#include<bits/stdc++.h>

using namespace std;

vector<string> solve(string s){

if(s=="0"){

vector<string> zero\_base;

zero\_base.push\_back("");

return zero\_base;

}

if(s=="1"){

vector<string> base;

base.push\_back("1");

return base;

}

if(s=="2"){

vector<string> base2;

base2.push\_back("11");

base2.push\_back("2");

return base2;

}

vector<string> ans;

int s\_to\_str = stoi(s);

if(s\_to\_str>=3){

int i1 = s\_to\_str-1;

string ss1 = to\_string(i1);

vector<string> res1 = solve(ss1);

for(string mres1: res1){

string s1 = "1" + mres1;

ans.push\_back(s1);

}

int i2 = s\_to\_str-2;

string ss2 = to\_string(i2);

vector<string> res2 = solve(ss2);

for(string mres2: res2){

string s2 = "2" + mres2;

ans.push\_back(s2);

}

int i3 = s\_to\_str-3;

string ss3 = to\_string(i3);

vector<string> res3 = solve(ss3);

for(string mres3: res3){

string s3 = "3" + mres3;

ans.push\_back(s3);

}

}

return ans;

}

vector<string> stair\_case(int n){

if(n==0){

vector<string> base;

base.push\_back("");

return base;

}

if(n<0){

vector<string> n\_base;

return n\_base;

}

vector<string> paths1 = stair\_case(n-1);

vector<string> paths2 = stair\_case(n-2);

vector<string> paths3 = stair\_case(n-3);

vector<string> res;

for(string path:paths1){

string str = "1"+path;

res.push\_back(str);

}

for(string path:paths2){

string str = "2"+path;

res.push\_back(str);

}

for(string path:paths3){

string str = "3"+path;

res.push\_back(str);

}

return res;

}

int main(){

// string s;

// cin >> s;

int n;

cin>>n;

vector<string> ans = stair\_case(n);

int cnt = 0;

cout << '[';

for (string str : ans){

if (cnt != ans.size() - 1)

cout << str << ", ";

else

cout << str;

cnt++;

}

cout << ']';

}