# AC

//5955

#include<bits/stdc++.h>

using namespace std;

const int M = 12345;

int ch[M][7];

int f[M],last[M],val[M];

bool vis[M];//与AC自动机每个点对应，用于统计

int ts[123];

int q[M];//手写队列

int L,L2;//L2是匹配串长度

double a[1234][1234];

int prin(int sz){

for(int i=1;i<=sz;i++){

for(int j=1;j<=sz+1;j++){

printf("%.2lf ",a[i][j]);

}

printf("\n");

}

printf("\n\n");

return 0;

}

int gauss(int width){

for(int i=1;i<=width;i++){

int r=i;

for(int k=i+1;k<=width;k++){

if(fabs(a[k][i])>fabs(a[r][i])) r=k;

}

for(int j=1;j<=width+1;j++){

swap(a[i][j],a[r][j]);

}

for(int k=i+1;k<=width;k++){

if(fabs(a[i][i])<1e-9) continue;

double kf=a[k][i]/a[i][i];

for(int j=i;j<=width+1;j++){

a[k][j]-=kf\*a[i][j];

}

}

}

//prin(width);

for(int i=width;i>=1;i--){

for(int k=i+1;k<=width;k++){

a[i][width+1]-=a[k][width+1]\*a[i][k];

}

a[i][width+1]/=a[i][i];

}

return 0;

}

int sz;//自动机长度

int init(){

memset(ch[0],0,sizeof(ch[0]));

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

val[i]=0;

vis[i]=0;

last[i]=0;

f[i]=0;

for(int j=0;j<7;j++)

ch[i][j]=0;

}

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

for(int j=0;j<=sz+2;j++){

a[i][j]=0.0;

}

}

sz=0;

return 0;

}

int len;

int insert(int s[]){

int u=0;

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

int c=s[i];

if(!ch[u][c]){

++sz;

ch[u][c]=sz;

}

u=ch[u][c];

}

val[u]++;

return 0;

}

int get\_fail(){

int left=1,right=0;

f[0]=0;

for(int c=1;c<7;c++){

int u=ch[0][c];

if(u)f[u]=0,q[++right]=u,last[u]=0;

}

while(left<=right){

int r=q[left];

left++;

if(left>M) left=0;

for(int c=1;c<7;c++){

int u=ch[r][c];

if(!u){

ch[r][c]=ch[f[r]][c];

continue;

}

right++;

if(right>M) right=0;

q[right]=u;

int v=f[r];

f[u]=ch[v][c];

last[u]=val[f[u]]?f[u]:last[f[u]];

}

}

return 0;

}

int n;

int main(){

int T;

cin>>T;

while(T--){

scanf("%d %d",&n,&len);

for(int i=1;i<=n;i++) {

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

cin>>ts[j];

insert(ts);

}

get\_fail();

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

a[i+1][i+1]+=-1.0;

a[i+1][sz+2]=0.0;

if(i==0) a[i+1][sz+2]=-1.0;

if(val[i]) continue;

for(int j=1;j<=6;j++){

int k=ch[i][j];

a[k+1][i+1]+=1.0/6.0;

}

}

sz++;

gauss(sz);

double ans=0.0;

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

if(val[i-1]) ans+=a[i][sz+1];

}

int ct=0;

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

if(val[i-1]){

ct++;

printf("%lf",a[i][sz+1]/ans);

if(ct<n) printf(" ");

}

}

printf("\n");

init();

}

return 0;

}

//6086

#include<cstdio>

#include<queue>

#include<string>

#include<iostream>

#include<algorithm>

#include<cstring>

using namespace std;

queue<int>q;

#define MOD 998244353

int child[2645][2],fail[2645],size,f[2][2645][64],tag[2645],tag2[2645];

void Insert(string S,int id)

{

int len=S.length();

int now=0;

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

{

if(!child[now][S[i]-'0'])

child[now][S[i]-'0']=size++;

now=child[now][S[i]-'0'];

}

tag[now]|=(1<<id);

}

void Inser2(string S,int id){

int len=S.length();

int now=0;

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

{

if(!child[now][S[i]-'0'])

child[now][S[i]-'0']=size++;

now=child[now][S[i]-'0'];

}

tag2[now]|=(1<<id);

}

void build()

{

fail[0]=-1;

q.push(0);

while(!q.empty())

{

int U=q.front(); q.pop();

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

if(child[U][i])

{

int V=fail[U];

while(V!=-1)

{

if(child[V][i])

{

fail[child[U][i]]=child[V][i];

break;

}

V=fail[V];

}

if(V==-1)

fail[child[U][i]]=0;

tag[child[U][i]]|=tag[fail[child[U][i]]];

tag2[child[U][i]]|=tag2[fail[child[U][i]]];

q.push(child[U][i]);

}

else if(U)

child[U][i]=child[fail[U]][i];

}

}

void Init()

{

memset(child,0,sizeof(child));

memset(fail,0,sizeof(fail));

memset(tag,0,sizeof(tag));

memset(tag2,0,sizeof(tag2));

size=1;

}

int n,m,T;

int main()

{

// freopen("b.in","r",stdin);

scanf("%d",&T);

string s;

for(;T;--T){

scanf("%d%d",&m,&n);

Init();

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

cin>>s;

int len=s.length();

// cout<<"::"<<s<<endl;

Insert(s,i);

string t=s;

reverse(t.begin(),t.end());

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

t[j]=((t[j]-'0')^1)+'0';

}

// cout<<"::"<<t<<endl;

Insert(t,i);

for(int j=0;j<min(len,n);++j){

bool flag=1;

for(int k=j+1,l=j;k<s.length() && l>=0;++k,--l){

if((s[k]-'0')^(s[l]-'0')!=1){

flag=0;

break;

}

}

if(!flag){

continue;

}

t=s.substr(0,j+1);

for(int k=(j+1)\*2;k<s.length();++k){

t=(char)(((s[k]-'0')^1)+'0')+t;

}

// cout<<":"<<t<<endl;

Inser2(t,i);

}

}

build();

memset(f,0,sizeof(f));

f[0][0][0]=1;

bool cur=0;

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

memset(f[cur^1],0,sizeof(f[cur^1]));

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

for(int k=0;k<(1<<m);++k){

if(!f[cur][j][k])

continue;

if(i<n-1)

for(int l=0;l<2;++l)

f[cur^1][child[j][l]][k|tag[child[j][l]]]=

(f[cur^1][child[j][l]][k|tag[child[j][l]]]+f[cur][j][k])%MOD;

else{

for(int l=0;l<2;++l)

f[cur^1][child[j][l]][k|tag[child[j][l]]|tag2[child[j][l]]]=

(f[cur^1][child[j][l]][k|tag[child[j][l]]|tag2[child[j][l]]]+f[cur][j][k])%MOD;

}

}

cur^=1;

}

int ans=0;

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

ans=(ans+f[cur][j][(1<<m)-1])%MOD;

printf("%d\n",ans);

}

return 0;

}