**PROBLEM\_1**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

#include <bits/stdc++.h>

using namespace std;

string maximizeNumber(string N, long long int M)

{

long long int i,j;

sort(N.begin(), N.end());

for ( i = 0; i < N.size(); i++)

{

for ( j = i + 1; j < N.size(); j++)

{

string t = N;

swap(t[j], t[i]);

if (stoll(t) > stoll(N) && stoll(t) <= M)

swap(N[i], N[j]);

}

}

return N;

}

int main()

{ string N;

long long int M;

cin>>N>>M;

cout << maximizeNumber(N, M) << endl;

return 0;

}

**PROBLEM\_3**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int t;

cin>>t;

while(t>0)

{

int b,p,f,h,c,max=0;

cin>>b>>p>>f;

cin>>h>>c;

if(h>c)

{

if(b >(2\*p))

{ b=b-(2\*p);

max=h\*p;

if(b>0 && b>=(2\*f)){

max=max+(f\*c);

}else if(b>0 && b<(2\*f))

{

max=max+(b/2)\*c;

}

}

else{

max=(b/2)\*h;

}

}

else

{

if(b>(2\*f))

{

b=b-(2\*f);

max=c\*f;

if(b>0 && b>=(2\*p)){

max=max+(p\*h);

}else if(b>0 && b<(2\*p))

{

max=max+(b/2)\*h;

}

}

else{

max=(b/2)\*c;

}

}

cout<<max<<endl;

t--;

}

return 0;

}

**STRING**

#include <bits/stdc++.h>

using namespace std;

int stringSimilarity(string s)

{

long long int i,j,k;

int count=s.size();

for(i=1; i<s.length(); i++)

{

for(j=0,k=i; j<s.length(); j++)

{

if(s[k]==s[j] )

{

count++;

k++;

}

else

{

break;

}

}

}

return count;

}

int main()

{

int t;

cin >> t;

while(t--)

{

string s;

cin>>s;

int re=stringSimilarity(s);

cout<<re<<endl;

}

return 0;

}