

单选 (3分)

2. 在一个请求页式存储系统中，一程序的页面走向为：5,4,3,2,5,4,6,5,4,3,2,6，并采用最近最久未使用页面淘汰算法。设分配给该程序的存储块数是4，则访问过程中发生的缺页率（）

A 5/12

B 6/12

C 7/12

D 8/12

下一题 →

🚩 标记

单选 (3分)

3. 假设有三个文件拥有者，分别为 usr1, usr2 和 usr3，他们同属 testgroup 这个群组，给定如下权限展示：

```
-rwxr-xr--  1  usr1      testgroup      5238  Jun  19  10:25  ping_test
```

对于上述权限相关的结论，下列选项中错误的是（）

A usr1 针对文件 ping_test 具有可读可写可执行的权限

B usr2 针对文件 ping_test 具有可读和可执行的权限

C 未加入 testgroup 群组的用户对于文件 ping_test 仅具备可执行的权限

D usr3 针对文件 ping_test 不具备可写的权限

下一题 →

🚩 标记

单选 (3分)

5. 访问者模式是设计模式中相对简单的一种模式，正因如此，访问者模式有许多随之而来的优点，下列选项中，不属于访问者模式优点的是（）

A 访问者模式类中的角色职责清晰，符合单一职责的原则

B 访问者模式具有非常优秀的扩展性

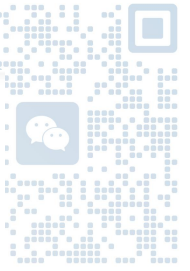
C 访问者模式具体元素容易变更，且无法访问一个类的任何内部细节

D 访问者模式适用于业务规则要求遍历多个不同对象的场景

下一题 →

🚩 标记





单选 (3分)

6. 若一个 n 阶方阵 A 满足：“对于任意 $i, j \in [0, 9]$ ”，方阵 A 满足： $A_{i,j} = A_{j,i}$ ，则称其为 n 阶对称矩阵。在一个 $A[0 \dots 9, 0 \dots 9]$ 的对称矩阵中，采用压缩存储（行优先）的方式存储其下三角部分，已知每个元素占用两个存储单元，第一个元素 $A[0,0]$ 的存储地址为500，则求 $A[6][7]$ 对应元素的存储地址为（）

A 564

B 566

C 568

D 570

下一题 →

🚫 标记

不定项

(3分)

⚠️ 错选不得分,少选得1/3分

1. 以下进程状态转换中，错误的是（）。

A 阻塞挂起→就绪挂起→阻塞挂起

B 运行→就绪挂起→就绪

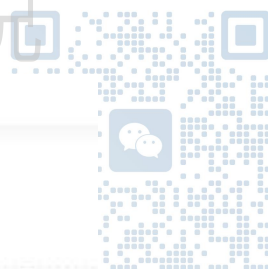
C 就绪→阻塞→阻塞挂起

D 阻塞→就绪→运行

下一题 →

🚫 标记





不定项 (3分) 错选不得分,少选得1/3分

2. 设计模式中, 下面属于结构型模式的有 ()

- A 适配器模式
- B 装饰器模式
- C 工厂方法模式
- D 单例模式

下一题 →

标记

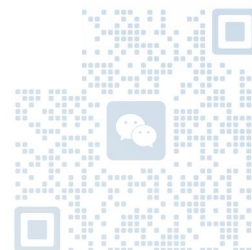
不定项 (3分) 错选不得分,少选得1/3分

3. 根据栈中后进先出/先进后出的原则, 如果栈中输入的序列是12345, 且在入栈过程中可以出栈, 则正确的输出结果是 ()

- A 34251
- B 52134
- C 45231
- D 32154

下一题 →

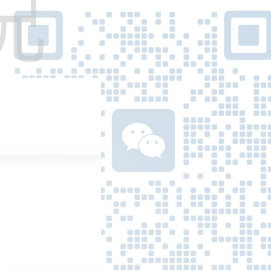
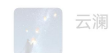
标记



云澜师兄

云澜师兄

云澜师兄



不定项 (3分) 错选不得分,少选得1/3分

4. 以下排序算法中, 通过比较相邻元素进行交换的是 ()

A 冒泡排序

B 插入排序

C 基数排序

D 希尔排序

下一题 →

标记



云澜师兄

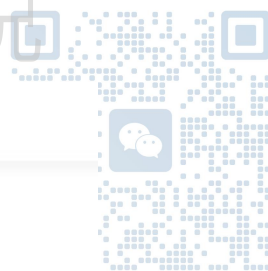
云澜师兄



云澜师兄

云澜师兄

云澜师兄



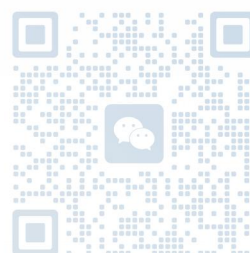
不定项 (3分) 错选不得分,少选得1/3分

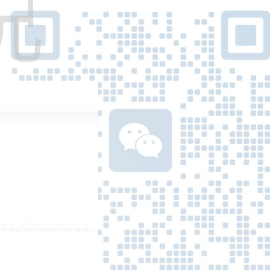
5. HTTP请求报文的请求行包含哪些字段?

- A 方法字段
- B URL字段
- C 主机字段
- D HTTP版本字段

下一题 →

标记





不定项 (3分) 错选不得分,少选得1/3分

7. 下面关于 Linux 中磁盘分区的说法正确的是 ()

- A 在 Linux 中,常用的磁盘分区类型有主分区、扩展分区和逻辑分区
- B 合理划分磁盘分区,可以实现数据的分离、管理和保护
- C 磁盘分区不会影响数据的存储和访问速度
- D 每个分区都可以被格式化为特定的文件系统类型,例如 Ext4、XFS、Btrfs 等

提文本题型

标记

Google Chrome 不兼容的默认浏览器 设为默认

2026届普招-报了么-工程0314 新考试链接: 01:21:39

反向构造 (100) 每题以多次提交中最高得分计

991 时间限制: 1秒

992 空间限制: 262144

语言: C/C++(1), C++(clang++11), Pascal(3), Java(jdk8), Python(2.7.9), PHP(7.4.7), C#(5.4), Objective-C(5.4), JavaScript(Node12.18.2), R(4.0.3), Go(1.14.4), Ruby(2.7.1), Swift(5.3), MATLAB(Octave 5.2), pypy2(pypy2.7.13), pypy3(pypy3.6.1), Rust(1.44), Scala(2.11.12), Kotlin(1.4.10), Groovy(3.0.6), TypeScript(4.1.2)

64bit IO Format: %ld

由企业出题,本考场试题暂时禁止离开页面,不会将本题ID,请使用考试系统提供的在线编辑进行编辑。请谨慎小心保存并提交,避免因提交失败。

小红拿到一份英文报纸,由一个长度为 n 的字符串组成,字符串仅由 '0','X','Y','Z','m','w', 组成。

小红把这份报纸抄了一遍,后来发现拿报纸的时候拿反了,因此抄写的字符串和报纸上表达的字符串上下颠倒了,比如 'w' 写成了 'm', 'm' 写成了 'w', 其他字母不变。

现在给出小红抄写的字符串,请你还原报纸上所表达的字符串。

输入描述

第一行一个长度为 $n(1 \leq n \leq 10^5)$, 表示字符串长度。

第二行一个长度为 n 且仅由 '0','X','Y','Z','m','w' 组成的字符串,表示小红抄写的字符串。

输出描述

一个长度为 n 的字符串,表示报纸上的字符串。

示例 1

输入

5

0X0ZW

输出

0W0ZX

下一题 标记

```
1 import java.util.Scanner;
2
3 // 注意类名必须为 Main, 不要有任何 package xxx 信息
4 public class Main {
5     public static void main(String[] args) {
6         Scanner in = new Scanner(System.in);
7         // 注意 hasNext 和 hasNextLine 的区别
8         while (in.hasNextInt()) { // 注意 while 处理多个 case
9             int a = in.nextInt();
10            int b = in.nextInt();
11            System.out.println(a + b);
12        }
13    }
14 }
```

执行结果 自动输入

点击保存并提交,系统将保存你提交的代码并返回运行结果。

本题累计运行2次

| 运行ID | 运行时间 | 运行类型 | 运行结果 | 用例通过数 | 运行时长 | 占用内存 |
|------|------|------|------|-------|------|------|
| | | | | | | |

提交答案



小红的排列构造 (2025) 每道题以多次提交中最高得分计分

时间限制: 1秒
空间限制: 262MB

语言支持: C/C++(clang11), C++(clang++11), Pascal(pascal 3.0.2), Java(javac 1.8), Python2(2.7.3), PHP(7.4.7), C#(mcs 5.4), Objective-C(5.4), Python3(3.9), JavaScript(Node12.18.2), R(R4.0.3), Go(1.14.4), Ruby(2.7.1), Swift(5.3), Matlab(Octave 5.2), pypy2(pypy2.7.13), pypy3(pypy3.6.1), Rust(1.44), Scala(2.11.12), Kotlin(1.4.10), Groovy(3.0.6), TypeScript(4.1.2)

Special Judge, 64bit IO Format: %lld

企业要求: 本场考试全程禁止离开页面, 不支持本地IDE, 请使用考试系统提供的在线编辑器进行调试, 调试后务必点击“保存并提交”按钮提交代码。

小红希望你构造一个长度为 n 的排列, 满足 $\sum_{i=1}^n a_i \cdot i$ 恰好是3的倍数。
定义排列是一个长度为 n 的数组, 满足 $a_i \in [1, n]$ 且不存在两个元素相同。

输入描述

第一行输入一个正整数 t , 代表测试用例的组数。
接下来 t 行, 每行输入一个正整数 n 。
 $1 \leq t \leq 10^4$
 $1 \leq n \leq 10^5$
保证所有 n 的总和不超过 10^6 。

输出描述

对于每组测试用例如果无解, 请输出-1。
否则输出 n 个正整数 a_i , 代表小红构造的排列。有多解时输出任意合法解。

示例 1

输入

```
2
2
4
```

输出

```
1 2
1 2 3 4
```

下一题 标记

Java ACM 模式

```
1 import java.util.Scanner;
2
3 // 注意类名必须为 Main, 不要有任何 package xxx 信息
4 public class Main {
5     public static void main(String[] args) {
6         Scanner in = new Scanner(System.in);
7         // 注意 hasNext 和 hasNextLine 的区别
8         while (in.hasNextInt()) { // 注意 while 处理多个 case
9             int a = in.nextInt();
10            int b = in.nextInt();
11            System.out.println(a + b);
12        }
13    }
14 }
```

执行结果 自动输入

2026届春招-报了么-工程0314 最新考试结果: 01:20:55

小红的数字问题 (2025) 每道题以多次提交中最高得分计分

时间限制: 1秒
空间限制: 262MB

语言支持: C/C++(clang11), C++(clang++11), Pascal(pascal 3.0.2), Java(javac 1.8), Python2(2.7.3), PHP(7.4.7), C#(mcs 5.4), Objective-C(5.4), Python3(3.9), JavaScript(Node12.18.2), R(R4.0.3), Go(1.14.4), Ruby(2.7.1), Swift(5.3), Matlab(Octave 5.2), pypy2(pypy2.7.13), pypy3(pypy3.6.1), Rust(1.44), Scala(2.11.12), Kotlin(1.4.10), Groovy(3.0.6), TypeScript(4.1.2)

64bit IO Format: %lld

企业要求: 本场考试全程禁止离开页面, 不支持本地IDE, 请使用考试系统提供的在线编辑器进行调试, 调试后务必点击“保存并提交”按钮提交代码。

对于给定的整数 n , 对于全部的二元组 (i, j) 满足 $1 \leq i < j \leq n$, 计算 $\frac{i+j}{\gcd(i, j)}$ 之和。由于答案可能很大, 请将答案对 $(10^9 + 7)$ 取模后输出。

gcd, 即最大公约数, 指两个整数共有约数中最大的一个。例如, 12 和 30 的公约数有 1, 2, 3, 6, 其中最大的约数是 6, 因此 $\gcd(12, 30) = 6$ 。

输入描述

每个测试文件均包含多组测试数据。第一行输入一个整数 T ($1 \leq T \leq 10^5$) 代表数据组数。
每组测试数据能描述如下:
在一行上输入一个整数 n ($1 \leq n \leq 10^6$) 代表给定的整数。
除此之外, 保证整个测试文件的 n 之和不超过 10^6 。

输出描述

对于每一组测试数据, 输出一行, 输出一个整数, 代表答案对 $(10^9 + 7)$ 取模后的结果。

示例 1

输入

```
4
2
2
4
7
```

输出

```
1
1
14
105
```

提交并测试 标记

Java ACM 模式

```
1 import java.util.Scanner;
2
3 // 注意类名必须为 Main, 不要有任何 package xxx 信息
4 public class Main {
5     public static void main(String[] args) {
6         Scanner in = new Scanner(System.in);
7         // 注意 hasNext 和 hasNextLine 的区别
8         while (in.hasNextInt()) { // 注意 while 处理多个 case
9             int a = in.nextInt();
10            int b = in.nextInt();
11            System.out.println(a + b);
12        }
13    }
14 }
```

执行结果 自动输入

