从后面的状态向前推，如果某一项的长度大于1e18，那么从此项开始，如果有询问此项后面的，也会在1e18的范围内，直接retunn（i-1，y-34）

题意：给你一个数字n，代表f(n)，然后给定数字k，在f(n)中查找第k个字符，如果k > f(n)的长度，返回‘.’；

思路：在f(n)中找到f(n-1)处，递归查找即可

#include<iostream>

#include<algorithm>

#include<cstdio>

#include<stack>

#include<cstring>

#define ll long long

using namespace std;

ll len[100];

ll p;

string s1,s2,s3,s4;

int l1,l2,l3,l4;

char dfs(int n,ll k){

if(n<p && k>len[n])

return '.';

if(n==0)

return s1[k-1];

if(k<=l2)

return s2[k-1];

k-=l2;

if(n>p || k<=len[n-1])

return dfs(n-1,k);

k-=len[n-1];

if(k<=l3)

return s3[k-1];

k-=l3;

if(n>p || k<=len[n-1])

return dfs(n-1,k);

k-=len[n-1];

return s4[k-1];

}

int main()

{

// freopen("input.txt","r",stdin);

ios::sync\_with\_stdio(false);

cin.tie(0);

s1="What are you doing at the end of the world? Are you busy? Will you save us?";

s2="What are you doing while sending \"";

s3="\"? Are you busy? Will you send \"";

s4="\"?";

l1=s1.length();

l2=s2.length();

l3=s3.length();

l4=s4.length();

len[0]=l1;

p=0;

while(len[p]<=1e18){

p++;

len[p]=l2+len[p-1]+l3+len[p-1]+l4;

}

int T,n;

ll k;

cin>>T;

while(T--){

cin>>n>>k;

cout<<dfs(n,k);

}

return 0;

}