class FS\_Background

{

    public string Name { get; set; }

    public bool Type { get; set; }

    public FS\_Background(string name, bool type)

    {

        Name = name;

        Type = type;

        Console.WriteLine("3");

    }

    static void parvietoties(string currentDirectory)

    {

        while (true)

        {

            Console.WriteLine($"\nCurrent Directory: {currentDirectory}");

            Console.WriteLine("Directories:");

            foreach (string dir in Directory.GetDirectories(currentDirectory))

            {

                Console.WriteLine("  [D] " + Path.GetFileName(dir));

            }

            Console.WriteLine("Files:");

            foreach (string file in Directory.GetFiles(currentDirectory))

            {

                Console.WriteLine("  [F] " + Path.GetFileName(file));

            }

            Console.WriteLine("\nCommands:");

            Console.WriteLine("cd <directory>  - change directory");

            Console.WriteLine("up              - go up one level");

            Console.WriteLine("exit            - exit program");

            Console.Write("\nEnter command: ");

            string input = Console.ReadLine();

            string[] parts = input.Split(' ', 2, StringSplitOptions.RemoveEmptyEntries);

            if (parts.Length == 0)

                continue;

            string command = parts[0].ToLower();

            if (command == "exit")

                break;

            else if (command == "cd" && parts.Length > 1)

            {

                string targetDir = Path.Combine(currentDirectory, parts[1]);

                if (Directory.Exists(targetDir))

                {

                    currentDirectory = targetDir;

                }

                else

                {

                    Console.WriteLine("Directory not found.");

                }

            }

            else if (command == "up")

            {

                string parent = Directory.GetParent(currentDirectory)?.FullName;

                if (parent != null)

                {

                    currentDirectory = parent;

                }

                else

                {

                    Console.WriteLine("Already at the root directory.");

                }

            }

            else

            {

                Console.WriteLine("Unknown command.");

            }

        }

    }

    static void dzest(string Name)

    {

        if (!File.Exists(Name))

        {

            Console.WriteLine("File not found.");

            return;

        }

        Console.WriteLine("Type command:");

        string command = Console.ReadLine();

        if (command == "del Y")

        {

            string fileContent = File.ReadAllText(Name);

            // Simple regex to match constructors (very basic)

            string className = Path.GetFileNameWithoutExtension(Name);

            string pattern = $@"\s\*public\s+{className}\s\*\([^)]\*\)\s\*\{{[^{{}}]\*\}}";

            Console.WriteLine("Constructors deleted.");

        }

        else

        {

            Console.WriteLine("Unknown command.");

        }

    }

    static void izveidot(string name, int amount)

    {

        for (int i = 0; i < amount; i++)

        {

            FS\_Background a = new FS\_Background(name, true);

        }

    }

    static void Main()

    {

        try

        {

            string command\_RAW = Console.ReadLine();

            string[] command = command\_RAW.Split(' ');

            if (command[0].ToLower() == "mkdir")

            {

                izveidot(command[1], 1);

                Console.WriteLine("4");

            }

            else

            {

                Console.WriteLine("5");

            }

        }

        catch (Exception e)

        {

            Console.WriteLine(e.Message);

            Main();

        }

    }

}