

2023年12月认证C++ 八级真题解析

CCF 编程能力等级认证，英文名 Grade Examination of Software Programming (以下简称 GESP)，由中国计算机学会发起并主办，是为青少年计算机和编程学习者提供学业能力验证的平台。GESP 覆盖中小学全学段，符合条件的青少年均可参加认证。GESP 旨在提升青少年计算机和编程教育水平，推广和普及青少年计算机和编程教育。

GESP 考察语言为图形化 (Scratch) 编程、Python 编程及 C++ 编程，主要考察学生掌握相关编程知识和操作能力，熟悉编程各项基础知识和理论框架，通过设定不同等级的考试目标，让学生具备编程从简单的程序到复杂程序设计的编程能力，为后期专业化编程学习打下良好基础。

本次为大家带来的是 2023 年 12 月份 C++ 八级认证真题解析。

一、单选题 (每题 2 分，共 30 分)

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	C	A	D	B	C	D	A	C	A	C	D	B	B	A	C

1、小杨要从 A 城到 B 城，又想顺路游览一番。他有两个选项：1、坐高铁路到 C 城游览，再坐高铁或飞机到 B 城；2、坐船到 D 城游览，再坐船、高铁或飞机到 B 城。请问小杨从 A 城到 B 城共有几种交通方案可以选择？（ ）。

- A. 2
- B. 3
- C. 5
- D. 6

【答案】C

【解析】本题可抽象为分类计数问题，应使用加法原理，而不是乘法原理。答案为 ACB 的方案数 2 加上 ADB 的方案数 3=5，选 C。

2、以下哪个函数声明是符合语法的，且在调用时可以将二维数组的名字作为实际参数传递给形式参数 a？（ ）。

- A. void QuickSort(int a[][10], int n);
- B. void QuickSort(int a[5][] , int m);
- C. void QuickSort(int a[][] , int n, int m);
- D. void QuickSort(int ** a, int n, int m);

【答案】A

【解析】C++在函数中把数组作为参数进行传递时，只会传递数组的首地址，函数中如果希望通过首地址计算数组中任意一位元素所处的地址时，就需要知道二维数组的长度，比如二维数组长度为 10 时，a[1][3]的地址就是 a[0][0]的地址+13，本题的选项中只有选项 A 给出了数组二维长度，所以本题选 A。

3、下面有关 C++类和对象的说法，错误的是（ ）。

- A. 对象的生命周期开始时，会执行构造函数。
- B. 对象的生命周期结束时，会执行析构函数。
- C. 类的析构函数可以为虚函数。
- D. 类的构造函数可以为虚函数。

【答案】D

【解析】对象的声明周期开始和结束时分别会执行构造函数和析构函数，选项 A、B 正确。对于选项 C、D，虚函数是指被 virtual 关键字修饰的成员函数，定义虚函数是为了允许用基类的指针来调用派生类的该函数。允许将析构函数定义为虚函数，是因为有使用“delete 基类指针”来销毁对象的需求，选项 C 正确。但对对象构造时必须指定准确的类，不能使用基类名构造派生类的对象，没有将构造函数定义为虚函数的需要，选项 D 错误。

4、使用邻接矩阵表达 n 个顶点的有向图，则该矩阵的大小为（ ）。

- A. $n \times (n+1)$
- B. $n \times n$
- C. $n \times (n-1)$
- D. $n \times (n-1)/2$

【答案】 B

【解析】邻接矩阵的行列均为 $[0 \sim n-1]$,所以矩阵的大小为 $n \times n$, 本题选 B。

5、5 位同学排队，其中一位同学不能排在第一，则共有多少种可能的排队方式？
()。

- A. 5
- B. 24
- C. 96
- D. 120

【答案】 C

【解析】按照第 1,2,3,4,5 位的顺序依次安排同学，某位同学不能在第一位，所以第 1 位有 4 种安排方法，第二位可以从剩余的 4 名同学中选一位，有 4 种方法，依次类推，第 3,4,5 各有 3,2,1 种，总方案数为 $4 \times 4 \times 3 \times 2 \times 1 = 96$ ，选 C。

6、一个无向图包含 n 个顶点，则其最小生成树包含多少条边？()。

- A. $n-1$
- B. n
- C. $n+1$
- D. 最小生成树可能不存在。

【答案】 D

【解析】 n 个顶点组成的树包含 $n-1$ 条边，但是题目没有保证图连通，所以可能不存在最小生成树，选 D。

7、已知三个 `double` 类型的变量 `a`、`b` 和 `theta` 分别表示一个三角形的两条边长及二者的夹角（弧度），则下列哪个表达式可以计算这个三角形的面积？()。

- A. $a * b * \sin(\theta) / 2$
- B. $(a + b) * \sin(\theta) / 2$
- C. $a * b * \cos(\theta) / 2$
- D. $\sqrt{a^2 + b^2 - 2 * a * b * \cos(\theta)}$

【答案】 A

【解析】 若 $\triangle ABC$ 中,已知两边 a, b 和它们的夹角 θ ,作边 a 上的高 h . 则 $S = (1/2)ah$,而 $h/b = \sin(\theta)$,即 $h = b * \sin(\theta)$
 $\therefore S = (1/2)ab \sin(\theta)$ 选 A。

- 8、对有 n 个元素的二叉排序树进行中序遍历,其时间复杂度是 ()。
- A. $O(1)$
 - B. $O(\log(n))$
 - C. $O(n)$
 - D. $O(n^2)$

【答案】 C

【解析】 树的遍历过程需要对每个元素访问一次,因此时间复杂度为 $O(n)$,选择 C。

- 9、假设输入参数 m 和 n 满足 $m \leq n$,则下面程序的最差情况的时间复杂度为()。

```
1 int gcd(int m, int n) {
2     while (m > 0) {
3         int t = m;
4         m = n % m;
5         n = t;
6     }
7     return n;
8 }
```

- A. $O(\log(n))$
- B. $O(n)$
- C. $O(n * m)$
- D. $O(m * \log(n))$

【答案】 A

【解析】本题代码为辗转相除法，复杂度为 $O(\log n)$ 。最差情况，输入为斐波那契数列的相邻两项时，循环次数为输入在数列中的位置。选 A。

10、下面程序的时间复杂度为（ ）。

```
1 long long power_mod(long long a, long long n, long long mod) {
2     if (n == 0)
3         return 1;
4     a = a % mod;
5     if (n == 1)
6         return a;
7     long long pw = power_mod(a, n / 2, mod);
8     long long pw2 = pw * pw % mod;
9     if (n % 2 == 0)
10        return pw2;
11    return pw2 * a % mod;
12 }
```

- A. $O(n)$
- B. $O(a^n)$
- C. $O(\log(n))$
- D. $O(\log(n) \times a)$

【答案】 C

【解析】本题代码为快速幂，复杂度为 $O(\log n)$ 。通过观察可得该函数的时间复杂度只与 n 相关，假设为 $T(n)$ ，则 $T(n)=T(n/2) + \text{常数}$ ，求解可得上述时间复杂度。选 C。

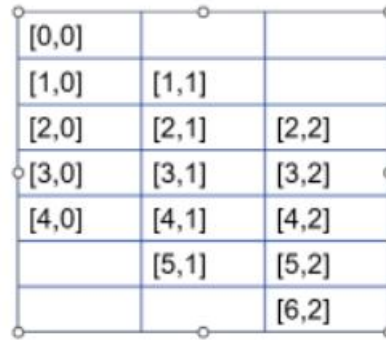
11、下面程序的时间复杂度为（ ）。

```
1 int record_choose[MAXN][MAXM];
2 int choose(int n, int m) {
3     if (m == 0 || m == n)
4         return 1;
5     if (record_choose[n][m] == 0)
6         record_choose[n][m] = choose(n - 1, m - 1) + choose(n - 1, m);
7     return record_choose[n][m];
8 }
```

- A. $O(2^n)$
- B. $O(2m \times (n - m))$
- C. $O(c(n, m))$
- D. $O(m \times (n - m))$

【答案】 D

【解析】本题代码是在计算 $C[n][m]$, 使用了递归的写法并加上了记忆化搜索, 可以通过画图来看所有被访



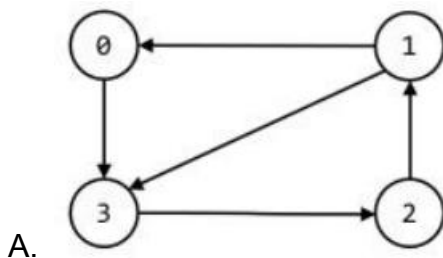
问到的二维数组个数, 以 $n=6, m=2$ 为例, 可以发现访问的元素个数为 $n*m-m*m$, 本题选 D。

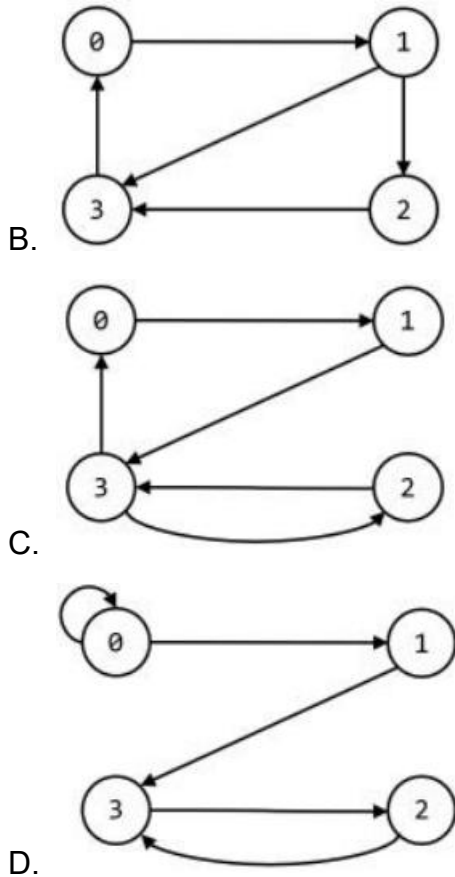
12、下面的程序使用出边的邻接表表达有向图, 则下列选项中哪个是它表达的图? ()。

```

1  #include <iostream>
2
3  struct Edge {
4      int e;
5      Edge * next;
6  };
7  struct Node {
8      Edge * first;
9  };
10
11 int main() {
12     Edge e[5] = {{1, nullptr}, {2, &e[2]},
13                 {3, nullptr}, {3, nullptr}, {0, nullptr}};
14     Node n[4] = {&e[0], &e[1], &e[3], &e[4]};
15     ; // 其他处理
16     return 0;
17 }

```





【答案】B

【解析】结构体 `edge` 里的 `next` 指向的是下一条边，`Node` 里的 `first` 指向的是每个点的第一条边，所以 0 号点指向 1 号点，1 号点指向 2,3 号点，2 号点指向 3 号点，3 号点指向 0 号点，选 B。

第 13 题 下面程序的输出为 ()。

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int cnt = 0;
6     for (int a = 1; a <= 10; a++)
7         for (int b = 1; b <= 10; b++)
8             for (int h = 1; h <= 10; h++)
9                 if ((a + b) * h == 20)
10                    cnt++;
11     cout << cnt << endl;
12     return 0;
13 }
```

A. 12

- B. 18
- C. 36
- D. 42

【答案】 B

【解析】代码中 a, b, h 的取值范围均为 $[1, 10]$ ，要 $(a+b) \cdot h = 20$ ，那么可能的 h 有 1, 2, 4, 5, 10， h 为 1 时， $(a+b) = 20$ ，有 1 种方法， h 为 2 时， $(a+b) = 10$ ，有 9 种方案，依次计算出 h 为 4, 5, 10 时， $(a+b)$ 的方案数依次为 4, 3, 1，总方案数为 $1+9+4+3+1=18$ ，选 B。

14、下面程序的输出为 ()。

```

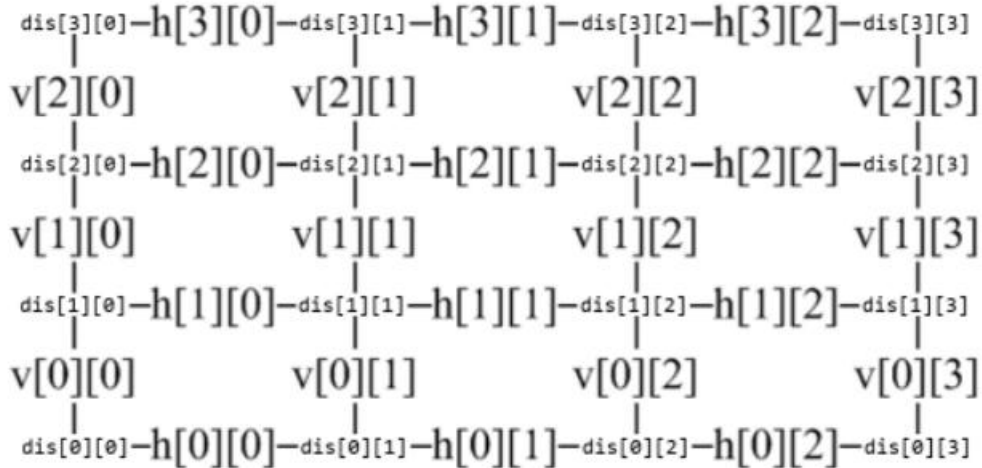
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      const int N = 30;
6      int cnt = 0;
7      for (int a = 1; a <= N; a++)
8          for (int b = a; a + b <= N; b++)
9              for (int c = b; a + b + c <= N; c++)
10                 if (a * a + b * b == c * c)
11                     cnt++;
12     cout << cnt << endl;
13     return 0;
14 }
```

- A. 3
- B. 6
- C. 11
- D. 22

【答案】 A

【解析】代码中要求 a, b, c 都是正整数，满足 $a+b+c \leq 30$ 且，符合要求的 a, b, c 有 3 4 5； 5 12 13； 6 8 10 共 3 种，选 A。

15、下面的程序中，二维数组 h 和 v 分别代表如下图所示的网格中的水平边的时间消耗和垂直边的时间消耗。程序使用动态规划计算从左下角到右上角的最小时间消耗，则横线处应该填写下列哪个选项的代码？ ()。



```

1 int dis[MAXY][MAXX];
2 int shortest_path(int x, int y) {
3     dis[0][0] = 0;
4     for (int i = 0; i < y; i++)
5         dis[i + 1][0] = dis[i][0] + v[i][0];
6     for (int j = 0; j < x; j++)
7         dis[0][j + 1] = dis[0][j] + h[0][j];
8     for (int i = 0; i < y; i++)
9         for (int j = 0; j < x; j++)
10            _____; // 在此处填写代码
11     return dis[y][x];
12 }

```

- A. $\text{dis}[i][j] = \min(\text{dis}[i - 1][j] + \text{v}[i - 1][j], \text{dis}[i][j - 1] + \text{h}[i][j - 1]);$
- B. $\text{dis}[i][j] = \min(\text{dis}[i - 1][j] + \text{h}[i - 1][j], \text{dis}[i][j - 1] + \text{v}[i][j - 1]);$
- C. $\text{dis}[i + 1][j + 1] = \min(\text{dis}[i][j + 1] + \text{v}[i][j + 1], \text{dis}[i + 1][j] + \text{h}[i + 1][j]);$
- D. $\text{dis}[i + 1][j + 1] = \min(\text{dis}[i][j + 1] + \text{h}[i][j + 1], \text{dis}[i + 1][j] + \text{v}[i + 1][j]);$

【答案】C

【解析】观察到 11 行的输出为 $\text{dis}[y][x]$ ，而我们枚举的范围是 $<y$ 与 $<x$ ，所以第 10 行计算的肯定是 $\text{dis}[i+1][j+1]$ ，排除 AB，接着看 C 选项第一个转移， $\text{dis}[i][j+1]$ ，说明这里是从第一维转移过来，而第 5 行也是从第一维转移过来的，使用的是 v 数组，选项 C 正确，排除 D 选项，本题选 C。

二、判断题 (每题 2 分, 共 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	×	√	√	×	√	√	×	×	×	√

1、C++语言非常强大，可以用来求解方程的解。例如，如果变量 x 为 `double` 类型的变量，则执行语句 `x * 2 - 4 = 0;` 后，变量 x 的值会变为 2.0。

【答案】错误

【解析】错误。`x*2-4=0;`既不是判断语句，也不是赋值语句。它不是合法的 C++ 语句，不能通过编译。

2、一个袋子中有 3 个完全相同的红色小球、2 个完全相同的蓝色小球。每次从中取出 1 个，且不放回袋子，这样进行 3 次后，将取出的小球依次排列，则可能的颜色顺序有 7 种。

【答案】正确

【解析】正确，可能出现红红红，蓝红红，红蓝红，红红蓝，蓝蓝红，红蓝蓝，蓝红蓝。

3、杨辉三角，是二项式系数的一种三角形排列，在中国南宋数学家杨辉 1261 年所著的《详解九章算法》一书中出现，是中国数学史上的一项伟大成就。

【答案】正确

【解析】正确，杨辉三角，是二项式系数在三角形中的一种几何排列，在中国南宋数学家杨辉所著的《详解九章算法》中出现。

4、 N 个顶点的有向完全图（不带自环）有 $N \times (N-1)/2$ 条边。

【答案】错误

【解析】错误。有向完全图， (x,y) 和 (y,x) 不算同一条边，所以有 $N*(N-1)$ 条边。

5、如果待查找的元素确定，只要哈希表的大小不小于查找元素的个数，就一定存在不会产生冲突的哈希函数。

【答案】正确

【解析】正确。极端情况，可以先将待查找元素放进哈希表中，再根据位置构造一个由 if-else 判断组成的哈希函数：当查找元素与某一项待查找相同时，返回对应的位置。虽然这样构造的哈希函数的时间复杂度较高，但满足不会产生冲突。

6、动态规划算法的时间复杂度一般为：必要状态的数量，乘以计算一次状态转移方程的时间复杂度。

【答案】正确

【解析】正确，动态规划的时间复杂度一般为状态数*转移复杂度。

7、已知 int 类型的变量 a、b 和 h 中分别存储着一个梯形的顶边长、底边长和高，则这个梯形的面积可以通过表达式 $(a + b) * h / 2$ 求得。

【答案】错误

【解析】错误，梯形面积可能带有小数，不能直接/2。

8、判断图是否连通只能用广度优先搜索算法实现。

【答案】错误

【解析】错误,还可以使用深度优先搜索算法。

9、在 N 个元素的二叉排序树中查找一个元素，最好情况的时间复杂度是 $O(\log N)$ 。

【答案】错误

【解析】错误，最好情况的时间复杂度为 $O(1)$ ，即二叉排序树的根即为查找的元素。

10、给定 double 类型的变量 x，且其值大于等于 0，我们可以通过二分法求出的 \sqrt{x} 近似值。

【答案】正确

【解析】正确，函数 $y = x^2$ 在值域 $[0, +\infty]$ 上具有单调性。

三、编程题（每题 25 分，共 50 分）

题号	1	2
答案		

1、奖品分配

问题描述

班上有 N 名同学，学号从 0 到 $N-1$ 。有 M 种奖品要分给这些同学，其中，第 i 种奖品总共有 a_i 个 ($i = 0, 1, \dots, M-1$)。巧合的是，奖品的数量不多不少，每位同学都可以恰好分到一个奖品，且最后剩余的奖品不超过 1 个（即：

$$N \leq a_0 + a_1 + \dots + a_{M-1} \leq N+1$$

现在，请你求出每个班级礼物分配的方案数，所谓方案，指的是为每位同学都分配一个种类的奖品。只要有一位同学获得了不同种类的奖品，即视为不同的方案。方便起见，你只需要输出方案数对 10^9+7 取模后的结果即可。

共有 T 个班级都面临着奖品分配的问题，你需要依次为他们解答。

输入描述

第一行一个整数 T ，表示班级数量。

接下来 T 行，每行若干用单个空格隔开的正整数。首先是两个正整数 N, M ，接着是 M 个正整数 a_0, a_1, \dots, a_{M-1} ，保证 $N \leq a_0 + a_1 + \dots + a_{M-1} \leq N+1$ 。

输出描述

输出 T 行，每行一个整数，表示该班级分配奖品的方案数对 10^9+7 取模的结果。

特别提醒

在常规程序中，输入、输出时提供提示是好习惯。但在本场考试中，由于系统限定，请不要在输入、输出中附带任何提示信息。

样例输入 1

```

1 | 3
2 | 3 2 1 2
3 | 3 2 1 3
4 | 5 3 3 1 1

```

样例输出 1

```

1 | 3
2 | 4
3 | 20

```

样例解释 1

对于第1个班级，学号为0,1,2的同学可以依次分别获得奖品 0,1,1,也可以依次分别获得奖品1,0,1,也可以依次分别获得奖品1,1,0,因此共有3种方案。

对于第2个班级，学号为0,1,2 的同学可以依次分别获得奖品0,1,1,也可以依次分别获得奖品 1,0,1,也可以依次分别获得奖品1,1,0,可以依次分别获得奖品1,1,1,因此共有4种方案。

对于第3个班级，可以把编号为1的奖品分配给5名同学中的任意一名，共有5种方案；再把编号为2的奖品分配给剩余4名同学中的任意一名，共有4种方案；最后给剩余3名同学自然获得0号奖品。因此，方案数为 $5 \times 4 = 20$ 。

样例输入2

```
1 5
2 100 1 100
3 100 1 101
4 20 2 12 8
5 123 4 80 20 21 3
6 999 5 101 234 499 66 99
```

样例输出2

```
1 1
2 1
3 125970
4 895031741
5 307187590
```

数据规模

对于30%的测试点，保证 $N \leq 10$ 。

对于另外30%的测试点，保证 $M = 2$ 。

对于所有测试点，保证 $N \leq 1,000$; 保证 $T \leq 1,000$; 保证 $M \leq 1,001$ 。

【解题思路】首先考虑奖品总个数=人数的情况，设奖品数分别是 a_0, a_1, \dots, a_{M-1} , 则总方案数为 $C_n^{a_0} * C_{n-a_0}^{a_1} * \dots * C_{a_{M-1}}^{a_{M-1}}$ ，可以提前预处理排列数C数组的值，方便计算，接着考虑奖品总个数=人数+1的情况，我们可以额外虚设1个人，依然按照奖品总个人=人数的情况计算，并输出方案数，具体的方案为不考虑虚设人拿的奖品种类，把前n个人拿的奖品种类作为方案，考虑为什么这样是对的，对于两种方案，如果虚设人拿的奖品编号相同，那么说明前面n个人拿的奖品编号必然有不同，那么方案是不同的，如果虚设人拿的奖品编号不同，因为总共只有n+1个奖品，所以前面n个人拿的奖品编号必然也不同，方案也是不同的，说明我们在虚



设人的情况下得到的任意两个方案都不会重复，所以直接按照人数=奖品总个数计算即可。

【参考程序】

```
1  #include <cstdio>
2  #include <cstdlib>
3  #include <cstring>
4  #include <algorithm>
5  #include <string>
6  #include <map>
7  #include <iostream>
8  #include <cmath>
9  #include <vector>
10 using namespace std;
11 const int N = 1005;
12 const int mod = 1e9 + 7;
13 int C[N + 5][N + 5], a[N + 5];
14 void add(int &a, int b) {
15     a += b; if(a >= mod) a -= mod;
16 }
17 void init() {
18     C[0][0] = 1;
19     for(int i = 1; i <= N; i++) {
20         C[i][0] = C[i][i] = 1;
21         for(int j = 1; j < i; j++)
22             C[i][j] = C[i - 1][j - 1] + add(C[i][j], C[i - 1][j]);
23     }
24 }
25 int main() {
26     // freopen("data/10.in", "r", stdin);
27     init();
28     int T;
29     scanf("%d", &T);
30     while(T --) {
31         int n, m, sum = 0;
32         scanf("%d%d", &n, &m);
33         for(int i = 1; i <= m; i++)
34             scanf("%d", &a[i]), sum += a[i];
35
36         int ans = 1;
37         for(int i = 1; i <= m; i++) {
38             ans = 1ll * ans * C[sum][a[i]] % mod;
39             sum -= a[i];
40         }
41         cout << ans << endl;
42     }
43     return 0;
44 }
```

2、大量的工作沟通

问题描述

某公司有 N 名员工，编号从 0 至 $N-1$ 。其中，除了 0 号员是老板，其余每名员工都有一个直接领导。我们假设编号为 i 的员工的直接领导是 f_i 。

该公司有严格的管理制度，每位员工只能受到本人或直接领导或间接领导的管理。

具体来说，规定员工 x 可以管理员工 y ，当且仅当 $x=y$ ，或 $x=f_y$ 或 x 可以管理 f_y 。

特别， 0 号员工老板只能自我管理，无法由其他任何员管理。

现在，有一些同事要开展合作，他们希望找到一位同事来主持这场合作，这位同事必须能够管理参与合作的所有同事。如果有多名满足这一条件的员工，他们希望找到编号最大的员工。你能帮帮他们吗？

输入描述

第一行一个整数 N ，表示员工的数量。

第二行 $N-1$ 个用空格隔开的正整数，依次为 f_1, f_2, \dots, f_{n-1} 。

第三行一个整数 Q ，表示共有 Q 场合作需要安排。

接下来 Q 行，每行描述一场合作：开头是一个整数 $m(2 \leq m \leq N)$ ，表示参与本次合作的员工数量；接着是 m 个整数，依次表示参与本次合作的员工编号（保证编号合法且不重复）。

保证公司结构合法，即不存在任意一名员工，其本人是自己的直接或间接领导。

输出描述

输出 Q 行，每行一个整数，依次为每场合作的主持人选。

特别提醒

在常规程序中，输入、输出时提供提示是好习惯。但在本场考试中，由于系统限定，请不要在输入、输出中附带任何提示信息。

样例输入 1

```
1 5
2 0 0 2 2
3 3
4 2 3 4
5 3 2 3 4
6 2 1 4
```

样例输出 1

```
1 2
2 2
3 0
```

样例解释 1

对于第一场合作，员工3, 4有共同领导2，可以主持合作。

对于第二场合作，员工2本人即可以管理所有参与者。

对于第三场合作，只有0号老板才能管理所有员工。

样例输入2

1	7
2	0 1 0 2 1 2
3	5
4	2 4 6
5	2 4 5
6	3 4 5 6
7	4 2 4 5 6
8	2 3 4

样例输出2

1	2
2	1
3	1
4	1
5	0

数据规模

对于25%的测试点，保证 $N \leq 50$ 。

对于50%的测试点，保证 $N \leq 300$ 。

对于所有测试点，保证 $3 \leq N \leq 10^5$; 保证 $Q \leq 100$ ，保证 $m \leq 10^4$

【解题思路】容易想到，能够管理员工 a_1, a_2, \dots, a_m 的人为 $\text{lca}(a_1, a_2, \dots, a_m)$ 以及该 lca 的祖先节点，题目还要求编号最大，也就是从根到该 lca 路径上所有点的编号最大值，这个可以提前预处理，设 $\text{mxID}[u]$ 表示从根到 u 路径上所有节点编号的最大值，对于每组询问，首先求出所有参与的员工的 lca ，接着输出 $\text{mxID}[\text{lca}]$ 即可，求 lca 比较常用的有倍增法或树链剖分等。

【参考程序】



```
1 #include <cstdio>
2 #include <cstdlib>
3 #include <cstring>
4 #include <algorithm>
5 #include <string>
6 #include <map>
7 #include <iostream>
8 #include <cmath>
9 #include <vector>
10 #include <queue>
11 using namespace std;
12 const int N = 100005;
13 int fa[N], sz[N], dep[N], son[N], tp[N], mxId[N];
14 int cnt, fir[N], tar[N], nxt[N];
15 void link(int a, int b) {
16     tar[++ cnt] = b, nxt[cnt] = fir[a], fir[a] = cnt;
17 }
18 void dfs(int x, int mxid) {
19     int Mx = 0; sz[x] = 1;
20     mxId[x] = max(x, mxid);
21     for (int i = fir[x]; i; i = nxt[i]) {
22         dep[tar[i]] = dep[x] + 1;
23         dfs(tar[i], mxId[x]);
24         sz[x] += sz[tar[i]];
25         if (Mx < sz[tar[i]])
26             Mx = sz[son[x] = tar[i]];
27     }
28 }
29 void gettp(int x) {
30     tp[x] = x;
31     if (son[fa[x]] == x)
32         tp[x] = tp[fa[x]];
33     for (int i = fir[x]; i; i = nxt[i])
34         gettp(tar[i]);
35 }
36 int lca(int x, int y) {
37     while (tp[x] != tp[y])
38         dep[tp[x]] > dep[tp[y]] ? x = fa[tp[x]] : y = fa[tp[y]];
39     return dep[x] < dep[y] ? x : y;
40 }
41
42 int main() {
43     int n;
44     scanf("%d", &n);
45     for (int i = 2; i <= n; i++)
46         scanf("%d", &fa[i]), link(++ fa[i], i);
47     dfs(1, 1), gettp(1);
48
49     int q;
50     scanf("%d", &q);
51     while (q --) {
52         int m, x, y;
53         scanf("%d%d", &m, &x), x ++;
54         for (int i = 2; i <= m; i++)
55             scanf("%d", &y), x = lca(x, ++ y);
56         cout << mxId[x] - 1 << endl;
57     }
58     return 0;
59 }
```



CCF-GESP编程能力等级认证
Grade Examination of Software Programming

GESP 2024年3月认证
认证语言: C++/Python/图形化编程
报名时间: 2024年1月18日至3月5日24点截止
缴费时间: 2024年1月18日至3月5日24点截止
认证时间: 1-4级 2024年3月16日 上午 09:30-11:30
5-8级 2024年3月16日 下午 13:30-16:30
认证方式: 全国统一线下机考

扫码即刻报名

GESP面向全国征集授权服务中心和考点 (考点仅限公立校)
Scratch 欢迎申请, 扫码至官网了解更多

【联系我们】

1. GESP微信: 关注“CCF GESP”公众号, 将问题以文字方式留言即可得到回复。
2. GESP邮箱: gesp@ccf.org.cn
注: 请在邮件中详细描述咨询的问题并留下考生的联系方式及姓名、身份证号, 以便及时有效处理。
3. GESP电话: 0512-67656856
咨询时间: 周一至周五(法定节假日除外): 上午 8:30-12:00; 下午 13:00-17:30

GESP第五期认证报名已启动, 扫描下方二维码, 关注GESP公众号即可报名

