Given a string XX formed out of single digit numbers from 0−90−9 , you are given a set of digits SS and you need to count total substring of string XX that contains all the digits in the set SS.   
**Input**   
First line contains a string as input. Next line contains a number nn as input denoting size of set SS. Next line contains nn space separated integers that denote the distinct integers in the set SS.   
**Output**  
In the output you have to give total count of substrings of the string XX such that they contain all the digits in the set SS  
**Constraints**  
1≤|X|≤1051≤|X|≤105  
1≤n≤101≤n≤10

Sample 1

333

1

3

Output

6

Sample 2

454545

2

4 5

Output

15

Sample 3

3332212

2

3 2

Output

12

Solution

1. *#include* <iostream>
2. *#include* <bits/stdc++.h>
3. *#define* li long int
5. using namespace std;
7. int main()
8. {
9. char str[100001];
10. li a,b,c,i,j,k,n,l,count=0,ans=0,vis[10];
11. cin>>str;
12. l=strlen(str);
13. cin>>n;
14. int arr[n];
15. for(i=0;i<n;i++)
16. cin>>arr[i];
17. for(i=0;i<l-n+1;i++)
18. {
19. count=0;
20. for(j=0;j<9;j++)
21. vis[j]=0;
22. for(j=0;j<n;j++)
23. {
24. vis[arr[j]]=1;
25. }
26. a=i;
27. for(a=i;a<l;a++)
28. {
29. b=str[a]-48;
31. if(vis[b]==1)
32. { vis[b]=0;
33. count++;
34. }
36. if(count==n)
37. break;
39. }
40. if(count==n)
41. ans+=(l-a);


45. }
46. cout<<ans;

49. return 0;
50. }

**Language:**C++

1. st=raw\_input()
2. n=int(raw\_input())
3. num=map(int,raw\_input().split())
4. l,ans=len(st),0
5. for i in xrange(l+1):
6. for j in xrange(i,l+1):
7. if len(st[i:j])>=len(num):
8. cnt=0
9. for k in num:
10. if not str(k) in st[i:j]:
11. break
12. else:
13. cnt+=1
14. if cnt!=len(num):
15. continue
16. else:
17. ans+=len(st)-j+1
18. break
19. print ans

second solution:-

1. *#include*<bits/stdc++.h>
2. using namespace std;
3. int main(){
4. string s;
5. cin>>s;
6. long n;
7. cin>>n;
8. vector <long> char1(10,0);
9. for(long i=0;i<n;i++){
10. long xx;
11. cin>>xx;
12. char1[xx]=1;
13. }
14. long result=0;
15. for(long i=0;i<s.size();i++){
16. long count=0;
17. vector<long> c2;
18. c2=char1;
19. for(long j=i;j<s.size();j++){
20. if(c2[s[j]-48]==1){
21. count++;
22. c2[s[j]-48]=0;
23. }
24. if(count==n){
25. result+=s.size()-j;
26. break;
27. }
29. }
31. }
32. cout<<result<<endl;
33. return 0;
34. }

Solution 3:

1. *#include* <stdio.h>
2. char a[100005];
3. int cnt[100005][15];
4. int dig[15];
5. int main()
6. {
7. scanf("%s",&a);
8. int i,n,j;
9. for(n=0;a[n];n++)
10. {
11. for(i=0;i<10;i++)cnt[n+1][i] = cnt[n][i];
12. j = a[n] - '0';
13. cnt[n+1][j]++;
14. }
15. int len = n;
16. scanf("%d",&n);
17. for(i=1;i<=n;i++)scanf("%d",&dig[i]);
18. long long ans = 0;
19. for(i=1;i<=len;i++)
20. {
21. int lo = i;
22. int hi = len;
23. for(j=1;j<=n;j++)
24. if(cnt[n][dig[j]] == cnt[i-1][dig[j]])break;
25. *// there is sum subarray*
26. *// printf("searching \n");*
27. int mid;
28. while(hi > lo+1)
29. {
30. mid = (lo + hi)/2;
31. int valid = 1;
32. for(j=1;j<=n;j++)
33. if(cnt[mid][dig[j]] == cnt[i-1][dig[j]]){
34. valid = 0; break;
35. }
36. if(valid)hi = mid;
37. else lo = mid;
38. }
39. for(mid=lo;mid<=hi;mid++)
40. {
41. int valid = 1;
42. for(j=1;j<=n;j++)
43. if(cnt[mid][dig[j]] == cnt[i-1][dig[j]]){
44. valid = 0; break;
45. }
46. if(valid)break;
47. }
48. ans += len + 1 - mid;
49. *// printf("from %d to minimum is %d\n",i,mid);*
50. }
51. printf("%lld\n",ans);
52. return 0;
53. }

Solution :4

1. *#include* <bits/stdc++.h>
2. using namespace std;
4. *#define* fill(x, y) memset(x, y, sizeof(x))
5. *#define* sci(x) int x; scanf("%d", &x)
6. *#define* sl(n) scanf("%lld", &n)
7. *#define* rep(i, x, y) for (\_\_typeof(x) i = x; i <= y; i ++)
8. *#define* repi(i, x, y) for (\_\_typeof(x) i = x; i >= y; i --)
9. typedef long long ll;
10. *#define* inf (1<<30)
11. *#define* eps 1e-9
12. *#define* mod 1e9 + 7
13. *#define* mp make\_pair
14. *#define* pb push\_back
15. *#define* fi first
16. *#define* se second
17. *#define* sz(x) (int)((x).size())
18. typedef pair<int, int> pii;
19. typedef vector<int> vi;
20. typedef vector<pii> vii;
21. *#define* fore(itr, x) for (\_\_typeof(x.begin()) itr = x.begin(); itr != x.end(); itr ++)
22. *#define* forei(itr, x) for (\_\_typeof(x.end()) itr = x.end() - 1; itr != x.begin() - 1; itr --)
23. *#define* all(x) (x).begin(), (x).end()
24. *#define* eprintf(...) fprintf(stderr, \_\_VA\_ARGS\_\_)
25. *#define* debug(x) { cerr << *#x << " = " << x << endl; }*
26. *#define* show(a, n) {cerr << *#a << " => ";for(int i=0;i<n;i++) cerr <<a[i]<<" \n"[i == (n-1)];}*
27. *#define* debugp(x) { cerr << *#x << " => " << x.first << " " << x.second << endl; }*
28. void tick(){static clock\_t oldt;clock\_t newt=clock();double diff=1.0\*(newt-oldt)/CLOCKS\_PER\_SEC;oldt=newt;debug(diff);}
29. int main(int argc, char const \*argv[] ) {
30. ios\_base::sync\_with\_stdio(0); *//will not work with scanf*
31. string s;
32. int ns, num, left=0, right=0, freq[10] = {0}, i, missing=0;
33. long long int res=0;
34. cin >> s;
35. cin >> ns;
36. int strlen = s.size();
37. rep(i,0,ns-1) {
38. cin >> num;
39. freq[num]++;
40. }
41. while (left < strlen) {
43. i = 0;
44. while (i <= 9 && right < strlen) {
45. *// debug(i);*
46. *// debug(right);*
47. if (freq[i] == 1) {
48. num = s[right]-'0';
49. if (freq[num] > 0){freq[num]++;}
50. if (num == i)i++;
51. right++;
52. } else {
53. *//not req or > freq*
54. i++;
55. }
56. *// show(freq, 10);*
57. }
58. *// show(freq, 10);*
59. while(i<=9)if (freq[i++]==1){missing=1;}
60. *// cerr << i << " " << left << " " << right << " " << s.substr(left, right-left) << endl;*
61. *//if rigth = strlen*
62. if(!missing)res += strlen-right+1;
63. num = s[left]-'0';
64. if (freq[num] > 0)freq[num]--;
65. left++;
66. }
67. cout << res << endl;
69. return 0;
70. }

Solution 6

1. *#include*<bits/stdc++.h>
2. using namespace std;
3. int check(int A[],int B[])
4. {
5. int i;
6. for(i=0;i<10;i++)
7. if(A[i]>B[i])
8. return 0;
9. return 1;
10. }
11. void print(string s,int a,int b)
12. {
13. int i;
14. printf("-- %d -- %d --",a,b);
15. for(i=0;i<a;i++) printf("\*");
16. for(i=a;i<=b;i++) printf("%c",s[i]);
17. for(i=b+1;i<s.size();i++) printf("\*");
18. printf("\n");
19. }
20. int main()
21. {
22. string s; cin>>s;
23. int n,i,j=0; cin>>n;
24. int A[n];
25. long long int ans=0;
26. for(i=0;i<n;i++) cin>>A[i];
27. int C[10]={0},B[10]={0};
28. for(i=0;i<n;i++) C[A[i]]++;
29. for(i=0;i<s.size();i++)
30. {
31. B[s[i]-'0']++;
32. if(check(C,B))
33. {
34. ans+=(s.size()-i);
35. while(j<=i)
36. {
37. B[s[j]-'0']--;
38. j++;
39. if(check(C,B)) ans+=(s.size()-i);
40. else break;
41. }
42. }
43. *//cout<<ans<<" "<<i<<endl;*
44. }
45. *//if(check(C,B)) ans++;*
46. cout<<ans<<endl;
47. }

Sol:7

1. *#include* <bits/stdc++.h>
2. *#define* ll long long
3. *#define* pb push\_back
4. using namespace std;
5. const int N = 2e5;
6. int a[N],kt[20],d[20];
7. string s;
8. int n;
9. void nhap()
10. {
11. getline(cin,s);
12. for (int i=1; i<=s.size(); i++) {a[i] = (int)s[i-1] - '0';}
13. scanf("%d\n",&n);
14. for (int i=1; i<=n; i++)
15. {
16. int x;
17. scanf("%d",&x);
18. kt[x] = 1;
19. }
20. }
21. void solve()
22. {
23. int l = 1;
24. int sl = 0;
25. long long res = 0;
26. for (int r = 1; r<=s.size(); r++)
27. {
28. int gt = a[r];
29. if (kt[a[r]]==1 && d[gt]==0) {sl++;}
30. if (kt[a[r]]==0)
31. {
32. if (sl==n) {res+=(long long)l;}
33. continue;
34. }
35. if (kt[a[r]]==1) {d[gt]++;}
36. if (sl==n)
37. {

while ((d[a[l]] > 1 && kt[a[l]]==1) || kt[a[l]]==0)

1. {
2. if (kt[a[l]]==1) {d[a[l]]--;}
3. l++;
4. }
5. res+=(long long)l;
6. }
7. }
8. cout<<res;
9. }
10. int main()
11. {
12. nhap();
13. solve();
14. }
15. import java.util.Scanner;
16. class stringmatchh2 {
17. public static void main(String args[]) throws Exception {
18. long count = 0;
19. Scanner sc = new Scanner(System.in);
20. int i, n;
21. String x;
22. x = sc.next();
23. n = sc.nextInt();
24. char s[] = new char[n];
25. for (i = 0; i < n; i++) {
26. s[i] = sc.next().charAt(0);
27. }
28. int start = -1;
29. int min;
30. int max = 0;
31. int a = 1;
32. int k = -1;
33. if (n != 1) {
34. while (a > 0) {
35. min = x.length() - 1;
36. for (i = 0; i < n; i++) {
37. int y = x.indexOf(s[i], start + 1);
38. if (y <= min)
39. min = y;
40. }
41. start = min;
42. if (min != -1) {
43. for (i = 0; i < n; i++) {
44. if ((x.indexOf(s[i], start)) < 0) {
45. a = -1;
46. break;
47. }
48. int y = 0;
49. if (s[i] != x.charAt(start))
50. y = x.indexOf(s[i], start + 1);
51. if (y > max)
52. max = y;
53. }
54. if (a == -1)
55. break;
56. count += (min - k) \* (x.length() - max);
57. k = start;
58. } else
59. break;
60. }
61. System.out.println(count);
62. }
63. else
64. {
65. while (a > 0) {
66. min = x.length() - 1;
67. for (i = 0; i < n; i++) {
68. int y = x.indexOf(s[i], start + 1) ;
69. if (y <= min)
70. min = y;
71. }
72. start = min;
73. if (min != -1) {
74. count += (min - k) \* (x.length() - start);
75. k = start;
76. }
77. else
78. break;
79. }
80. System.out.println(count);
81. }
82. }
83. }