# 算法组第二次考核

完成题目数：30

账号：郭庆

# 二分：

## A - 银行贷款

### 思路：

二分

### 代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. double n,m,k,l,r;
4. bool pd(double x){
5. return (pow(1.0/(1.0+x),k)>=1-n/m\*x);
6. }
7. int main(){
8. cin>>n>>m>>k;
9. l=0;r=10;
10. while(r-l>=0.0001){
11. double mid=(l+r)/2;
12. if(pd(mid))r=mid;
13. else l=mid;
14. }
15. cout<<fixed<<setprecision(1)<<l\*100;
16. return 0;
17. }

## B - 地标访问

### 思路：

二分

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. int t,n;
16. vector<int>a;
17. bool check(int x){
18. for(int r=x;r<=n;r++){
19. int l=r-x+1;
20. if(a[r]<=0){
21. if(-a[l]<=t)re true;
22. }
23. if(a[l]>=0){
24. if(a[r]<=t)re true;
25. }
26. if(a[l]<=0&&a[r]>=0){
27. if(min(a[r],-a[l])+a[r]-a[l]<=t)re true;
28. }
29. }
30. return 0;
31. }
32. void go(){
33. cin >> t >> n;
34. a.assign(n+1,0);
35. for(int i = 1;i  <= n;i++){cin >> a[i];}
36. sort(a.begin()+1,a.end());
37. int l=-1,r=n+1;
38. while(l+1<r){
39. int mid=(l+r)>>1;
40. if(check(mid))l=mid;
41. else r=mid;
42. }
43. cout<<l<<endl;
45. }
46. signed main() {
47. ios::sync\_with\_stdio(false);
48. cin.tie(nullptr);cout.tie(nullptr);
49. int ti = 1;
50. while(ti--){
51. go();
52. }
53. return 0;
54. }

## D - 跳石头

### 思路：

二分，遍历统计，比较

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. vector<int>a;
16. int l,n,m;
17. bool check(int d) {
18. int p = 0, ans = 0;
19. for (int i = 1; i <= n+1; i++) {
20. if (a[i] - p < d) ans++;
21. else p = a[i];
22. }
23. if (ans <= m) return true;
24. else return false;
25. }
26. void go(){
27. cin>>l>>n>>m;
28. a.assign(n+2,0);
29. a[0]=0,a[n+1]=l;
30. for (int i = 1; i <= n; ++i) {
31. cin>>a[i];
32. }
33. int le=0,ri=l,mid=0;
34. while (le < ri) {
35. mid = (le + ri + 1) / 2;
36. if (check(mid)) le = mid;
37. else ri = mid - 1;
38. }
39. cout<<le;el;
40. }
41. signed main() {
42. ios::sync\_with\_stdio(false);
43. cin.tie(nullptr);cout.tie(nullptr);
44. int t=1;
45. while(t--){
46. go();
47. }
48. return 0;
49. }

## E - 木材加工

### 思路：

二分

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int n,k;cin>>n>>k;
17. vector<int>arr(n);
18. for (int i = 0; i < n; ++i) {
19. cin>>arr[i];
20. }
21. int l = 0,r = 100000001,ans = 0,mid = 0;
22. while (l < r-1){
23. int cnt=0;
24. mid = (l+r)/2;
25. for (int i = 0; i < n; ++i) {
26. cnt += arr[i]/mid;
27. }
29. if(cnt == k){
30. l = mid;
31. } else if(cnt > k){
32. l = mid;
33. } else{
34. r = mid;
35. }
36. }
37. cout<<l;el;
38. }
39. signed main() {
40. ios::sync\_with\_stdio(false);
41. cin.tie(nullptr);cout.tie(nullptr);
42. int t=1;
43. while(t--){
44. go();
45. }
46. return 0;
47. }

## M - 路标设置

### 思路：

二分

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. int len,n,k;
16. vector<int>arr;
17. bool check(int m){
18. int cnt=0;
19. for (int i = 1; i < n; ++i) {
20. cnt += (arr[i]-arr[i-1]-1)/m;
21. }
22. *//cout<<cnt;el;*
23. if(cnt <= k){re false;}
24. else re true;
25. }
26. void go(){
27. cin>>len>>n>>k;
28. arr.assign(n,0);
29. for (int i = 0; i < n; ++i) {
30. cin>>arr[i];
31. }
32. sort(all(arr));
33. *// if(arr.front()!=0){arr.insert(arr.begin(),0);}*
34. *// if(arr.back() != len){arr.push\_back(len);}*
35. *// sort(all(arr));*
36. int n = arr.size();
38. *// for (int i = 0; i < n; ++i) {*
39. *//  cout<<arr[i]<<' ';*
40. *// }el;*
41. *//*
42. int l=1,r=1e9,ans=0,mid=0;
43. while (l <= r){
44. *//cout<<l<<' '<<r<<' ';*
45. mid = (l+r)/2;
46. if(check(mid)){l = mid+1;}
47. else {
48. r =mid-1;
49. ans=mid;
50. }
51. }
52. cout<<l;el;
53. }
54. signed main() {
55. ios::sync\_with\_stdio(false);
56. cin.tie(nullptr);cout.tie(nullptr);
57. int t=1;
58. while(t--){
59. go();
60. }
61. return 0;
62. }

# 动态规划：

# 思维：

## A - 2D Traveling

### 思路：

暴力枚举

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int n,k,a,b;cin >>n>>k>>a>>b;
17. vector<pll>coor(n+1,{-1,-1});
18. for(int i = 1;i <= n;i++){cin >> coor[i].ff >> coor[i].ss;}
19. int ans = abs(coor[a].ff-coor[b].ff)+abs(coor[a].ss-coor[b].ss);
20. int ans1 = 1e18,ans2 = 1e18;
21. if(a<=k) ans1=0;
22. else {
23. for (int i = 1; i <= k; ++i) {
24. ans1 = min(ans1, abs(coor[i].ff - coor[a].ff) + abs(coor[i].ss - coor[a].ss));
25. }
26. }
27. if(b<=k) ans2=0;
28. else {
29. for (int i = 1; i <= k; ++i) {
30. ans2 = min(ans2, abs(coor[i].ff - coor[b].ff) + abs(coor[i].ss - coor[b].ss));
31. }
32. }
33. cout<<min(ans,ans1+ans2);el;
34. }
35. signed main() {
36. ios::sync\_with\_stdio(false);
37. cin.tie(nullptr);cout.tie(nullptr);
38. int t;cin >>t;
39. while(t--){
40. go();
41. }
42. return 0;
43. }

## C - Swap and Reverse

### 思路：

若k为偶数，则直接排序；否，则奇偶位分开排序并合并。

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int n,k;cin>>n>>k;
17. str s;cin>>s;
18. if(k%2){
19. str s1,s2;
20. for (int i = 0; i < n; ++i) {
21. if(i%2){s1 += s[i];}
22. else s2 += s[i];
23. }
24. sort(all(s1)), sort(all(s2));
25. for (int i = 0; i < n; ++i) {
26. if(i%2){cout<<s1[i/2];}
27. else cout<<s2[i/2];
28. }
29. } else{
30. sort(all(s));
31. cout<<s;
32. }
33. el;
34. }
35. signed main() {
36. ios::sync\_with\_stdio(false);
37. cin.tie(nullptr);cout.tie(nullptr);
38. int t;cin >>t;
39. while(t--){
40. go();
41. }
42. return 0;
43. }

## F - Two Binary Strings

### 思路：

要找到能否通过多次对a,b的操作使得a=b，只需找到是否有一个索引使得ai == bi == 0 and ai+1 == bi+1 == 1。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. string a,b;cin >> a >> b;
10. if(a == b){
11. cout << "YES" << endl;
12. return;
13. }
14. int len = a.length();
15. for(int i = 0;i < len-1;i++){
16. if(a[i] == '0' and a[i] == b[i] and a[i+1] == b[i+1] and a[i+1] == '1'){
17. cout << "YES" << endl;
18. return;
19. }
20. }
21. cout << "NO" << endl;
22. }
23. int main() {
24. ios::sync\_with\_stdio(false);
25. cin.tie(nullptr);cout.tie(nullptr);
26. int t;cin >>t;
27. while(t--){
28. go();
29. }
30. return 0;
31. }

## J - Fibonaccharsis

### 思路：

二分，注意斐波那契数列的特性

### 代码：

1. #include <iostream>
2. #include<string>
3. #include<algorithm>
4. #include <vector>
5. #include <cmath>
6. #include <set>
7. #include <queue>
8. using namespace std;
9. int main()
10. {
11. ios::sync\_with\_stdio(false);
12. cin.tie(0);
13. int T;
14. cin >> T;
15. while (T--) {
16. int n; int k;
17. cin >> n >> k;
18. int ans = 0;
19. for (int i = 1; i <= n; i++) {
20. int second = n;
21. int first = i;
22. bool valid\_seq = true;
23. for (int j = 0; j < k - 2; j++) {
24. int fx = first;
25. first = second - fx;
26. second = fx;
27. valid\_seq &= first <= second;
28. valid\_seq &= min(first, second) >= 0;
29. if (!valid\_seq) break;
30. }
31. if (valid\_seq) ans++;
32. }
33. cout << ans << endl;
34. }
35. }

## L - Cardboard for Pictures

### 思路：

二分

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int n,c;cin>>n>>c;
17. vector<int>a(n);
18. int pow\_sum = 0,sum = 0;
19. for (int i = 0; i < n; ++i) {
20. cin>>a[i];
21. }
22. int l = 1,r = 1e9,ans = 0,mid = 0;
23. while (l < r){
24. *//cout<<l<<' '<<r;el;*
25. mid = (l+r)/2;
26. int cal = 0;
27. for (int i = 0; i < n; ++i) {
28. cal += (int)pow(a[i]+2\*mid,2);
29. if(cal > c){
30. break;
31. }
32. }
33. ans = mid;
34. if(cal == c){
35. break;
36. }else if(cal < c){
37. l = mid+1;
38. } else{
39. r = mid;
40. }
41. }
42. cout<<ans;el;
43. }
44. signed main() {
45. ios::sync\_with\_stdio(false);
46. cin.tie(nullptr);cout.tie(nullptr);
47. int t;cin >>t;
48. while(t--){
49. go();
50. }
51. return 0;
52. }

# 数论：

## A - Humble Numbers

### 思路：

递归暴力打表

### 代码：

1. #include <iostream>
2. #include <map>
3. #include <set>
4. #include <vector>
5. #include <algorithm>
6. #include <string>
7. typedef long long int int64;
8. typedef double dd;
9. typedef std::string str;
10. #define el std::cout << std::endl
11. #define re return
12. #define ff first
13. #define ss second
14. #define all(x) x.begin(), x.end()
15. #define pb(x) push\_back(x)
16. #define sz(x) static\_cast<int>((x).size())
17. int prim[5] = {1, 2, 3, 5, 7};
18. std::map<int64, int> mp;
19. std::set<int64> s;
20. std::vector<int64> nums;
21. void go() {
22. int n;
23. while (1) {
24. std::cin >> n;
25. if (n == 0) {
26. break;
27. }
28. std::cout << "The " << n;
29. if (n % 10 == 1 && n % 100 != 11) {
30. std::cout << "st ";
31. } else if (n % 10 == 2 && n % 100 != 12) {
32. std::cout << "nd ";
33. } else if (n % 10 == 3 && n % 100 != 13) {
34. std::cout << "rd ";
35. } else {
36. std::cout << "th ";
37. }
38. std::cout << "humble number is " << nums[n - 1] << "." << std::endl;
39. }
40. }
41. void dfs(int j, int64 ji) {
42. s.insert(ji);
43. if (s.size() > 60000 || ji > static\_cast<int64>(3e10) || mp[ji] > 0) {
44. re;
45. }
46. mp[ji]++;
47. for (int i = 0; i < 5; ++i) {
48. dfs(j + 1, ji \* prim[i]);
49. }
50. }
51. int main() {
52. int t = 1;
53. s.insert(1);
54. dfs(0, 1);
55. for (std::set<int64>::iterator i = s.begin(); i != s.end(); ++i) {
56. nums.push\_back(\*i);
57. }
58. std::sort(all(nums));
60. while (t--) {
61. go();
62. }
63. return 0;
64. }

## G - Death to Binary?

### 思路：

略

### 代码：

1. #include <iostream>
2. #include <string>
3. #include <vector>
4. #include <algorithm>
5. using namespace std;
6. int fibonacciBaseToDecimal(const string& s) {
7. vector<int> fib;
8. fib.push\_back(1);
9. fib.push\_back(2);
10. while (fib.size() < s.length()) {
11. fib.push\_back(fib[fib.size() - 1] + fib[fib.size() - 2]);
12. }
13. int decimal = 0;
14. for (size\_t i = 0; i < s.length(); ++i) {
15. if (s[s.length() - 1 - i] == '1') {
16. decimal += fib[i];
17. }
18. }
19. return decimal;
20. }
21. string decimalToFibonacciBase(int n) {
22. if (n == 0) return "0";
23. vector<int> fib;
24. fib.push\_back(1);
25. fib.push\_back(2);
26. while (fib.back() <= n) {
27. fib.push\_back(fib[fib.size() - 1] + fib[fib.size() - 2]);
28. }
29. string result;
30. for (int i = fib.size() - 1; i >= 0; --i) {
31. if (n >= fib[i]) {
32. result += '1';
33. n -= fib[i];
34. } else {
35. result += '0';
36. }
37. }
38. size\_t firstNonZero = result.find\_first\_not\_of('0');
39. if (firstNonZero != string::npos) {
40. result = result.substr(firstNonZero);
41. } else {
42. result = "0";
43. }
44. return result;
45. }
46. string addFibonacciNumbers(const string& x, const string& y) {
47. int xDec = fibonacciBaseToDecimal(x);
48. int yDec = fibonacciBaseToDecimal(y);
49. int sumDec = xDec + yDec;
50. return decimalToFibonacciBase(sumDec);
51. }
52. void formatOutput(const string& x, const string& y, const string& result) {
53. string xCanon = decimalToFibonacciBase(fibonacciBaseToDecimal(x));
54. string yCanon = decimalToFibonacciBase(fibonacciBaseToDecimal(y));
55. string resultCanon = result;
57. size\_t maxLen = max(xCanon.length(), max(yCanon.length(), resultCanon.length()));
59. string xPadded = string(maxLen - xCanon.length()+2, ' ') + xCanon;
60. string yPadded = string(maxLen - yCanon.length()+2, ' ') + yCanon;
61. string resultPadded = string(maxLen - resultCanon.length(), ' ') + resultCanon;
63. cout << xPadded << endl;
64. cout << "+" << yPadded.substr(1) << endl;
65. cout << "  " << string(resultPadded.length(), '-') << endl;
66. cout << "  " << resultPadded << endl;
67. cout << endl;
68. }
69. int main() {
70. string x, y;
71. while (cin >> x >> y) {
72. string result = addFibonacciNumbers(x, y);
73. formatOutput(x, y, result);
74. }
75. return 0;
76. }

## I - Maximum GCD

### 思路：

暴力即可，注意输入处理

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int temp = 0;
17. vector<int>a;
18. str s;
19. getline(cin,s);
20. for (int i = 0; i < s.length(); ++i) {
21. if(s[i] != ' '){
22. temp = temp\*10+(s[i]-'0');
23. } else{
24. a.push\_back(temp);
25. temp = 0;
26. }
27. }
28. if(temp != 0){a.push\_back(temp);}
30. int ans = INT\_MIN;
31. for (int i = 0; i < a.size()-1; ++i) {
32. for (int j = i+1; j < a.size(); ++j) {
33. int cal = \_\_gcd(a[i], a[j]);
34. ans = max(cal, ans);
35. }
36. }
37. cout<<ans;el;
38. }
39. signed main() {
40. ios::sync\_with\_stdio(false);
41. cin.tie(nullptr);cout.tie(nullptr);
42. int t;cin >>t;
43. cin.ignore();
44. while(t--){
45. go();
46. }
47. return 0;
48. }

## N - Harmonic Number (II)

### 思路：

注意到，当 i 的值较大时，n / i 的值会变得相对较小，因此可以利用这个特性，计算一个范围内的连续相同值的累加和。

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. using namespace std;
4. void go(){
5. int n;
6. cin >> n;
7. int res = 0;
8. *// 处理小值部分*
9. for (int i = 1; i <= n / 2; i++) {
10. int val = n / i;
11. int next\_i = n / val + 1;
12. res += val \* (next\_i - i);
13. i = next\_i - 1;
14. }
15. res += n / 2 + (n % 2 == 1);
16. cout << res << endl;
17. }
18. signed main() {
19. ios::sync\_with\_stdio(false);
20. cin.tie(nullptr);
21. int t;
22. cin >> t;
23. int i = 1;
24. while (t--) {
25. cout << "Case " << i++ << ": ";
26. go();
27. }
28. return 0;
29. }

## P - 余数之和

### 思路：

枚举

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. int n,k;cin>>n>>k;
17. int ans=0;
18. for (int i = 1; i <= min(k,n); ++i) {
19. ans += k%i;
20. }
21. ans += max(n-k,0ll)\*k;
22. cout<<ans;el;
23. }
24. signed main() {
25. ios::sync\_with\_stdio(false);
26. cin.tie(nullptr);cout.tie(nullptr);
27. int t=1;
28. while(t--){
29. go();
30. }
31. return 0;
32. }

# 简单搜索：

## A - 迷宫

### 思路：

用dfs暴力搜索即可

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. int n,m,t,sx,sy,fx,fy,ans = 0;
15. set<pll>coor;
16. void dfs(int x,int y,vector<vector<bool>>vis){
17. if(coor.find({x,y}) != coor.end()){
18. re;
19. }
20. if(x < 1 or x > n or y < 1 or y > m){
21. re;
22. }
23. if(x == fx and y == fy){
24. ans++;
25. re;
26. }
27. if(vis[x][y]){re;}
28. vis[x][y] = true;
29. dfs(x+1,y,vis);
30. dfs(x-1,y,vis);
31. dfs(x,y+1,vis);
32. dfs(x,y-1,vis);
34. }
35. void go(){
36. cin>>n>>m>>t>>sx>>sy>>fx>>fy;
37. for (int i = 0; i < t; ++i) {
38. int x,y;cin>>x>>y;
39. coor.insert({x,y});
40. }
41. vector<vector<bool>>vis(n+1,vector<bool>(m+1, false));
42. dfs(sx,sy,vis);
43. cout<<ans;el;
44. }
45. signed main() {
46. ios::sync\_with\_stdio(false);
47. cin.tie(nullptr);cout.tie(nullptr);
48. go();
49. return 0;
50. }

## B - 马的遍历

### 思路：

bfs遍历即可

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<ll,ll>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. const vector<vector<int>>moving = { {1,2},{2,1},{-1,-2},{-2,-1},{1,-2},{-1,2},{2,-1},{-2,1} };
15. signed main() {
16. int n, m, x, y;
17. scanf("%d%d%d%d", &n, &m, &x, &y);
18. vector<vector<int>>coordinated(n + 1, vector<int>(m + 1, INT\_MAX));
19. queue<pair<int, int>> q;
20. q.push({ x, y });
21. int cx, cy;
22. coordinated[x][y] = 0;
23. while (!q.empty()) {
24. pair<int, int> top = q.front();
25. int cx = top.first;
26. int cy = top.second;
27. q.pop();
28. for (const auto& move : moving) {
29. int nx = cx + move[0];
30. int ny = cy + move[1];
31. if (nx > 0 && nx <= n && ny > 0 && ny <= m && (coordinated[nx][ny] == INT\_MAX)) {
32. coordinated[nx][ny] = coordinated[cx][cy] + 1;
33. q.push({ nx, ny });
34. }
35. }
36. }
37. if (coordinated[x][y] == INT\_MAX)coordinated[x][y] = 0;
38. for (int i = 1;i <= n;i++) {
39. for (int j = 1;j <= m;j++) {
40. if(coordinated[i][j] == INT\_MAX)
41. printf("-1  ");
42. else {
43. printf("%d  ", coordinated[i][j]);
44. }
45. }
46. printf("\n");
47. }
48. return 0;
49. }

## C - 八皇后 Checker Challenge

### 思路：

dfs

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. int n,ans=0;
15. int row[105],ls[105],rs[105],col[105];
16. void print()
17. {
18. if(ans<=2)
19. {
20. for(int i=1;i<=n;i++)
21. cout << col[i] << " ";
22. cout<<endl;
23. }
24. ans++;
25. }
26. void dfs(int k)
27. {
28. if(k>n)
29. {
30. print();
31. return ;
32. }
33. for(int i=1;i<=n;i++)
34. {
35. if(!row[i] && !ls[i + k] && !rs[i - k + n])
36. {
37. col[k]=i;
38. row[i]=1;
39. ls[k+i]=1;
40. rs[i-k+n]=1;
41. dfs(k+1);
42. row[i]=0;
43. ls[k+i]=0;
44. rs[i-k+n]=0;
45. }
46. }
47. }
48. void go(){
49. cin>>n;
50. dfs(1);
51. cout << ans;el;
52. }
53. signed main() {
54. ios::sync\_with\_stdio(false);
55. cin.tie(nullptr);cout.tie(nullptr);
56. int t = 1;
57. while(t--){
58. go();
59. }
60. return 0;
61. }

## D - Corn Maze S

### 思路：

bfs

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. int n,m,ans = INT\_MAX,sx = 0,sy = 0,ex = 0,ey = 0;
15. const int N=350;
16. char a[N][N];
17. bool vis[N][N];
18. struct coor{
19. int x;
20. int y;
21. int t;
22. };
23. queue<coor> q;
24. int dx[4]={1,0,-1,0};
25. int dy[4]={0,1,0,-1};
26. void portals(int &nx, int &ny, int k)
27. {
28. for(int i=1;i<=n;i++)
29. {
30. for(int j=1;j<=m;j++)
31. {
32. if(a[i][j]==a[nx][ny]&&(i!=nx||j!=ny))
33. {
34. nx=i,ny=j;
35. re;
36. }
37. }
38. }
39. }
40. void go(){
41. cin>>n>>m;
42. string s;
43. for(int i=1;i<=n;i++)
44. {
45. cin>>s;
46. for(int j=1;j<=m;j++)
47. {
48. a[i][j]=s[j-1];
49. if(a[i][j]=='@')
50. {
51. sx=i;
52. sy=j;
53. }
54. }
55. }
56. q.push((coor){sx, sy, 0});
58. while(!q.empty())
59. {
60. coor f=q.front();
61. q.pop();
62. if(a[f.x][f.y]=='=')
63. {
64. cout<<f.t;
65. re;
66. }
67. if(a[f.x][f.y]>='A'&&a[f.x][f.y]<='Z')
68. {
69. portals(f.x, f.y, f.t);
70. }
71. for(int i=0;i<=3;i++)
72. {
73. int nx=f.x+dx[i];
74. int ny=f.y+dy[i];
75. if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&a[nx][ny]!='#'&&!vis[nx][ny])
76. {
77. vis[nx][ny]=true;
78. q.push((coor){nx, ny, f.t + 1});
79. }
80. }
81. }
82. re;
83. }
84. signed main() {
85. ios::sync\_with\_stdio(false);
86. cin.tie(nullptr);cout.tie(nullptr);
87. int t = 1;
88. while(t--){
89. go();
90. }
91. return 0;
92. }

## E - 单词方阵

### 思路：

暴力

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. int n;str check = "yizhong";
16. vector<vector<char>>a,ans;
17. int MOVE[3] = {-1,0,1};
18. void finding(int x,int y,int xp,int yp){
19. bool find = true;int sx = x,sy = y,cx = x,cy = y;
20. for (int i = 0; i < 7; ++i) {
21. if(0<=cx and cx<n and 0<=cy and cy<n and a[cx][cy] == check[i]){
22. } else{
23. find = false;
24. break;
25. }
26. cx += xp,cy += yp;
27. }
28. if(find){
29. for (int i = 0; i < 7; ++i) {
30. ans[sx][sy] = a[sx][sy];
31. sx+=xp,sy+=yp;
32. }
33. }
34. }
35. void go(){
36. cin>>n;
37. a.assign(n,vector<char>(n,'\*'));
38. ans.assign(n,vector<char>(n,'\*'));
39. for (int i = 0; i < n; ++i) {
40. str s;cin >>s;
41. for (int j = 0; j < n; ++j) {
42. a[i][j] = s[j];
43. }
44. }
45. for (int i = 0; i < n; ++i) {
46. for (int j = 0; j < n; ++j) {
47. if(a[i][j] == 'y'){
48. for (int x = 0; x < 3; ++x) {
49. for (int y = 0; y < 3; ++y) {
50. if (x == 1 && y == 1) continue;
51. int nx = i+MOVE[x],ny = j+MOVE[y];
52. if(0<=nx and nx< n and 0<=ny and ny<n and a[nx][ny] == 'i'){
53. finding(i,j,MOVE[x],MOVE[y]);
54. }
55. }
56. }
57. }
58. }
59. }
60. for (int i = 0; i < n; ++i) {
61. for (int j = 0; j < n; ++j) {
62. cout<<ans[i][j];
63. }el;
64. }
65. }
66. signed main() {
67. ios::sync\_with\_stdio(false);
68. cin.tie(nullptr);cout.tie(nullptr);
69. int t=1;
70. while(t--){
71. go();
72. }
73. return 0;
74. }

# 字符串：

## A - 字符串替换

### 思路：

略

### 代码：

1. #include<iostream>
2. #include<string>
3. #define int long long
4. #define dd double
5. #define str string
6. #define el cout << endl
7. #define re return
8. #define ff first
9. #define ss second
10. #define pll pair<ll,ll>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. string s;
17. while(getline(cin,s)){
18. int len = s.length();
19. for(int i = 0;i <len;i++){
20. if(s[i]=='y' and s[i+1]=='o' and s[i+2]=='u'){
21. cout<<"we";
22. i+=2;
23. } else{
24. cout<<s[i];
25. }
26. }
27. el;
28. }
29. }
30. signed main() {
31. ios::sync\_with\_stdio(false);
32. cin.tie(nullptr);cout.tie(nullptr);
33. int t = 1;
34. while(t--){
35. go();
36. }
37. return 0;
38. }

## B - Way Too Long Words

### 思路：

略

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<ll,ll>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. void go(){
15. string s;cin>>s;
16. int l = s.length();
17. if(l<=10){
18. cout<<s;el;
19. } else{
20. cout<<s[0]<<l-2<<s[l-1];el;
21. }
22. }
23. signed main() {
24. ios::sync\_with\_stdio(false);
25. cin.tie(nullptr);cout.tie(nullptr);
26. int t;cin >>t;
27. while(t--){
28. go();
29. }
30. return 0;
31. }

## C - All in All

### 思路：

略

### 代码：

1. #include<iostream>
2. #include<string>
3. #include <algorithm>
4. #define int long long
5. #define dd double
6. #define str string
7. #define el cout << endl
8. #define re return
9. #define ff first
10. #define ss second
11. #define pll pair<ll,ll>
12. #define all(x) x.begin(),x.end()
13. #define pb(x) push\_back(x)
14. #define sz(x) (int) x.size()
15. using namespace std;
16. void go(){
17. str s,t;
18. while (cin>>s>>t){
19. int ls = s.length(),lt = t.length();
20. int i = 0,j = 0;
21. while (i<ls and j<lt){
22. if(s[i] == t[j]){
23. i++;
24. }
25. j++;
26. }
27. if(i==ls){
28. cout<<"Yes";
29. } else{
30. cout<<"No";
31. }el;
32. }
34. }
35. signed main() {
36. ios::sync\_with\_stdio(false);
37. cin.tie(nullptr);cout.tie(nullptr);
38. int t =1;
39. while(t--){
40. go();
41. }
42. return 0;
43. }

## D - A Tide of Riverscape

### 思路：

首先把所有‘.’都变为1，然后判断i−p与 i+p的点是否相同，相同的话就就换为 0，最后判断存不存在一组周期 p，若有，则输出No。

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<ll,ll>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. void go(){
15. int n,p;cin>>n>>p;
16. str s;cin>>s;
17. int len = s.length(),cnt = 0;
18. for (int i = 0; i < len; ++i) {
19. if(s[i]=='.')
20. {
21. s[i]='1';
22. if((i+p<n&&s[i+p]=='1')||(i-p>=0&&s[i-p]=='1')) s[i]='0';
23. }
24. if(i+p<n&&s[i]!=s[i+p]) cnt=1;
25. }
26. if(cnt){
27. cout<<s;el;
28. } else{
29. cout<<"No";el;
30. }
32. }
33. signed main() {
34. ios::sync\_with\_stdio(false);
35. cin.tie(nullptr);cout.tie(nullptr);
36. int t = 1;
37. while(t--){
38. go();
39. }
40. return 0;
41. }

## J - Bear and String Distance

### 思路：

略

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<ll,ll>
10. #define all(x) x.begin(),x.end()
11. #define pb(x) push\_back(x)
12. #define sz(x) (int) x.size()
13. using namespace std;
14. void go(){
15. int n,k;cin>>n>>k;
16. int c = k;
17. str s,ans;cin>>s;
18. for (int i = 0; i < n; ++i) {
19. if(k > 0){
20. int da = abs(s[i]-'a'),dz = abs('z'-s[i]);
21. int up = max(da,dz),down = min(da,dz);
22. if(k >= up){
23. ans += da>dz?'a':'z';
24. k-=up;
25. }else if(k <= down){
26. ans += (s[i]-k);
27. k -= k;
28. } else{
29. if(da>dz){
30. ans += s[i]-k;
31. k-=k;
32. } else{
33. ans+=s[i]+k;
34. k-=k;
35. }
36. }
37. } else{
38. ans+=s[i];
39. }
40. }
41. if(k>0){
42. cout<<-1;el;
43. } else{
44. cout<<ans;el;
45. }
46. }
47. signed main() {
48. ios::sync\_with\_stdio(false);
49. cin.tie(nullptr);cout.tie(nullptr);
50. int t = 1;
51. while(t--){
52. go();
53. }
54. return 0;
55. }

## R - Barnicle

### 思路：

分别对小数点、整数、小数进行特判即可

### 代码：

1. #include<bits/stdc++.h>
2. #define int long long
3. #define dd double
4. #define str string
5. #define el cout << endl
6. #define re return
7. #define ff first
8. #define ss second
9. #define pll pair<int,int>
10. #define pdd pair<dd,dd>
11. #define all(x) x.begin(),x.end()
12. #define pb(x) push\_back(x)
13. #define sz(x) (int) x.size()
14. using namespace std;
15. void go(){
16. str s;cin>>s;
17. int len = s.length();
18. str s1,s2;int cnt = 0;
19. bool fe = false,point = false;
20. for (int i = 0; i < len; ++i) {
21. if(s[i] == 'e'){
22. fe = true;
23. continue;
24. }
25. if(!fe){
26. if(s[i] == '.'){
27. point = true;continue;
28. }
29. if(point){
30. s2 += s[i];
31. } else{
32. s1 += s[i];
33. }
34. } else{
35. if(s[i] != '-'){
36. cnt = cnt\*10 + (int)(s[i]-'0');
37. }
38. }
39. *//cout<<"p3 ";*
40. }
41. int l2 = s2.length(),l1 = s1.length();
42. bool pp = (s2.length()>cnt);
43. *//cout<<"p1 ";*
44. while (!s2.empty() && s2.back() == '0') {
45. s2.pop\_back();
46. }
47. while (!s1.empty() && s1.front() == '0') {
48. s1.erase(s1.begin());
49. }
50. *//cout<<"p2 ";*
51. while (!s2.empty() and cnt > 0){
52. s1 += s2.front();
53. s2.erase(s2.begin());
54. cnt--;
55. }
56. if(s1.empty() ){cout<<"0";}
57. else cout<<s1;
58. if(!s2.empty()){cout<<".";}
59. cout<<s2;
60. for (int i = 0; i < cnt; ++i) {
61. cout<<"0";
62. }
63. el;
64. }
65. signed main() {
66. ios::sync\_with\_stdio(false);
67. cin.tie(nullptr);cout.tie(nullptr);
68. int t=1;
69. while(t--){
70. go();
71. }
72. return 0;
73. }

# 模拟构造：

## A - BerSU Ball

### 思路：

先排序，再上下双指针移动

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. int n;cin >> n;vector<int>ma(n);
10. for(int i = 0;i < n;i++){cin >> ma[i];}
11. int m;cin >> m;vector<int>fe(m);
12. for(int i = 0;i < m;i++){cin >> fe[i];}
13. sort(all(ma));
14. sort(all(fe));
15. int i = 0,j = 0,cnt = 0;
16. while (i < n and j < m){
17. if(abs(ma[i] - fe[j]) <= 1){
18. cnt++;
19. i++;
20. j++;
21. }else if(ma[i] > fe[j]){
22. j++;
23. }else{
24. i++;
25. }
26. }
27. cout << cnt <<endl;
28. }
29. int main() {
30. ios::sync\_with\_stdio(false);
31. cin.tie(nullptr);cout.tie(nullptr);
32. int t = 1;
33. while(t--){
34. go();
35. }
36. return 0;
37. }

## B - AB Flipping

### 思路：

由于每一个下标只能用一次，且只有si ==A 并且 si+1 ==B时才能互换，

因此只需找到第一次A出现的下标ia和B最后一次出现的下标ib,

L = ib - ia,答案为max(L,0)。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. int n;cin >>n;
10. string s;cin >> s;
11. int ia = INT\_MAX,ib = INT\_MIN;
12. for(int i = 0;i <n;i++){
13. if(s[i] == 'A'){
14. ia = min(ia,i);
15. } else{
16. ib = max(ib,i);
17. }
19. }
20. int ans = max(ib-ia,0);
21. cout <<ans << endl;
22. }
23. int main() {
24. ios::sync\_with\_stdio(false);
25. cin.tie(nullptr);cout.tie(nullptr);
26. int t;cin >>t;
27. while(t--){
28. go();
29. }
30. return 0;
31. }

## C - Two Divisors

### 思路：

题目给x中最大的两个因数a,b(a < b < x)，先分解公约数，反复计算a和b的最大公约数（gcd），并将它们除以该公约数，直到gcd为1。每次分解的公约数累积到结果ans中。将分解后的剩余部分相乘，并乘以累积的公约数，得到初步结果。当结果等于原输入b时，说明x可能是一个平方数，此时需要调整结果为该平方数。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. ll a,b;cin>> a>>b;
10. ll same = \_\_gcd(a,b);
11. ll ca = a,cb = b;
12. ll ans = 1;
13. while(same != 1){
14. a /= same;
15. b /= same;
16. ans \*= same;
17. same = \_\_gcd(a,b);
19. }
20. ans \*= a\*b;
21. if(ans == cb){
22. ans = ans\*ans/ca;
23. }
24. cout << ans << endl;
25. }
26. int main() {
27. ios::sync\_with\_stdio(false);
28. cin.tie(nullptr);cout.tie(nullptr);
29. int t;cin >>t;
30. while(t--){
31. go();
32. }
33. return 0;
34. }

## D - Two Out of Three

### 思路：

依题意，给一个数组a，问是否找到一个新数组b恰好只满足题中的要求三个中的两个。所以，只需检查a中是否有2组或以上的数字计数大于1个，如果有则可以找到，否，则无。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. int n;cin >> n;
10. vector<int>a(n,0),b(n,0);
11. bool same = true;
12. map<int,vector<int>>cnt;
13. for(int i = 0;i < n;i++){
14. cin >>a[i];
15. if(cnt[a[i]].empty()){
16. cnt[a[i]] = vector<int>(0);
17. }
18. cnt[a[i]].push\_back(i);
19. if(i and a[i] != a[i-1]){same = false;}
20. }
21. int c = 0;
22. for(auto& i:cnt){
23. if(i.second.size() >= 2){
24. c++;
25. if(c == 1){
26. for(int j = 0;j < i.second.size();j++){
27. b[i.second[j]] = 1;
28. }
29. b[i.second[0]] = 2;
30. }else if (c == 2){
31. for(int j = 0;j < i.second.size();j++){
32. b[i.second[j]] = 2;
33. }
34. b[i.second[0]] = 3;
35. }
37. }
38. }
39. if(c < 2){cout << -1 << endl;
40. return;
41. }
42. for(int i = 0;i < n;i++){
43. if(b[i] == 0){
44. b[i] = 1;
45. }
46. }
47. for(auto& i:b){cout << i <<' ';}
48. cout << endl;
49. }
50. int main() {
51. ios::sync\_with\_stdio(false);
52. cin.tie(nullptr);cout.tie(nullptr);
53. int t;cin >>t;
54. while(t--){
55. go();
56. }
57. return 0;
58. }

## F - Two Binary Strings

### 思路：

要找到能否通过多次对a,b的操作使得a=b，只需找到是否有一个索引使得ai == bi == 0 and ai+1 == bi+1 == 1。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. string a,b;cin >> a >> b;
10. if(a == b){
11. cout << "YES" << endl;
12. return;
13. }
14. int len = a.length();
15. for(int i = 0;i < len-1;i++){
16. if(a[i] == '0' and a[i] == b[i] and a[i+1] == b[i+1] and a[i+1] == '1'){
17. cout << "YES" << endl;
18. return;
19. }
20. }
21. cout << "NO" << endl;
22. }
23. int main() {
24. ios::sync\_with\_stdio(false);
25. cin.tie(nullptr);cout.tie(nullptr);
26. int t;cin >>t;
27. while(t--){
28. go();
29. }
30. return 0;
31. }

## G - Arranging Cats

### 思路：

依题意，通过题目的要求操作经过n次使a==b，求最少操作数，因此，要使操作尽可能少，则尽可能多的执行操作3，尽可能少的执行操作1和2。所以只需统计ai,bi不相等使ai == 1 的个数c1 和 ai == 0的个数c0，那么答案是max(c1,c0)。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define all(x) x.begin(),x.end()
5. #define pb(x) push\_back(x)
6. #define sz(x) (int) x.size()
7. using namespace std;
8. void go(){
9. int n; cin >> n;
10. string a,b;cin >> a >> b;
11. int c0 = 0,c1 = 0;
12. for(int i = 0 ;i <n;i++){
13. if(a[i] != b[i]){
14. if(a[i] == '0'){
15. c0++;
16. } else{
17. c1++;
18. }
19. }
20. }
21. int ans = max(c0,c1);
22. cout << ans << endl;
24. }
25. int main() {
26. ios::sync\_with\_stdio(false);
27. cin.tie(nullptr);cout.tie(nullptr);
28. int t;cin >>t;
29. while(t--){
30. go();
31. }
32. return 0;
33. }

## H - Maximum Sum

### 思路：

前缀和。依题意，在k次操作中，要么去掉最小两个数，要么去掉最大的一个数。因此，统计前缀和，最多去除2\*k个最小值，或者去除最多k个最大值。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define el endl
5. #define all(x) x.begin(),x.end()
6. #define pb(x) push\_back(x)
7. #define sz(x) (int) x.size()
8. using namespace std;
9. void go(){
10. int n,k;cin >> n >> k;
11. vector<ll>arr(n);
12. for(int i = 0;i < n;i++){cin >> arr[i];}
13. ll sum = 0,cnt = 0,l = 0,r = n-1;
14. sort(all(arr));
15. vector<ll>front\_sum(n+1,0);
16. *//cout << "mid" <<el;*
17. for(int i = 0;i <= n;i++){
18. front\_sum[i]  = front\_sum[i-1]+arr[i-1];
19. }
20. ll ans = INT\_MIN;
21. for(int i = 0;i <= k;i++){
22. ans = max(ans,front\_sum[n-(k-i)]-front\_sum[i\*2]);
23. }
25. cout <<ans << endl;
26. }
27. int main() {
28. ios::sync\_with\_stdio(false);
29. cin.tie(nullptr);cout.tie(nullptr);
30. int t;cin >>t;
31. while(t--){
32. go();
33. }
34. return 0;
35. }

## I - Unnatural Language Processing

### 思路：

先转CV串，当si == V and si+2 == V时，该原字母加‘.’，而如果si == V and si+2 == C，该原字母的下一个字母加‘.’。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define el endl
5. #define ff first
6. #define ss second
7. #define pll pair<ll,ll>
8. #define all(x) x.begin(),x.end()
9. #define pb(x) push\_back(x)
10. #define sz(x) (int) x.size()
11. using namespace std;
12. void go(){
13. int n;cin >> n;
14. string s;cin >> s;
15. string c;
16. for(int i = 0;i < n;i++){
17. if(s[i] == 'a' or s[i] == 'e'){
18. c += 'V';
19. } else{
20. c += 'C';
21. }
22. }
23. c += "CV";
24. *//cout << c << el;*
25. string ans;
26. for(int i = 0;i < n;i++){
27. ans += s[i];
28. if(c[i] == 'V' and c[i+2] == 'V'){
29. ans += ".";
30. } else if(c[i] == 'V' and c[i+2] == 'C'){
31. ans += s[i+1];
32. ans += ".";
33. i++;
34. }
35. }
36. for(int i = 0;i < ans.length()-1;i++){
37. cout << ans[i];
38. }
39. cout << endl;
40. }
41. int main() {
42. ios::sync\_with\_stdio(false);
43. cin.tie(nullptr);cout.tie(nullptr);
44. int t;cin >>t;
45. while(t--){
46. go();
47. }
48. return 0;
49. }

## J - Cheap Travel

### 思路：

纯数学计算。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define el << endl
5. #define ff first
6. #define ss second
7. #define pll pair<ll,ll>
8. #define all(x) x.begin(),x.end()
9. #define pb(x) push\_back(x)
10. #define sz(x) (int) x.size()
11. using namespace std;
12. void go(){
13. int n,m,a,b;cin >> n>>m>>a>>b;
14. int total = 0;
15. if(m\*a <= b){
16. total += n\*a;
17. } else{
18. total += n/m\*b + min((n%m)\*a,b);
19. }
20. cout << total el;
21. }
22. int main() {
23. ios::sync\_with\_stdio(false);
24. cin.tie(nullptr);cout.tie(nullptr);
25. int t = 1;
26. while(t--){
27. go();
28. }
29. return 0;
30. }

## M - New Theatre Square

### 思路：

遍历统计计算。

### 代码：

1. #include<bits/stdc++.h>
2. #define ll long long
3. #define dd double
4. #define el << endl
5. #define ff first
6. #define ss second
7. #define pll pair<ll,ll>
8. #define all(x) x.begin(),x.end()
9. #define pb(x) push\_back(x)
10. #define sz(x) (int) x.size()
11. using namespace std;
12. int n,m,x,y;
13. int pri(string s){
14. int ans = 0;
15. for(int j = 0;j < m;j++){
16. if(s[j] == '.'){
17. if(s[j+1] == '.'){
18. ans += min(2\*x,y);
19. j++;
20. } else{
21. ans += x;
22. }
23. }
24. }
25. return ans;
26. }
27. void go(){
28. cin >>n>>m>>x>>y;
29. int sum = 0;
30. while (n--){
31. string s;cin >>s;
32. sum += pri(s);
33. }
34. cout << sum el;
35. }
36. int main() {
37. ios::sync\_with\_stdio(false);
38. cin.tie(nullptr);cout.tie(nullptr);
39. int t;cin >>t;
40. while(t--){
41. go();
42. }
43. return 0;
44. }

# 并查集：