专题7

目录

1001

AC代码

```
#include<bits/stdc++.h>
using namespace std;

int flag;
struct tr {
    int num;
    tr* ch[2];
    tr()
    {
        for (int i = 0; i < 2; i++) {
            ch[i] = 0;
        }
        num = 0;
    }
}*root;

void insert(string s)
{
    if (flag)return;
    tr* p = root;
    for (int i = 0; s[i] != '\0'; i++) {
        int temp = s[i] - '0';
        if (!p->ch[temp]) {
```

```
            p->ch[temp] = new tr;
        }
        p = p->ch[temp];
        if (p->num == 2) {
            flag = 1;
            return;
        }
        p->num = 1;
    }
    p->num = 2;
}
void Free(tr* p)
{
    for (int i = 0; i < 2; i++)
        if (p->ch[i])
            Free(p->ch[i]);
    delete p;
}
void solve()
{
    string s;
    int cnt = 1;
    root = new tr;
    while (cin >> s) {
        if (s == "9") {
            if (flag) {
                cout << "Set " << cnt++;
                cout << " is not immediately decodable\n";
            }
            else {
                cout << "Set " << cnt++;
                cout << " is immediately decodable\n";
            }
            Free(root);
            root = new tr;
            flag = 0;
            continue;
        }
        insert(s);
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1002

AC代码

```
#include<bits/stdc++.h>
using namespace std;

struct tr {
    int cnt;
    tr* ch[27];
    tr()
    {
        for (int i = 0; i < 26; i++)
            ch[i] = 0;
        cnt = 0;
    }
}* root;
void init()
{
    root = new tr;
}
void insert(string s)
{
    tr* p = root;
    for (int i = 0; s[i] != '\0'; i++) {
        int temp = s[i] - 'a';
        if (p->ch[temp] == 0)
            p->ch[temp] = new tr;
        p = p->ch[temp];
        p->cnt++;
    }
}
void Free(tr* p)
{
    for (int i = 0; i < 26; i++)
        if (p->ch[i])
            Free(p->ch[i]);
    delete p;
}
int count(string s)
{
    tr* p = root;
    for (int i = 0; s[i] != '\0'; i++) {
        int temp = s[i] - 'a';
        if (p->ch[temp] == 0)
            return 0;
        else
            p = p->ch[temp];
    }
    return p->cnt;
}
void solve()
{
    init();
    string s;
    while (getline(cin, s), !s.empty())
        insert(s);
```

```
    while (cin >> s)
        cout << count(s) << endl;
    Free(root);
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1003

AC代码

```
#include<bits/stdc++.h>
using namespace std;

int flag;
struct tr {
    int num;
    tr* ch[11];
    tr()
    {
        for (int i = 0; i < 10; i++) {
            ch[i] = 0;
        }
        num = 0;
    }
}*root;

void insert(string s)
{
    if (flag)return;
    tr* p = root;
    int len = s.size();
    for (int i = 0; i<len; i++) {
        int temp = s[i] - '0';
        if (!p->ch[temp]) {
            p->ch[temp] = new tr;
        }
        p = p->ch[temp];
        if ((p->num == 2)||(p->num==1&&i==len-1)) {
            flag = 1;
            return;
        }
        p->num = 1;
    }
    p->num = 2;
}
void Free(tr* p)
{
    for (int i = 0; i < 10; i++)
```

```
        if (p->ch[i])
            Free(p->ch[i]);
    delete p;
}
void init()
{
    root = new tr;
}
void solve()
{
    string s;
    int n;
    init();
    cin >> n;
    while (n--) {
        cin >> s;
        insert(s);
    }
    if (flag)cout << "NO\n";
    else cout << "YES\n";
    Free(root);
    flag = 0;
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int tt;
    cin >> tt;
    while (tt--)solve();
    return 0;
}
```

1004

AC代码

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    char s[3005], a[100];
    string s1, s2, s3;
    int k=0;
    map<string, string>v;
    cin >> s;
    while (cin >> s1) {
        if (s1 == "END")break;
        cin >> s2;
        v[s2] = s1;
    }
    getchar();
    while (gets(s))
    {
```

```
        if (strcmp(s, "START") == 0)continue;
        if (strcmp(s, "END") == 0)break;
        for (int i = 0; s[i] != '\0'; i++) {
            if (islower(s[i]))
                a[k++] = s[i];
            else {
                a[k] = '\0';
                if (v.count(a))
                    cout << v[a];
                else cout << a;
                cout << s[i];
                for (int j = 0; j < k; j++)
                    a[j] = '\0';
                k = 0;
            }
        }
        cout << endl;
    }
    return 0;
}
```

1005

AC代码

```
#include<bits/stdc++.h>
using namespace std;
string s;
struct tr {
    char c;
    vector<tr*>ch;
    tr() {
        c = 0;
        for (int i = 0; i < ch.size(); i++)
            ch[i] = 0;
    }
}*root,*add;
void insert(tr* p, int cnt)
{
    if (cnt == s.size())
        return;
    for (int i = 0; i < p->ch.size(); i++) {
        if (p->ch[i]->c != s[cnt])
            continue;
        p = p->ch[i];
        insert(p, cnt + 1);
        return;
    }
    add = new tr;
    add->c = s[cnt];
    p->ch.push_back(add);
    p = add;
    insert(p, cnt + 1);
}
```

```
int dfs(tr* p)
{
    int sum = 0;
    int len = p->ch.size();
    for (int i = 0; i < len; i++)
        sum += dfs(p->ch[i]) + 1;
    return sum;
}
void Free(tr* p)
{
    int len = p->ch.size();
    for (int i = 0; i < len; i++)
        Free(p->ch[i]);
    delete p;
}
void solve()
{
    int tt;
    while (cin >> tt) {
        root = new tr;
        int mx = 0;
        for (int i = 0; i < tt; i++) {
            cin >> s;
            insert(root, 0);
            mx = max(mx, int(s.size()));
        }
        cout << dfs(root)*2-mx+tt << endl;
        Free(root);
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1006

AC代码

```
#include<bits/stdc++.h>
using namespace std;

struct tr {
    int num;
    tr* ch[27];
    tr(){
        for (int i = 0; i < 26; i++)
            ch[i] = 0;
        num = 0;
    }
}*root;
```

```cpp
int sum;
void insert(string s)
{
    tr* p = root;
    for (int i = 0; s[i] != '\0'; i++) {
        if (s[i] == ' ') {
            if (i == 0 || s[i - 1] == ' ')
                continue;
            if (p->num == 0)
                p->num = 1, sum++;
            p = root;
            continue;
        }
        int temp = s[i] - 'a';
        if (!p->ch[temp])
            p->ch[temp] = new tr;
        p = p->ch[temp];
    }
    if (p->num == 0 && s[s.size() - 1] != ' ' && s[s.size() - 1] != '\0')
        p->num = 1, sum++;
}

void Free(tr* p)
{
    for (int i = 0; i < 2; i++)
        if (p->ch[i])
            Free(p->ch[i]);
    delete p;
}
void solve()
{
    string s;
    root = new tr;
    while (getline(cin,s),s!="#") {
        sum = 0;
        insert(s);
        cout << sum << endl;
        Free(root);
        root = new tr;
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1007

AC代码

```
#include<bits/stdc++.h>
using namespace std;
char str[50000+5][20];
struct tries{
    tries *next[26];
    bool sign;
    tries(){
        memset(next,NULL,sizeof(next));
        sign=false;
    }
}* root;

tries * create(){
    tries *tmp=new tries();
    return tmp;
}

void Insert(char *p,tries * root){

    while(*p){
        int id=*p-'a';
        if(root->next[id]==NULL){
            root->next[id]=create();
        }
        root=root->next[id];
        p++;
    }
    root->sign=true;
}

bool searchTwo(char *p){
    tries *tmp=root;
    while(*p){
        int id=*p-'a';
        if(tmp->next[id]!=NULL){
            p++;
            tmp=tmp->next[id];
        }
        else return false;
    }
    if(tmp->sign) return true;
    return false;
}

bool searchOne(char *p){
    tries *tmp=root;
    while(*p){
        int id=*p-'a';
        if(tmp->next[id]!=NULL){
            if(tmp->sign){
                if(searchTwo(p))return true;
            }
            p++;tmp=tmp->next[id];
```

```
        }
        else return false;
    }
    return false;
}

int main()
{
    int cnt;
    root=new tries();
    while(scanf("%s",str[cnt])!=EOF){
        Insert(str[cnt],root);
        cnt++;
    }
    for(int i=0;i<cnt;i++){
        if(searchOne(str[i]))
            puts(str[i]);
    }
    return 0;
}
```

1008

AC代码

```
#include<bits/stdc++.h>
const int N=500005;
int n,m,cnt=1,cur;
char str[30];
struct node
{
    int p[26];
    int flag;
    int last;
} tree[N];
void build(char *s,int cur)
{
   int p=0;
   while(*s)
   {
      if(tree[p].p[*s-'a']==-1)
         tree[p].p[*s-'a']=cnt++;
      p=tree[p].p[*s-'a'];
      if(tree[p].last!=cur)
      {
        tree[p].last=cur;
        tree[p].flag++;
      }
      s++;
   }
}

int find(char *s)
{
```

```
    int p=0;
    while(*s)
    {
       if(tree[p].p[*s-'a']==-1)
           return 0;
       p=tree[p].p[*s-'a'];
       s++;
    }
    return tree[p].flag;
}
int main()
{
  int n,i,m,num;
  for(i=0;i<N;i++)
  {
     tree[i].flag=0;
     tree[i].last=-1;
     memset(tree[i].p,-1,sizeof(tree[i].p));
  }
  scanf("%d",&n);
  for(cur=0;cur<n;cur++)
  {
       scanf("%s",str);
      for(i=0;i<=strlen(str);i++)
          build(str+i,cur);
  }
   scanf("%d",&m);
   while(m--)
   {
       scanf("%s",str);
       printf("%d\n",find(str));
   }
    return 0;
}
```

1009

AC代码

```
#include<bits/stdc++.h>
using namespace std;
char word[110],res[110],tmp[110];
char phone[10][4]={{'a','b','c'},{'d','e','f'},{'g','h','i'},{'j','k','l'},
{'m','n','o'},{'p','q','r','s'},{'t','u','v'},{'w','x','y','z'}};
int num[10]={3,3,3,3,3,4,3,4},flag,max_p;

struct tree{
    int cnt;
    tree *next[26];
}*root;

tree *Create(){
    tree *p;
    p=(tree *)malloc(sizeof(tree));
```

```
    p->cnt=0;
    for(int i=0;i<26;++i) p->next[i]=NULL;
    return p;
}

void Insert(char *word,int k){
    tree *p=root;
    int i=0,x,l=strlen(word);
    while(i<l){
        x=word[i++]-'a';
        if((p->next[x])==NULL) p->next[x]=Create();
        p=p->next[x];
        p->cnt+=k;
    }
}

void Search(int l,int len,tree *p){
    if(l==len){
        flag=1;
        if(p->cnt>max_p){
            max_p=p->cnt;
            for(int i=0;i<len;++i) res[i]=tmp[i];
            res[len]='\0';
        }
        return;
    }
    int pos=word[l]-'2',x;
    char t;
    for(int i=0;i<num[pos];++i){
        t=phone[pos][i];
        x=t-'a';
        if((p->next[x])==NULL) continue;
        tmp[l]=t;
        Search(l+1,len,p->next[x]);
    }
}

int main(){
    int t,c=0,n,p,m,mark;
    scanf("%d",&t);
    while(t--){
        printf("Scenario #%d:\n",++c);
        scanf("%d",&n);
        root=Create();
        for(int i=0;i<n;++i){
            scanf("%s %d",word,&p);
            Insert(word,p);
        }
        scanf("%d",&m);
        for(int i=0;i<m;++i){
            mark=0;
            scanf("%s",word);
            int l=strlen(word);
            for(int j=0;j<l-1;++j){
                flag=0,max_p=0;
                if(!mark) Search(0,j+1,root);
                if(flag) printf("%s\n",res);
                else printf("MANUALLY\n"),mark=1;
```

```
            }
            printf("\n");
        }
        printf("\n");
    }
}
```