

[← Practice Programming Problems / Roy and Birthday](#)

Roy and Birthday

Submissions
 Attempted by: 185 | Solved by: 50 | Partially Solved by: 103 | ★★★★★

Combinatorics
 Easy
 Implementation
 Math
 Edit

[Problem](#)
[Editorial](#)
[My Submissions](#)
[Analytics](#)

June Easy Challenge

We arrange consonants first, and we will put vowels in the empty spaces between them. Unique permutations for N elements are $N! / (x! y! z! \dots)$ where x, y, z ... are the counts of same type of elements.

If there are n consonants and m vowel we have $\text{choose}(n+1, m)$ ways to choose m points. Answer is -1 if $n+1 < m$ ie. vowels are more than the spaces available.

Also, we multiply this with unique permutations of vowels and consonants.

[Edit Editorial](#)

IS THIS EDITORIAL HELPFUL?



Yes, it's helpful



No, it's not helpful

1 developer(s) found this editorial helpful.

Author Solution by [Ravi Ojha](#)

```

1. #include<stdio>
2. #include<cstring>
3. #include<iostream>
4. #define MOD 1000000007
5. using namespace std;
6. long long f[1000011];
7. void fact()
8. {
9.     f[0] = 1;
10.    int i = 1;
11.    while(i<1000011)
12.    {
13.        f[i] = f[i-1]*i;
14.        f[i] = f[i]%MOD;
15.        i++;

```

```
16.     }
17. }
18. long long fast_pow(long long a, long long n)
19. {
20.     long long result = 1;
21.     long long power = n;
22.     long long value = a;
23.     while(power > 0)
24.     {
25.         if(power & 1)
26.         {
27.             result = result * value;
28.             result = result % MOD;
29.         }
30.         value = value * value;
31.         value = value % MOD;
32.         power /= 2;
33.     }
34.     return result;
35. }
36. long long inv(long long x)
37. {
38.     return fast_pow(x, MOD - 2);
39. }
40. int main()
41. {
42.     fact();
43.     int t;
44.     scanf("%d", &t);
45.     while(t--)
46.     {
47.         char goku[1000011];
48.         scanf("%s", goku);
49.         long long v_count = 0, c_count = 0;
50.         int freq[30] = {0};
51.         int len = strlen(goku);
52.         int i;
53.         for(i = 0; i < len; i++)
54.         {
55.             freq[goku[i] - 'a']++;
56.             if(goku[i] == 'a' || goku[i] == 'e' || goku[i] == 'i'
57.             {
58.                 v_count++;
59.             }
60.             else
61.             {
62.                 c_count++;
63.             }
64.         }
65.         int possible = c_count + 1 - v_count;
66.         if(possible < 0)
67.         {
68.             printf("-1\n");
69.         }
```

```

70.         else
71.         {
72.             long long c_res = f[c_count];
73.             long long v_res = f[c_count+1]*inv(f[possible]);
74.             v_res %= MOD;
75.             for(i=0;i<26;i++)
76.             {
77.                 int tmp = i+'a';
78.                 if(tmp == 'a' || tmp == 'e' || tmp == 'i' || tmp
79.                 {
80.                     v_res = v_res*inv(f[freq[i]]);
81.                     v_res %= MOD;
82.                 }
83.                 else
84.                 {
85.                     c_res = c_res*inv(f[freq[i]]);
86.                     c_res %= MOD;
87.                 }
88.             }
89.             //printf("%lld %lld\n",v_res, c_res);
90.             printf("%lld\n",(v_res*c_res)%MOD);
91.         }
92.     }
93.     return 0;
94. }

```

Tester Solution by Lalit Kundu

```

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define pb push_back
4. #define mp make_pair
5. #define clr(x) x.clear()
6. #define sz(x) ((int)(x).size())
7. #define F first
8. #define S second
9. #define REP(i,a,b) for(i=a;i<b;i++)
10. #define rep(i,b) for(i=0;i<b;i++)
11. #define repl(i,b) for(i=1;i<=b;i++)
12. #define pdn(n) printf("%d\n",n)
13. #define sl(n) scanf("%lld",&n)
14. #define sd(n) scanf("%d",&n)
15. #define pn printf("\n")
16. typedef pair<int,int> PII;
17. typedef vector<PII> VPII;
18. typedef vector<int> VI;
19. typedef vector<VI> VVI;
20. typedef long long LL;
21. #define MOD 1000000007
22. LL mpow(LL a, LL n)
23. {LL ret=1;LL b=a;while(n) {if(n&1)
24.     ret=(ret*b)%MOD;b=(b*b)%MOD;n>>=1;}

```

```

25. return (LL)ret;}
26. #define MAXN 1000100
27. #define assn(n,a,b) assert(n<=b && n>=a)
28. LL fac[MAXN+10]={},inv[MAXN+10]={};
29. void precalc()
30. {
31.     LL i;
32.     fac[0]=1;
33.     inv[0]=1;
34.     fac[1]=1;
35.     inv[1]=1;
36.     for(i=2; i<MAXN; i++)
37.     {
38.         fac[i]=(i*fac[i-1])%MOD;
39.         inv[i]=mpow(fac[i],MOD-2);
40.     }
41. }
42. LL choose(LL n, LL r)
43. {
44.     if(n<=0 || r<0 || n<r) return 0ll;
45.     LL p=((fac[n]*(inv[r]))%MOD*(inv[n-r]))%MOD;
46.     return p;
47. }
48. int main()
49. {
50.     precalc();
51.     int t,cnt=0;
52.     cin >> t;
53.     assn(t,1,100);
54.     while(t--)
55.     {
56.         string s;
57.         LL i,j,n,a=0,b=0,ar[5]={},arr[26]={},ss=0,sss=0;
58.         cin >> s;
59.         n=s.length();
60.         cnt+=n;
61.         assn(n,1,1000000);
62.         for(i=0; i<n; i++)
63.         {
64.             if(s[i]=='a')ar[0]++;
65.             else if(s[i]=='e')ar[1]++;
66.             else if(s[i]=='i')ar[2]++;
67.             else if(s[i]=='o')ar[3]++;
68.             else if(s[i]=='u')ar[4]++;
69.             else{arr[s[i]-'a']++;ss++;}
70.         }
71.         sss=ar[0]+ar[1]+ar[2]+ar[3]+ar[4];
72.         LL p=choose(ss+1,sss);
73.         if(ss+1<sss)
74.         {
75.             cout << -1 << endl;
76.             continue;
77.         }
78.         LL q=fac[sss];

```

```
79.         for(i=0; i<5; i++)
80.             q=(q*inv[ar[i]])%MOD;
81.         q=(q*fac[ss])%MOD;
82.         for(i=0; i<26; i++)
83.             q=(q*inv[arr[i]])%MOD;
84.         p=(p*q)%MOD;
85.         cout << p << endl;
86.     }
87.     assn(cnt,1,1000000);
88.     return 0;
89. }
```

PROFILE IMPACT

[Complete Profile](#)

*Excellent profile will increase your profile discoverability and keep you on top among others.

PROBLEMS SUGGESTED FOR YOU

[HackerEarth Tour](#)

Solved by 68

[The Xperiment](#)

Solved by 689

[ALS Ice Bucket Challenge](#)

Solved by 16

[more...](#)

RECENT SUBMISSIONS

User	Result	Time	Lang
Priyansh...		1.006	C++
Abhik Pa...		1.0059	C
Abhik Pa...		1.0058	C
Abhik Pa...		1.0058	C
Sayan Na...		1.006	C++
Sayan Na...		1.006	C++
Sayan Na...		1.0062	C++
View All			

TRENDING NOTES

[Number Theory - III](#)

written by Boris Sokolov

[Exact String Matching Algorithms](#)

written by Alei Reyes

[Binary Indexed Tree or Fenwick Tree](#)

written by Chandan Mittal

[Small tricks in for loop](#)

written by Rangeesh

[Strings And String Functions](#)

written by Vinay Singh

[more ...](#)

DEVELOPERS TO FOLLOW



[Abhimanyu Kumar](#)

0 followers



[Trung Nguyen](#)

18 followers



[Shaumik Daityari](#)

30 followers

COMPANIES TO FOLLOW

[Practo](#)

5293 followers

[Gigstart](#)

1717 followers

[Snapdeal](#)

5355 followers

RECOMMENDED CHALLENGES

[Horlicks Hack 4 Fun](#)

03 Sep 2015, 09:00 PM IST

[Register](#)[CODE-HUNT-2F](#)

21 Oct 2015, 05:00 PM IST

[Register](#)[Zoomcar Ruby Challenge](#)

23 Oct 2015, 06:00 PM IST

[Register](#)[Zomato Hiring Challenge](#)

23 Oct 2015, 06:00 PM IST

[Register](#)[Diona iOS Developer Hiring Challenge](#)

24 Oct 2015, 12:00 PM IST

[Register](#)[Tipstat Android Developer Hiring Challenge](#)

24 Oct 2015, 12:00 PM IST

[Register](#)[D'code](#)

SUBSCRIBE TO HACKEREARTH NEWS

[Subscribe](#)

JOIN PROGRAMMING CLUB ON FACEBOOK

[Join now](#)

ABOUT US

[Blog](#)
[Engineering Blog](#)
[Updates & Releases](#)
[Team](#)
[Careers](#)
[In the Press](#)

HACKEREARTH

[API](#)
[Chrome Extension](#)
[CodeTable](#)
[HackerEarth Academy](#)
[Developer Profile](#)
[Resume](#)
[Campus Ambassadors](#)
[Get Me Hired](#)
[Privacy](#)
[Terms of Service](#)

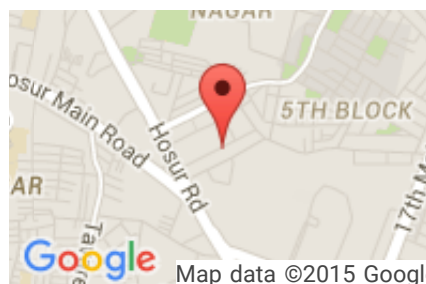
DEVELOPERS

[AMA](#)
[Code Monk](#)
[Judge Environment](#)
[Solution Guide](#)
[Problem Setter Guide](#)
[Practice Problems](#)
[HackerEarth Challenges](#)
[College Challenges](#)

RECRUIT

[Developer Sourcing](#)
[Lateral Hiring](#)
[Campus Hiring](#)
[FAQs](#)
[Customers](#)
[Annual Report](#)

REACH US



Map data ©2015 Google
IIIrd Floor, Salarpuria Business Center,
4th B Cross Road, 5th A Block,
Koramangala Industrial Layout,
Bangalore, Karnataka 560095, India.

✉ contact@hackerearth.com

☎ +91-80-4155-4695

☎ +1-650-461-4192

