int n;  
 int a[6]={8,1,2,7,3,9};

/\*  
 \* sorting with custom comparator  
 \* sort even first(small to great) odd second(great to small)  
 \*/

sort(a,a+n,[&](int i,int j){  
 int pi=i%2,pj=j%2;  
 if (pi==pj){  
 if (pi==0)//even , even  
 return i<j;  
 else//odd , odd  
 return i>j;  
 }  
 else if(pi==0)//even , odd  
 return true;  
 else//odd , even  
 return false;  
 });  
 for(auto i:a)  
 cout<<i<<" ";  
  
 cin>>n;  
 string s[n];  
 for (int i = 0; i < n; ++i)  
 cin>>s[i];

// concatenate for n strings, print the smallest lexicographical string

sort(s,s+n,[&](string s1,string s2){  
 string ch1 =s1+s2;  
 string ch2 =s2+s1;  
 return ch1<ch2;  
 });  
  
 for(auto i:s)  
 cout<<i;  
  
 int ind[n];  
 for (int i = 0; i < n; ++i)  
 ind[i]=i;  
 // ====  
 iota(ind,ind + n,0);

//concatenate for n strings, print the index smallest lexicographical string

sort(ind,ind+n,[&](int i,int j){  
 string s1=s[i],s2=s[j];  
 string ch1 =s1+s2;  
 string ch2 =s2+s1;  
 return ch1<ch2;  
 });  
  
 cout<<"\n";  
 for(auto i:ind)  
 cout<<i<<" ";

// stay a head (just walks forward)  
 // max number of lecturer

int x;  
 cin>>x;  
 int l[x],r[x],ind[x];  
 for (int i = 0; i < x; ++i)  
 cin>>l[i]>>r[i];  
 iota(ind,ind+x,0);  
 sort(ind,ind+x,[&](int x, int y){  
 return r[x]<r[y];  
 });  
 int end=0,c=0;  
 for (int i = 0; i < x; ++i) {  
 int j =ind[i];  
 if(l[j]>=end){  
 ++c;  
 end=r[j];  
 }  
 }  
 cout<<c;

// max number of noise(hsshhs)=>2+2+0

int n;  
 cin>>n;  
 string s[n];  
 int S[n],H[n];  
 for (int i = 0; i < n; ++i){  
 cin>>s[i];  
 S[i]= count(s[i].begin(),s[i].end(),'s');  
 H[i]=count(s[i].begin(),s[i].end(),'h');  
 }  
  
 auto score =[&](string &s){  
 int ch=0;  
 long long noise=0;  
 for (int i = s.size(); i >= 0; --i) {  
 if(s[i]=='h')  
 ch++;  
 else  
 noise+=ch;  
 }  
 return noise;  
 };  
 int ind[n];  
 iota(ind,ind+n,0);  
 sort(ind,ind+n,[&](int i, int j){  
 return 1ll\*S[i]\*H[j] > 1ll\*S[j]\*H[i];  
 });  
 string tot;  
 for (int x = 0; x < n; ++x) {  
 int i = ind[x];  
 tot+=s[i];  
 }  
 cout<<score(tot);  
  
 or  
  
 int n;  
 cin>>n;  
 string s[n];  
 for (int i = 0; i < n; ++i)  
 cin>>s[i];  
  
 auto score =[&](string &s){  
 int ch=0;  
 long long noise=0;  
 for (int i = s.size(); i >= 0; --i) {  
 if(s[i]=='h')  
 ch++;  
 else  
 noise+=ch;  
 }  
 return noise;  
 };  
 int ind[n];  
 iota(ind,ind+n,0);  
 sort(ind,ind+n,[&](int i, int j){  
 string s1=s[i]+s[j],s2=s[j]+s[i];  
 return score(s1)>score(s2);  
 });  
 string tot;  
 for (int x = 0; x < n; ++x) {  
 int i = ind[x];  
 tot+=s[i];  
 }  
 cout<<score(tot);  
  
 // Prefix sum  
int a[]={2,4,5,6,7,5,3,7,9};  
int n= sizeof(a)/ sizeof(a[0]);  
int Prefix\_sum[n]={0};  
 for (int i = 0; i < n ; ++i)  
 Prefix\_sum[i]=Prefix\_sum[i-1]+a[i];  
 for(auto i:Prefix\_sum)  
 cout<<i<<" ";  
//how many 5  
 int Prefix\_five[n]={0};  
 for (int i = 0; i < n ; ++i)  
 if (a[i]==5)  
 Prefix\_five[i]=Prefix\_five[i-1]+1;  
 else  
 Prefix\_five[i]=Prefix\_five[i-1];  
 for(auto i:Prefix\_five)  
 cout<<i<<" ";

//Partial sum  
//add k for rang[l,r]

int n,t;  
 cin>>n>>t;  
 int a[n+5]={0};  
 for (int i = 0; i < t; ++i) {  
 int l,r,k;  
 cin>>l>>r>>k;  
 a[l]+=k;  
 a[r+1]-=k;  
 }  
// for(int i: a)  
// cout<<i<<" ";  
 for (int i = 1; i <= n; ++i)  
 a[i]+=a[i-1];  
 for(int i: a)  
 cout<<i<<" ";

//Partial sum 2D

vector<vector<int>> A =  
 **{**{0,0,0,0,0,0},  
 {0,1,2,2,4,1},  
 {0,3,4,1,5,2},  
 {0,2,3,3,2,4},  
 {0,4,1,5,4,6},  
 {0,6,3,2,1,3}**}**;  
 for (int i = 0; i < (int)A.size() ; ++i)  
 for (int j = 0; j < (int)A[0].size(); ++j)  
 A[i][j] += A[i][j-1];  
  
 for (int j = 0; j < (int)A[0].size(); ++j)  
 for (int i = 0; i < (int)A.size() ; ++i)  
 A[i][j] += A[i-1][j];  
 int k,l,i,j;  
 cin>>i>>j>>k>>l;  
 int sum =A[k][l] - A[k][j-1] - A[i-1][l] +A[i-1][j-1];  
 cout<<sum;