大疆科技的校招笔试题（的一个小编程题）

底下的答案仅供参考。

**//两个小车，走一步能量消耗1，方向为1向右，-1为向左，首先输入路途长度，然后输入两行，每行第一个为小车的能量，第二个位小车起始位置，第三个为方向。求几个小车可以走出去？**

#include <iostream>  
#include <vector>  
using namespace std;  
  
  
int Many(vector<int> &ve,int len)  
{  
if(ve.size() < 6 || len <= 0)  
return 0;  
int end = 0;  
int power\_a = ve[0];  
int local\_a = ve[1];  
int direc\_a = ve[2];  
int power\_b = ve[3];  
int local\_b = ve[4];  
int direc\_b = ve[5];  
while(power\_a > 0 || power\_b > 0)  
{  
if(direc\_a == direc\_b)  
{  
if(direc\_a == 1)  
{  
if(local\_a + 1 == local\_b && power\_b == 0)  
{  
if(power\_a > 1)  
{  
local\_a++;  
local\_b++;  
power\_a -= 2;  
}  
}  
if(power\_a > 0)  
{  
local\_a++;  
power\_a--;  
}  
if(power\_b > 0)  
{  
local\_b++;  
power\_b--;  
}  
}  
else  
{  
if(local\_a + 1 == local\_b && power\_a == 0)  
{  
if(power\_b > 1)  
{  
local\_a--;  
local\_b--;  
power\_a -= 2;  
}  
}  
if(power\_a > 0)  
{  
local\_a--;  
power\_a--;  
}  
if(power\_b > 0)  
{  
local\_b--;  
power\_b--;  
}  
}  
}  
else  
{  
if(local\_a < 0 || local\_a > len-1)  
power\_a = 0;  
if(local\_b < 0 || local\_b > len-1)  
power\_b = 0;  
if(abs(local\_a - local\_b) == 2 && direc\_a == 1 && direc\_b == -1)  
{  
power\_a -= 2;  
power\_b -= 2;  
}  
else  
{  
if(abs(local\_a - local\_b) == 1)  
{  
if(direc\_a == 1 && direc\_b == -1)  
{  
local\_a--;  
local\_b++;  
power\_a -= 2;  
power\_b -= 2;  
}  
else  
{  
local\_a --;  
local\_b++;  
power\_a --;  
power\_b++;  
}  
}  
else  
{  
if(direc\_a == -1)  
{  
local\_a --;  
power\_a--;  
}  
else  
{  
local\_a++;  
power\_a--;  
}  
if(direc\_b == -1)  
{  
local\_b --;  
power\_b--;  
}  
else  
{  
local\_b++;  
power\_b--;  
}  
}  
}  
}  
}  
if(local\_a < 0 || local\_a > len-1)  
end++;  
if(local\_b < 0 || local\_b > len-1)  
end++;  
return end;  
}  
int main()  
{  
vector<int> ve;  
int len;  
cin>>len;  
int tmp;  
for(int i = 0; i < 6; ++i)  
{  
cin>>tmp;  
ve.push\_back(tmp);  
}  
cout<<Many(ve,len)<<endl;  
}  
  
  
//一共N种花，插花需要每次选M种，每种R支。第二行输入每种花个数，求最多有多少种插花方法。  
#include <iostream>  
#include <vector>  
#include <algorithm>  
using namespace std;  
  
  
bool fun(vector<int> &ve,int m)  
{  
int i = ve.size()-1;  
int num = 0;  
for(; i >= 0; --i)  
{  
if(ve[i] != 0)  
num++;  
else  
break;  
}  
if(num < m)  
return false;  
return true;  
}  
int Many(vector<int> &ve,int n,int m,int r,int num)  
{  
if(ve.size() <= 0 || n <= 0 || m <= 0 || r <= 0)  
return 0;  
int end = 0;  
int i = ve.size()-1;  
while(fun(ve,m))  
{  
int size = i;  
sort(ve.begin(),ve.end());  
for(int j = 0; j < m; ++j)  
{  
ve[size] --;  
size--;  
}  
end++;  
num -= m;  
}  
return end;  
}  
int main()  
{  
int n,m,r;  
cin>>n>>m>>r;  
vector<int> ve(n);  
int tmp;  
int num = 0;  
for(int i = 0; i < n; ++i)  
{  
cin>>tmp;  
tmp /= r;  
ve.push\_back(tmp);  
num += tmp;  
}  
cout<<Many(ve,n,m,r,num)<<endl;  
}  
  
  
//输入初始位置和结束位置，以及二维数组的大小，与其中的元素，为0可以走，为1，其上下左右不能走，如果为2，则该位置的上两个，下两个，左两个，右两个不能走。以此类推，求最短路径？

#include <iostream>  
#include <vector>  
using namespace std;  
  
  
vector<int> endpath;  
void Get\_num(vector<vector<int>> &path,int n,int m)  
{  
for(int i = 0; i < n; ++i)  
{  
for(int j = 0; j < m; ++j)  
{  
if(path[i][j] != -2)  
{  
int tmp = 0;  
for(int k = 0; k < path[i][j]; k++)  
{  
tmp = i - k - 1;  
if(tmp >= 0)  
path[tmp][j] = -2;  
else  
break;  
}  
for(int k = 0; k < path[i][j]; k++)  
{  
tmp = i + k + 1;  
if(tmp < n)  
path[tmp][j] = -2;  
else  
break;  
}  
for(int k = 0; k < path[i][j]; k++)  
{  
tmp = j - k - 1;  
if(tmp >= 0)  
path[i][tmp] = -2;  
else  
break;  
}  
for(int k = 0; k < path[i][j]; k++)  
{  
tmp = j + k + 1;  
if(tmp < m)  
path[i][tmp] = -2;  
else  
break;  
}  
}  
}  
}  
}  
void Less\_len(vector<vector<int>> &path,int n,int m,int i,int j,int endi,int endj,int num)  
{  
path[i][j] = -1;  
if(i == endi && j == endj)  
endpath.push\_back(num);  
if(i < n && j < m)  
{  
if(i - 1 >= 0 && path[i-1][j] == 0)  
Less\_len(path,n,m,i-1,j,endi,endj,num+1);  
else if(i + 1 < n && path[i+1][j] == 0)  
Less\_len(path,n,m,i+1,j,endi,endj,num+1);  
else if(j - 1 >= 0 && path[i][j-1] == 0)  
Less\_len(path,n,m,i,j-1,endi,endj,num+1);  
else if(j + 1 < m && path[i][j+1] == 0)  
Less\_len(path,n,m,i,j+1,endi,endj,num+1);  
}  
}  
int Less\_len(vector<int> &ve,vector<vector<int>> &path,int n,int m)  
{  
if(ve.size() <= 0 || path.size() <= 0 || n <= 0 || m <= 0)  
return -1;  
Get\_num(path,n,m);  
int num = 0;  
Less\_len(path,n,m,ve[0],ve[1],ve[2],ve[3],num);  
  
  
int less = 0;  
for(int i = 0; i < endpath.size(); ++i)  
{  
if(i == 0)  
less = endpath[i];  
if(less > endpath[i])  
less = endpath[i];  
}  
return less;  
}  
int main()  
{  
int n,m;  
vector<int> ve;  
int tmp;  
cin>>n>>m;  
for(int i = 0; i < 4; ++i)  
{  
cin>>tmp;  
ve.push\_back(tmp);  
}  
vector<vector<int>> path(n);  
for(int i = 0; i < n; ++i)  
{  
path[i].resize(m);  
}  
for(int i = 0; i < n; ++i)  
{  
for(int j = 0; j < m; ++j)  
{  
cin>>tmp;  
path[i][j] = tmp;  
}  
}  
cout<<Less\_len(ve,path,n,m)<<endl;  
return 0;  
}  
  
  
  
  
//H的水桶，注水速度X，h处有洞，流水速度Y，S秒以后水深（四舍五入）。  
#include <iostream>  
using namespace std;  
  
  
int Time(int H,double x,double y,double h,int s)  
{  
if(H < 0 || H > 5000 || x < 0 || x > 5000 || h < 0 || h > 5000  
|| s < 0 || s > 5000 || h > H)  
return -1;  
double end = x \* s - y \* (s - h / x);  
if(end >= H)  
return H;  
int tmp = end \* 10;  
tmp %= 10;  
if(tmp > 4)  
return end+1;  
return end;  
}  
int main()  
{  
int H,s;  
double h,x,y;  
cin>>H>>x>>y>>h>>s;  
cout<<Time(H,x,y,h,s)<<endl;  
return 0;  
}