



using namespace std;

int n,m,k;

int s[1005],g[1005];

int f[102][1005][1005],ans;

char ma[103];

int map[103];

int get(int x)//计算某一状态含有多少个1(即有多少个炮兵)

{

int e=0;

while(x>0)

{

++e;

x-=x&(-x);

}

return e;

}

int main()

{

cin>>n>>m;

for(int i=1;i<=n;++i)//读入地图，将山地（不能放兵）的地方设为1

{

scanf("%s",ma);

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

if(ma[j]=='H') map[i]+=1<<j;

}

for(int i=0;i<=(1<<m)-1;++i)//枚举所有的状态

if(((i&(i<<1))==0)&&((i&(i<<2))==0)&&((i&(i>>1))==0)&&((i&(i>>2))==0))//判断每个1左右各两个是否有1，即判断这种状态是否存在

{

++k;

s[k]=i;

g[k]=get(i);

if((i&map[1])==0) f[1][0][k]=g[k];//初始化第一行

}

//初始化第二行

for(int i=1;i<=k;++i)//枚举第一行状态

for(int j=1;j<=k;++j)//枚举第二行状态

if(((s[i]&s[j])==0)&&((s[j]&map[2])==0)) f[2][i][j]=max(f[2][i][j],f[1][0][i]+g[j]);//判断是否与地形和第一行冲突

//dp过程

for(int i=3;i<=n;++i)//枚举当前行数

for(int j=1;j<=k;++j)//枚举当前行数状态

if((map[i]&s[j])==0)//不与地形冲突

for(int p=1;p<=k;++p)//枚举前一行状态

if((s[p]&s[j])==0)//当前行状态不与前一行冲突

for(int q=1;q<=k;++q)//枚举前两行

//不与前两行冲突，且前两行自身不冲突

if(((s[q]&s[p])==0)&&((s[q]&s[j])==0)) f[i][p][j]=max(f[i][p][j],f[i-1][q][p]+g[j]);

for(int i=1;i<=k;++i)//枚举最后两行为结尾的情况，统计答案

for(int j=1;j<=k;++j)

ans=max(f[n][i][j],ans);

cout<<ans; //输出

return 0;

}