/\*

@author: Jose Luis Rocabado, Octavio Sales

\*/

#include<stdio.h>

#include <algorithm>

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <time.h>

#include <string>

#include <vector>

#define NU 0.000001

#define K 10000

int static pat;

float init\_w() {

return(float)rand() / (float)RAND\_MAX;

}

void training(int cnt, float &y, std::vector<std::vector<float>> x, std::vector<float> &w, std::vector<float> z) {

int k = 0;

while (k < K) { //STEP 4

for (int i = 0; i < pat; i++) { //STEP 7

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

y += x[i][j] \* w[j]; //STEP 8

}

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

w[j] = w[j] + (NU \* (z[i] - y)\*x[i][j]); //STEP 9

}

}

k++;

}

}

int main() {

std::ifstream in("patterns1.txt");

auto cinbuf = std::cin.rdbuf(in.rdbuf());

//std::cin.rdbuf(in.rdbuf());

std::string aux;

int j, cnt=0;

for (int i = 0; i < 2; ++i) {

std::cin >> aux;

std::cout << aux << std::endl;

}

std::cin >> j;

pat = j;

float y = 0;

//std::vector<float> x(pat);

do{

std::cin >> aux;

cnt++;

}while(aux != "y");

std::vector<std::vector<float>> x(pat, std::vector<float>(cnt)); //Matrix of input patterns

std::vector<float> w(cnt); //vector of wieghts

std::vector<float> z(pat); //vector of output patterns

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

for (int j = 0; j < cnt; ++j)

std::cin >> x[i][j];

std::cin >> z[i];

}

for (int j = 0; j < cnt; ++j)

w[j] = init\_w(); //STEP 3

training(cnt, y, x, w, z);

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

std::cout << w[i] << " ";

std::cin.rdbuf(cinbuf);

system("PAUSE");

return 0;

}