**Dinic()**

**#include<cstdio>**

**#include<cstring>**

**#define INF 0x7fffffff**

**#define MaxNode 15000**

**#define MaxEdge 80000**

**#define MaxQue 1000000**

**int M,N,m0=1,Ans,Total,S,T;**

**int head[MaxNode],u[MaxEdge],v[MaxEdge],next[MaxEdge],con[MaxEdge];**

**int Q[MaxQue],dis[MaxNode],cur[MaxNode],vis[MaxNode];**

**int Min(int a,int b){return a<b?a:b;}**

**int Node(int x,int y){return 2+(x-1)\*N+y;}**

**void Add(int a,int b,int c)**

**{**

**v[++m0]=b;next[m0]=head[a];head[a]=m0;con[m0]=c;**

**v[++m0]=a;next[m0]=head[b];head[b]=m0;**

**}**

**int bfs()**

**{**

**int Qhead=0,Qtail=0;**

**memset(vis,0,sizeof(vis));**

**memset(dis,127,sizeof(dis));**

**dis[S]=0;vis[S]=1;Q[++Qtail]=S;**

**while(Qhead<Qtail)**

**{**

**++Qhead;**

**for(int i=head[Q[Qhead]];i;i=next[i])**

**if(!vis[v[i]] && con[i])**

**{**

**vis[v[i]]=1;**

**dis[v[i]]=dis[Q[Qhead]]+1;**

**Q[++Qtail]=v[i];**

**}**

**}**

**return vis[T];**

**}**

**int dfs(int now,int lim)**

**{**

**if (now==T || !lim) return lim;**

**int flow=0,f;**

**for (int& i=cur[now];i;i=next[i])**

**{**

**if (dis[v[i]]>dis[now] && con[i])**

**if (f=dfs(v[i],Min(lim-flow,con[i])))**

**{**

**flow+=f;**

**con[i]-=f;**

**con[i^1]+=f;**

**if (flow==lim) break;**

**}**

**}**

**return flow;**

**}**

**int DINIC()**

**{**

**int flow=0;**

**while(bfs())**

**{**

**memcpy(cur,head,sizeof(head));**

**flow+=dfs(S,INF);**

**}**

**return flow;**

**}**

**int main()**

**{**

**scanf("%d%d",&M,&N);**

**S=1;T=2;Total=0;**

**int A;**

**for (int i=1;i<=M;i++)**

**for (int j=1;j<=N;j++)**

**{**

**scanf("%d",&A);**

**Total+=A;**

**if ((i+j)&1) Add(S,Node(i,j),A);**

**else Add(Node(i,j),T,A);**

**}**

**for(int i=1;i<=M;i++)**

**for(int j=1;j<=N;j++)**

**if ((i+j)&1)**

**{**

**if (i!=1)**

**Add(Node(i,j),Node(i-1,j),INF);**

**if (i!=M)**

**Add(Node(i,j),Node(i+1,j),INF);**

**if (j!=1)**

**Add(Node(i,j),Node(i,j-1),INF);**

**if (j!=N)**

**Add(Node(i,j),Node(i,j+1),INF);**

**}**

**Ans=DINIC();**

**printf("%d\n",Total-Ans);**

**return 0;**

**}**