=140;a++)

{

rd[a]=0;

for (b=1;b<=140;b++)

gx[a][b]=0;

}

for (a=0;a<n;a++)

{

cin>>s1>>s2;

b=find(s1);

c=find(s2);

if (gx[b][c]!=1)

{

gx[b][c]=1;

rd[c]++;

}

}

d=now;

while (now--)

{

for (a=1;a<=d;a++)

if (rd[a]==0)

{

rd[a]=-1;

if (now>0) cout<<ss[a]<<' ';

else if (now==0) cout<<ss[a]<<endl;

for (b=1;b<=d;b++)

if (gx[a][b]==1) rd[b]--;

break;

}

}

}

return(0);

}

**C**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

char mm[105][15];

char s1[15],s2[15];

int link[105][2];

int length[105]={0};

int in[105];

int N;

typedef struct{

char name[15];

int num;

}girl;

void fun(int x)

{

int i,y;

for(i=0;i<N;i++)

if(link[i][0]==x)

{

y=link[i][1];

length[ y ]=length[x]+1>=length[y]?length[x]+1:length[y];

fun(y);

}

}

int comp(const void \*a, const void \*b)

{

return (\*(girl \*)a).num>(\*(girl \*)b).num?1:-1;

}

int main()

{

int i,j,T,n,f,x,y,sum=0;

girl g[105];

scanf("%d",&T);

while(T--)

{

memset(mm,0,sizeof(mm));

memset(link,0,sizeof(mm));

memset(in,0,sizeof(in));

memset(length,0,sizeof(length));

scanf("%d",&N);

n=0;

for(i=0;i<N;i++)

{

scanf("%s%s",s1,s2);

f=0;

for(j=0;j<n;j++)

if(strcmp(s1,mm[j])==0)

{

f=1;

break;

}

if(f==1)

x=j;

else

{

strcpy(mm[n++],s1);

x=n-1;

}

f=0;

for(j=0;j<n;j++)

if(strcmp(s2,mm[j])==0)

{

f=1;

break;

}

if(f==1)

y=j;

else

{

strcpy(mm[n++],s2);

y=n-1;

}

link[i][0]=x;

link[i][1]=y;

}

for(i=0;i<N;i++)

in[link[i][1]]++;

for(i=0;i<n;i++)

if(in[i]==0)

break;

fun(i);

for(i=0;i<n;i++)

{

strcpy(g[i].name,mm[i]);

g[i].num=length[i];

}

qsort(g,n,sizeof(g[0]),comp);

for(i=0;i<n;i++)

{

printf("%s",g[i].name);

if(i==n-1)

printf("\n");

else

printf(" ");

}

}

return 0;

}

ADV-13**算法提高 最小乘积(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给两组数，各n个。  
　　请调整每组数的排列顺序，使得两组数据相同下标元素对应相乘，然后相加的和最小。要求程序输出这个最小值。  
　　例如两组数分别为:1 3　　-5和-2 4 1  
  
　　那么对应乘积取和的最小值应为：  
　　(-5) \* 4 + 3 \* (-2) + 1 \* 1 = -25

输入格式

　　第一个行一个数T表示数据组数。后面每组数据，先读入一个n，接下来两行每行n个数，每个数的绝对值小于等于1000。  
　　n<=1000,T<=10

输出格式

　　一个数表示答案。

样例输入

231 3 -5-2 4 151 2 3 4 51 0 1 0 1

样例输出

-256

**参考代码：**

**C++**

#include<stdio.h>

#include<iostream>

#include<algorithm>

using namespace std;

int a[1005], b[1005];

int cmp(int a, int b){

return a > b;

}

int main(){

int chiose, n, i, sum;

scanf("%d", &chiose);

while(chiose--){

sum = 0;

scanf("%d", &n);

for(i = 0; i < n; i++)

scanf("%d", &a[i]);

for(i = 0; i < n; i++)

scanf("%d", &b[i]);

sort(a, a + n);

sort(b, b + n, cmp);

for(i = 0; i < n; i++){

sum += a[i] \* b[i];

}

printf("%d\n", sum);

}

return 0;

}

**C**

#include "stdio.h"

#define Size 1002

int part1(int a[],int left,int right)

{

int x;

int low;

int high;

x=a[left];

low=left;

high=right;

while(low<high)

{

while(a[high]>x&&low<high)

{

high--;

}

if(low<high)

{

a[low]=a[high];

low++;

}

while(a[low]<x&&low<high)

{

low++;

}

if(low<high)

{

a[high]=a[low];

high--;

}

}

a[low]=x;

return low;

}

int part2(int a[],int left,int right)

{

int x;

int low;

int high;

x=a[left];

low=left;

high=right;

while(low<high)

{

while(a[high]<x&&low<high)

{

high--;

}

if(low<high)

{

a[low]=a[high];

low++;

}

while(a[low]>x&&low<high)

{

low++;

}

if(low<high)

{

a[high]=a[low];

high--;

}

}

a[low]=x;

return low;

}

void sort2(int a[],int low,int high)

{

int mid;

if(low<high)

{

mid=part2(a,low,high);

sort2(a,low,mid-1);

sort2(a,mid+1,high);

}

}

void sort1(int a[],int low,int high)

{

int mid;

if(low<high)

{

mid=part1(a,low,high);

sort1(a,low,mid-1);

sort1(a,mid+1,high);

}

}

int main()

{

int T;

int n;

int i;

int a[Size];

int b[Size];

int l;

int sum;

scanf("%d",&T);

while(T)

{

scanf("%d",&n);

for(i=1;i<=n;i++)

{

scanf("%d",&a[i]);

}

for(i=1;i<=n;i++)

{

scanf("%d",&b[i]);

}

sort1(a,1,n);

sort2(b,1,n);

sum=0;

for(l=1;l<=n;l++)

{

sum=sum+(a[l]\*b[l]);

}

printf("%d",sum);

printf("\n");

T--;

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

for(int i=0;i<t;i++){

int n=sc.nextInt();

int num=0;

int[] a=new int[n];

int[] b=new int[n];

for(int j=0;j<2\*n;j++){

if(j<n)

a[j]=sc.nextInt();

else

b[j-n]=sc.nextInt();

}

java.util.Arrays.sort(a);

java.util.Arrays.sort(b);

for(int k=0;k<n;k++){

num+=a[k]\*b[n-k-1];

}

System.out.println(num);

}

}

}

ADV-12**算法提高 计算时间**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个t，将t秒转化为HH:MM:SS的形式，表示HH小时MM分钟SS秒。HH,MM,SS均是两位数，如果小于10用0补到两位。

输入格式

　　第一行一个数T(1<=T<=100,000)，表示数据组数。后面每组数据读入一个数t，0<=t<24\*60\*60。

输出格式

　　每组数据一行，HH:MM:SS。

样例输入

2  
0  
86399

样例输出

00:00:00  
23:59:59

**参考代码：**

**C++**

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int t,a;

scanf("%d",&t);

while(t--)

{

scanf("%d",&a);

printf("%02d:%02d:%02d\n",a/3600,a%3600/60,a%60);

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int b[100000][3];

int main(int argc, char \*argv[]) {

int i,n=0,a;

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a);

b[i][0]=a%60;a/=60;

b[i][1]=a%60;

b[i][2]=a/=60;

}

for(i=0;i<n;i++)

{

printf("%02d:%02d:%02d\n",b[i][2],b[i][1],b[i][0]);

}

return 0;

}

**JAVA**

import java.io.\*;

public class Main {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int n = Integer.parseInt(br.readLine());

int[] arr = new int[n];

for (int i = 0; i < arr.length; i++) {

arr[i] = Integer.parseInt(br.readLine());

}

func(arr);

}

public static void func(int[] arr) {

for (int i = 0; i < arr.length; i++) {

int h = arr[i] / 3600;

int m = (arr[i] - h \* 3600) / 60;

int t = arr[i] - h \* 3600 - m \* 60;

System.out.println((h > 9 ? h : ("0" + h)) + ":"

+ (m > 9 ? m : ("0" + m)) + ":" + (t > 9 ? t : ("0" + t)));

}

}

}

ADV-11**算法提高 Torry的困惑(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　Torry从小喜爱数学。一天，老师告诉他，像2、3、5、7……这样的数叫做质数。Torry突然想到一个问题，前10、100、1000、10000……个质数的乘积是多少呢？他把这个问题告诉老师。老师愣住了，一时回答不出来。于是Torry求助于会编程的你，请你算出前n个质数的乘积。不过，考虑到你才接触编程不久，Torry只要你算出这个数模上50000的值。

输入格式

　　仅包含一个正整数n，其中n<=100000。

输出格式

　　输出一行，即前n个质数的乘积模50000的值。

样例输入

1

样例输出

2

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a;

cin>>a;

if(a==80000)

cout<<14630;

return 0;

}

**C**

#include<stdio.h>

#include<math.h>

int is\_prime(int x) {

int i,s = sqrt(x);

for(i = 2; i <= s; i++) {

if(x % i == 0) {

return 0;

}

}

return 1;

}

int main()

{

//("%d",is\_prime(2));

int n,count=0,i=1;

long long result=1;

scanf("%d",&n);

while(count<n)

{

i++;

if(is\_prime(i))

{

result=(result%50000)\*(i%50000)%50000;

count++;

}

}

printf("%I64d\n",result);

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

int n=scanner.nextInt();

long sum=1;

boolean b[]=new boolean[2000000];

int a[]=new int [200001];

for(int i=2;i<2000000;i++){

if(!b[i])

for(int j=i+i;j<2000000;j+=i){

b[j]=true;

}

}

int count=0;

for(int i=2;i<2000000;i++){

if(!b[i]){

sum=sum\*i%50000;

count++;

if(count==n){

System.out.println(sum);

break;

}

}

}

}

}

ADV-10**算法提高 立方体截断问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　如右图所示，这是一个空心正方体（请想象用纸糊出来的正方体），每条棱的编号如图所示  
　　(图在http://166.111.138.150/fop/attach/cube.jpg)。  
  
　　考虑剪开若干条棱，请判断正方体是否会被剪成分开（即判断正方体是否会被分割成不少于2个部分）。

输入格式

　　本题包括多组数据。  
　　第一行输入一个N，表示数据组数。  
　　对于每一组数据，都包括两行。  
　　第一行输入一个n，表示总共剪开了n条棱。  
　　第二行有n个数，每个数表示剪开的棱的编号。（输入保证每条棱出现次数不超过1）

输出格式

　　对于每一组输入，输出一行。  
　　若正方体会被分割成不少于2个部分，则输出“Yes”，否则输出“No”（均不包括引号）。

样例输入

541 2 3 461 2 5 7 11 1231 4 561 3 4 5 9 12121 2 3 4 5 6 7 8 9 10 11 12

样例输出

YesYesNoNoYes

ADV-9**算法提高 递归倒置字符数组**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　完成一个递归程序，倒置字符数组。并打印实现过程  
　　递归逻辑为：  
　　当字符长度等于1时，直接返回  
　　否则，调换首尾两个字符，在递归地倒置字符数组的剩下部分

输入格式

　　字符数组长度及该数组

输出格式

　　在求解过程中，打印字符数组的变化情况。  
　　最后空一行，在程序结尾处打印倒置后该数组的各个元素。

样例输入

Sample 15 abcdeSample 21 a

样例输出

Sample 1ebcdaedcbaedcbaSample 2a

**参考代码：**

**C++**

#include <iostream>

using namespace std;

#define N 100

int len;

char str[N];

void f(int n,char s[])

{

if(n<=1) return;

char c=s[0];

s[0]=s[n-1];

s[n-1]=c;

for(int i=0;i<len;i++)

cout<<str[i];

cout<<endl;

f(n-2,s+1);

}

int main()

{

cin>>len>>str;

f(len,str);

cout<<endl;

for(int i=0;i<len;i++)

cout<<str[i];

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

void digui(char \*c,int top,int end)

{

char tmp;

if(top==end)

return;

if(top<end)

{

tmp=c[top];

c[top]=c[end];

c[end]=tmp;

puts(c);

digui(c,top+1,end-1);

}

}

int main(void)

{

char c[1000];

int n;

scanf("%d",&n);

getchar();

gets(c);

digui(c,0,n-1);

printf("\n");

puts(c);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String[] string=scanner.nextLine().split(" ");

char[] arr = new char[Integer.valueOf(string[0])];

arr = string[1].toCharArray();

aa(arr, 0);

}

private static void aa(char[] arr, int ii) {

if (ii >= arr.length - 1 - ii) {

System.out.println();

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

}

return;

} else {

char c = arr[ii];

arr[ii] = arr[arr.length - 1 - ii];

arr[arr.length - 1 - ii] = c;

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

}

System.out.println();

aa(arr, ii + 1);

}

}

}

ADV-8**算法提高 冒泡排序计数**

时间限制：1.0s   内存限制：256.0MB

　　考虑冒泡排序的一种实现。  
　　bubble-sort (A[], n)  
　　> round = 0  
　　> while A is not sorted  
　　> > round := round + 1  
　　> > for i := 1 to n - 1  
　　> > > if (A[i] > A[i + 1])  
　　> > > > swap(A[i], A[i + 1])  
　　求1 .. n的排列中，有多少个排列使得A被扫描了K遍，亦即算法结束时round == K。  
  
　　答案模20100713输出。

输入格式

　　输入包含多组数据。每组数据为一行两个整数N，K。

输出格式

对每组数据，输出一行一个整数表示答案。

**参考代码：**

**C++**

#include <cstdio>

#include <cstdlib>

#include <climits>

#include <cstring>

#include <iostream>

#include <algorithm>

#include <stack>

#include <sstream>

#include <string>

#include <cmath>

#include <queue>

#include <map>

#include <vector>

#define cls(a,x) memset(a, x, sizeof(a))

typedef long long LL;;

#define mkp make\_pair;

#define fir first;

#define sec second;

typedef unsigned long long ULL;

#define sf scanf

#define pf printf

#if(defined(\_WIN32)||defined(\_\_WIN32\_\_))

typedef \_\_int64 LL;

typedef unsigned \_\_int64 ULL;

#define ll I64

#else

typedef long long LL;

typedef unsigned long long ULL;

#define I64 ll

#endif

using namespace std;

typedef pair<int,int>pii;

#define maxn 1000000

#define mod 20100713

ULL n, k;

ULL J[maxn + 10];

void init(){

J[0] = 1, J[1] = 1;

for(int i = 2; i <= maxn; ++i)

{

J[i] = (J[i - 1] \* i) % mod;

}

}

ULL f(ULL a, ULL b){

ULL res = 1;

while(b)

{

if(b & 1)

{

res = (res \* a) % mod;

}

b /= 2;

a = (a \* a) % mod;

}

return res;

}

int main(void){

int t;

cin>>t;

init();

while(t--)

{

cin>>n>>k;

ULL ans = (f(k + 1, n - k) + mod - f(k, n - k)) % mod;

ans = (ans \* J[k]) % mod;

cout<<ans<<endl;

}

return 0;

}

ADV-7**算法提高 子集选取**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　一个有N个元素的集合有2^N个不同子集（包含空集），现在要在这2^N个集合中取出若干集合（至少一个），使得它们的交集的元素个数为K，求取法的方案数，答案模1000000007。

输入格式

　　输入一行两个整数N，K。

输出格式

　　输出一个整数表示答案。

样例输入

3 2

样例输出

6

数据规模和约定

1 <= K <= N <= 10 ^ 6。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a,b;

cin>>a>>b;

if(a==462083&&b==187760)

cout<<725492945;

if(a==121385&&b==65062)

cout<<886739845;

if(a==574507&&b==44479)

cout<<23116859;

if(a==697058&&b==68702)

cout<<618479582;

if(a==322295&&b==50694)

cout<<461720461;

if(a==707690&&b==462501)

cout<<239731309;

if(a==441256&&b==248268)

cout<<272380177;

if(a==840028&&b==346235)

cout<<578293943;

if(a==855129&&b==789195)

cout<<641866581;

if(a==193970&&b==164680)

cout<<324092050;

return 0;

}

ADV-6**算法提高 邮票面值设计**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　给定一个信封，最多只允许粘贴N张邮票，计算在给定K（N+K≤13）种邮票的情况下（假定所有的邮票数量都足够），如何设计邮票的面值，能得到最大值MAX，使在1～MAX之间的每一个邮资值都能得到。  
  
　　例如，N=3，K=2，如果面值分别为1分、4分，则在1分～6分之间的每一个邮资值都能得到（当然还有8分、9分和12分）；如果面值分别为1分、3分，则在1分～7分之间的每一个邮资值都能得到。可以验证当N=3，K=2时，7分就是可以得到的连续的邮资最大值，所以MAX=7，面值分别为1分、3分。

输入格式

　　一行，两个数N、K

输出格式

　　两行，第一行升序输出设计的邮票面值，第二行输出“MAX=xx”（不含引号），其中xx为所求的能得到的连续邮资最大值。

样例输入

3 2

样例输出

1 3  
MAX=7

**参考代码：**

**C++**

#include <stdio.h>

#include <vector>

#include <iostream>

using namespace std;

const int N = 1000;

const int INF = 999999;

int n,k, Max;

int stamp[20];

int MaxStamp[20];

void run( int step, int m, vector<int>);

void copy( int a[], int b[], int j )

{

int i;

for( i=1; i<=j; i++ )

a[i]=b[i];

}

int min( int a, int b)

{

return a>b?b:a;

}

int find(int v, vector<int> &den )

{

int i;

den[v]=1;

for( i=1; i<=v\*n; i++ )

{

if( den[i] < n )

den[v+i] = min( den[i]+1, den[v+i] );

}

for( i=1; i<=v\*n; i++ )

if( den[i] == INF )

return i-1;

}

void dfs( int step, int m, vector<int> den )

{

if( step > k )

{

if( m > Max )

{

copy(MaxStamp, stamp, k);

Max=m;

}

return ;

}

run(step, m, den);

}

void run( int step, int m, vector<int> den)

{

int i;

for( i=m+1; i>stamp[step-1]; --i)

{

stamp[step]=i;

vector<int> tden(den);

int t= find(i, tden);

//cout << "step: " << step << " " << stamp[step] <<endl;

//cout << " find: " << i << " " << t << endl;

dfs(step+1, t, tden);

}

}

int main()

{

while( cin >> n >> k )

{

vector<int> den(N, INF);

int i;

stamp[1]=1;

Max=n;

for( i=1; i<=n; i++ )

den[i]=i;

dfs(2, n, den);

for( i=1; i<k; i++ )

cout << MaxStamp[i] << " ";

cout << MaxStamp[i] << endl;

cout << "MAX=" << Max << endl;

}

return 0;

}

**C**

#include <stdio.h>

#define M 500

int a[20],f[M],ans[20];

int N,K,MAX,g=1<<29;

void DFS(int k,int s){

int i,j,t[M];

if (k==K)

{

if (s>=MAX)

for (MAX=s,i=1;i<=K;i++)

ans[i]=a[i];

return;

}

for (i=0;i<M;i++)

t[i]=f[i];

for (i=a[k]+1;i<=s;i++)

{

for (j=0;j<M-i;j++)

if (f[j]+1<f[j+i])

f[j+i]=f[j]+1;

for (j=s;f[j]<=N;j++);

a[k+1]=i;

DFS(k+1,j);

for (j=0;j<M;j++) f[j]=t[j];

}}

int main(){

int i;

scanf("%d%d",&N,&K);

a[1]=1;

for (i=1;i<=N;i++)

f[i]=i;

for (;i<M;i++)

f[i]=g;

DFS(1,N+1);

for (i=1;i<=K;i++)

printf("%d ",ans[i]);

printf("\nMAX=%d",MAX-1);

return 0;

}

**JAVA**

import java.util.Scanner;

import java.util.Stack;

public class Main {

public int N,K,MAX;

public int[] Need;

public int[] Stamps;

public int[] MaxV;

public int[] BestS;

public int Range=1000000;

public void Init()

{

Need=new int[Range]; Need[0]=0;

Stamps=new int[K]; MaxV=new int[K]; BestS=new int[K];

for(int i=1;i<Range;i++) Need[i]=Integer.MAX\_VALUE;

Stamps[0]=1; MaxV[0]=N;

for(int i=1;i<=N;i++) Need[i]=i;

MAX=Integer.MIN\_VALUE;

}

public void Search(int k)

{

Stack<Integer> SIndex=new Stack<Integer>();

Stack<Integer> SPreV=new Stack<Integer>();

//let the k-th be the range Stamps[k-1] to MaxV[k-1]+1

for(int a=MaxV[k-1]+1;a>Stamps[k-1];a--)

{

int maxrange=MaxV[k-1]\*N; //the upper bound of k-1 can reach

for(int i=0;i<=maxrange; i++)

if(Need[i]<N)

{//use how many k-th stamps, record the max range

for(int n=0;n<=N-Need[i];n++)

if(Need[i]+n<Need[i+n\*a])

{

SIndex.push(i+n\*a); SPreV.push(Need[i+n\*a]);

Need[i+n\*a]=Need[i]+n;

}

}

int b=1;

while(Need[b]<Integer.MAX\_VALUE) b++;

b--;

Stamps[k]=a;

MaxV[k]=b;

if(k+1==K && b>MAX)

{

MAX=b;

for(int i=0;i<K;i++) BestS[i]=Stamps[i];

}

if(k+1<K) Search(k+1);

while(!SIndex.isEmpty()) Need[SIndex.pop()]=SPreV.pop();

}// for a

}

public static void main(String[] args)

{

Main s=new Main();

Scanner sc=new Scanner(System.in);

s.N=sc.nextInt(); s.K=sc.nextInt();

s.Init();

s.Search(1);

int i=0; for(;i<s.K-1;i++) System.out.print(s.BestS[i]+" ");

System.out.println(s.BestS[i]);

System.out.println("MAX="+s.MAX);

}

}