**算法训练 矩阵乘法**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入两个矩阵，分别是m\*s，s\*n大小。输出两个矩阵相乘的结果。

输入格式

　　第一行，空格隔开的三个正整数m,s,n（均不超过200）。  
　　接下来m行，每行s个空格隔开的整数，表示矩阵A（i，j）。  
　　接下来s行，每行n个空格隔开的整数，表示矩阵B（i，j）。

输出格式

　　m行，每行n个空格隔开的整数，输出相乘後的矩阵C（i，j）的值。

样例输入

2 3 2  
1 0 -1  
1 1 -3  
0 3  
1 2  
3 1

样例输出

-3 2  
-8 2  
  
**提示**  
矩阵C应该是m行n列，其中C(i,j)等于矩阵A第i行行向量与矩阵B第j列列向量的内积。  
例如样例中C(1,1)=(1,0,-1)\*(0,1,3) = 1 \* 0 +0\*1+(-1)\*3=-3

本题的C++参考代码如下：

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<ctype.h>

int main()

{

int m,s,n;

int sum=0;

int i,j,l;

int a[200][200],b[200][200],c[200][200];

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

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

{

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

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

}

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

{

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

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

}

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

{

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

{

for(l=0;l<s;l++)

{

sum+=(a[i][l]\*b[l][j]);

}

c[i][j]=sum;

sum=0;

}

}

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

{

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

printf("%d ",c[i][j]);

printf("\n");

}

return 0;

}

本题的C参考代码如下：

# include<stdio.h>

int main(){

int m,s,n,i,j,k,a[200][200],b[200][200],c[200][200];

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

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

for(j=1;j<=s;j++)

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

}

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

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

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

}

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

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

c[i][j]=0;

}

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

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

for(k=1;k<=s;k++){

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

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

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

printf("%d ",c[i][j]);

printf("\n");

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static int m,s,n;

public static int[][]a;

public static int[][]b;

public static int[][]c;

public static boolean input() throws IOException{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str=br.readLine();

String[] strArr=str.split(" ");

if(strArr.length<3) return false;

m=Integer.parseInt(strArr[0]);

s=Integer.parseInt(strArr[1]);

n=Integer.parseInt(strArr[2]);

if(m<1||s<1||n<1) return false;

a=new int[m][s];

b=new int[s][n];

c=new int[m][n];

String[] str\_a=new String[m];

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

str\_a[i]=br.readLine();

String [] m\_s\_arr=str\_a[i].split(" ");

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

a[i][j]=Integer.parseInt(m\_s\_arr[j]);

}

}

String[] str\_b=new String[s];

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

str\_b[i]=br.readLine();

String[] s\_n\_arr=str\_b[i].split(" ");

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

b[i][j]=Integer.parseInt(s\_n\_arr[j]);

}

}

return true;

}

public static void cal(){

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

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

int sum=0;

for(int k=0;k<s;k++){

sum+=a[i][k]\*b[k][j];

}

c[i][j]=sum;

}

}

}

public static void print(){

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

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

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

}

System.out.println();

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

try {

if(input()){

cal();

print();

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}