Q1

#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]);

sort(t.begin() + i + 1, t.end());

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;

}

Q2

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int t;

cin>>t;

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

{

int b,p,f;

int x,y,z,ans;

cin>>b>>p>>f;

int h,c;

cin>>h>>c;

if(b==2)

cout<<std::max(h,c)<<endl;

else if(b<2)

cout<<0<<endl;

else{

x=std::max(p,f);

int w=std::min(p,f);

y=(p+f);

z=b/2;

if(y<z)

{

ans=(x\*std::max(h,c))+((y-x)\*std::min(h,c));

cout<<ans<<endl;

}

else if(b==p==f)

{

ans=z\*std::max(h,c);

cout<<ans<<endl;

}

else if(p==f && p>=z)

{

ans=z\*std::max(h,c);

cout<<ans<<endl;

}

else

{

ans=(std::min(p,f)\*std::max(h,c))+((z-w)\*std::min(h,c));

cout<<ans<<endl;

}

}

}

return 0;

}

Q3

(Java Header Files)

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

for (int tc = 0; tc < T; tc++) {

int A = sc.nextInt();

int B = sc.nextInt();

int X = sc.nextInt();

System.out.println(solve(A, B, X));

}

sc.close();

}

static int solve(int A, int B, int X) {

if (B >= 0) {

return powMod(A, B, X);

} else {

return powMod(inverseMod(A, X), -B, X);

}

}

static int powMod(int base, int exponent, int modulus) {

int result = 1;

while (exponent != 0) {

if ((exponent & 1) != 0) {

result = multiplyMod(result, base, modulus);

}

base = multiplyMod(base, base, modulus);

exponent >>= 1;

}

return result;

}

static int multiplyMod(int x, int y, int modulus) {

return (int) ((long) x \* y % modulus);

}

static int inverseMod(int x, int modulus) {

return BigInteger.valueOf(x).modInverse(BigInteger.valueOf(modulus)).intValue();

}

}

Q4

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main()

{

int t;

cin>>t;

while(t--)

{

string s;

long m=0,i=0,c=0,sum=0;

cin>>s;

for(long k=0;k<s.length();k++)

{

for(long j=0+m;j<s.length();j++)

{

//cout<<"j is"<<j<<endl;

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

{

c++;

i++;

//cout<<c<<endl;

}

else

break;

}

// sum=sum+c;

i=0;

m++;

}

cout<<c<<endl;

}

}