技术小组纳新试题

以下共有8题，题目难度和类型也不相同，你可以按照自己的情况选取若干题目完成，三至四题即可。其中第0题包含了题目解答上传的途径，因此必须完成。在解题过程中，你可以搜索相关资料，但请不要与其他人交流。

这些题目旨在了解各位的即时学习能力以及面向文档的学习能力，每道题目对于你来说都可能是完全陌生的，我们不需要你有多么优秀的基础，但我们欣赏你努力解题的过程。

如果在努力尝试后仍未能完全解答，可以谈一谈你对这些题目的理解，也可以将你对题目的探索过程作为附件提交，我们会酌情加分。

技术小组的纳新提交将于2024年10月18日23:59截止，逾期不候哦~

Have fun coding~

0. 来签到啦！- Git

作为一位（准）开发人员，你会遇到很多很多与他人合作完成项目的场景。你也许听说过“这个项目我Fork了”“这个项目我Star了”“我提交了一个PR”这样的说法，其实这都与Git息息相关。

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。它是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

你的任务

1. 新建一个Git远程仓库，并保证技术小组管理员可以访问到你的仓库，托管平台不限。
2. 使用你的学号作为密钥，将管理员发送给你的一串密文进行解密，并将解密结果放在一个文本文件里，和其他题目的作答分别放在以题目序号命名的文件夹中，推送到你自己的远程仓库。文件格式不限。
3. 技术小组的纳新试题也会同步放在ZJU Git,Github以及Gitea托管平台，当你完成时，请在纳新试题仓库提出issue，并附上你的个人仓库链接。

明文的加密过程

1. 明文固定4位，密钥固定学号后4位
2. 将明文与学号对应位置的字符先转化为整数，再相乘，将结果转换为16进制
3. 将4个独立的运算结果之间以"/"分割，再整合为一个字符串（密文）
4. 如果搞不清楚这个运算过程，下面有一个利用Python实现的加密示例：

```
# 明文固定4位，密钥为你的学号后4位
encrypt = lambda clear, key: "".join([hex(ord(clear[i]) * ord(key[i]))[2:] + '/'
for i in range(4)])[:-1]
```

你可能会用到的Git命令：

- `git init`：在当前目录初始化Git仓库
- `git add .` ：将当前目录下所有文件加入暂存区
- `git commit -m "xxx"`：提交暂存区的更改，并附加评论“xxx”
- `git push`：将本地仓库的更改推送到远程仓库
- `git clone https://xxxx`：将[URL]的仓库克隆到本地

附加题

- 请简单解释一下这个函数是如何只用一行代码就完成工作的。它们还能再简化吗？

如果你实在无法通过Git将代码上传到你的个人仓库，请于2024年10月18日23:59之前打包发送到jeffreyqjfing@gmail.com，邮件标题格式为 **姓名-学号-部门-技术小组纳新。**

参考资料

- Git
- Learn Git Branching
- ZJU Git
- Python入门

1. 集装箱? - Docker

Docker是一种容器化技术，可以快速构建、测试和部署应用程序，相比于传统的运维，使用Docker可以解决“为什么在我的机器上跑得起来，在你的机器上不行呢？”等等一系列问题，同时也能够充分利用系统资源。当有一个DockerFile的时候，甚至不用去了解项目就可以部署它，那么，这么好用的东西...

你的任务

- 安装Docker(Windows, Linux都可)
- 使用Docker启动一个centos，并截图记录
- 尝试使用Docker挂载卷，并截图记录
- 使用Docker启动一个MySQL服务，并映射端口，截图记录

一些可能需要用到的命令：

- `docker ps`： 列出Docker容器
- `docker run`： 创建并启动一个新的容器
- `docker start/stop/restart`： 启动/停止/重启容器

相关的教程可以参考： Docker教程

2. 我来造轮子 - Bash

编写一个bash脚本`search_file.sh`，要求：

- 要求用户输入要搜索的扩展名和要搜索的目录。当用户输入q，程序退出

> 如：

```
Please input file extension (q to quit):
Please input directory to search (q to quit):
```

- 根据用户输入的扩展名和目录，递归搜索所有扩展名匹配的文件，将其路径输出到屏幕（绝对/相对路径都可以）。
- 若文件路径是绝对路径，且以家目录开头，将整个家目录替换成~

> 注意：环境变量$HOME就是家目录的绝对路径。在终端中，可以用echo $HOME查看家目录的绝对路径。在bash脚本中，同样使用$HOME来指代家目录的绝对路径。

```
$ echo $HOME
/c/Users/<user name>      # git bash
/home/<user name>         # linux
```

> 示例：将以家目录开头的目录替换为~开头：

```
/home/fracher/code/test.c   ->   ~/code/test.c
```

- 输出一共有多少个文件
- 从第一步开始重复，直到用户输入q

附加要求

- 使输出更加好看，可以在echo时为文本添加颜色。
- 此脚本不符合Unix哲学，即一件工具只做一件事。要求使用参数来控制该脚本的行为。仅当参数为`--interactive`时，才进行上面的轮询，否则仅根据参数进行一次搜索。如：

  ```
  $ ./search_file.sh --extension docx --directory ./Documents
  ./Documents/file1.docx
  ./Documents/file2.docx
  Total 2 files.
  $ ./search_file.sh --interactive
  Please input file extension (q to quit): ...
  Please input directory to search (q to quit): ...
  ...
  $ ./search_file.sh --help                # 输出帮助信息
  search_file.sh is a script to search files in a particular directory.
  USAGE:
  search_file.sh [OPTIONS]
  ... ... ...
  ```

  注：帮助信息的样例可以参考系统中的命令的帮助信息，如：

```
$ ls --help
$ cp --help
```

简单写一点即可，不必过于详细。

可以参考的项目

- Fracher的 配置文件管理器
- i3lock-fancy

另编

Vim之父 Bram Moolenaar 于2023年8月3日去世。让我们深切悼念这位开发者，他为自己的人生写下了最后一句 ":wq"。

3. 网页小实践 - Frontend

请制作一个本地网页，实现一些简单的效果。要求如下：

> 功能要求：

- 这个网页有一个文本输入框和一个按钮，按钮上写着“生成”字样
- 在文本框中输入一个数字，按下“生成”按钮，下方会生成对应数量的小方块，小方块从左至右，从上至下以一定间距排列
- 左键点击小方块可以消除该小方块，其他小方块顺位排列填补空缺
- 在已经有生成小方块的情况下点击“生成”按钮，会在原有基础上增加相应数量的小方块

> 外观样式要求：

- 没有硬性要求，当然我们鼓励你按自己想法装饰网页

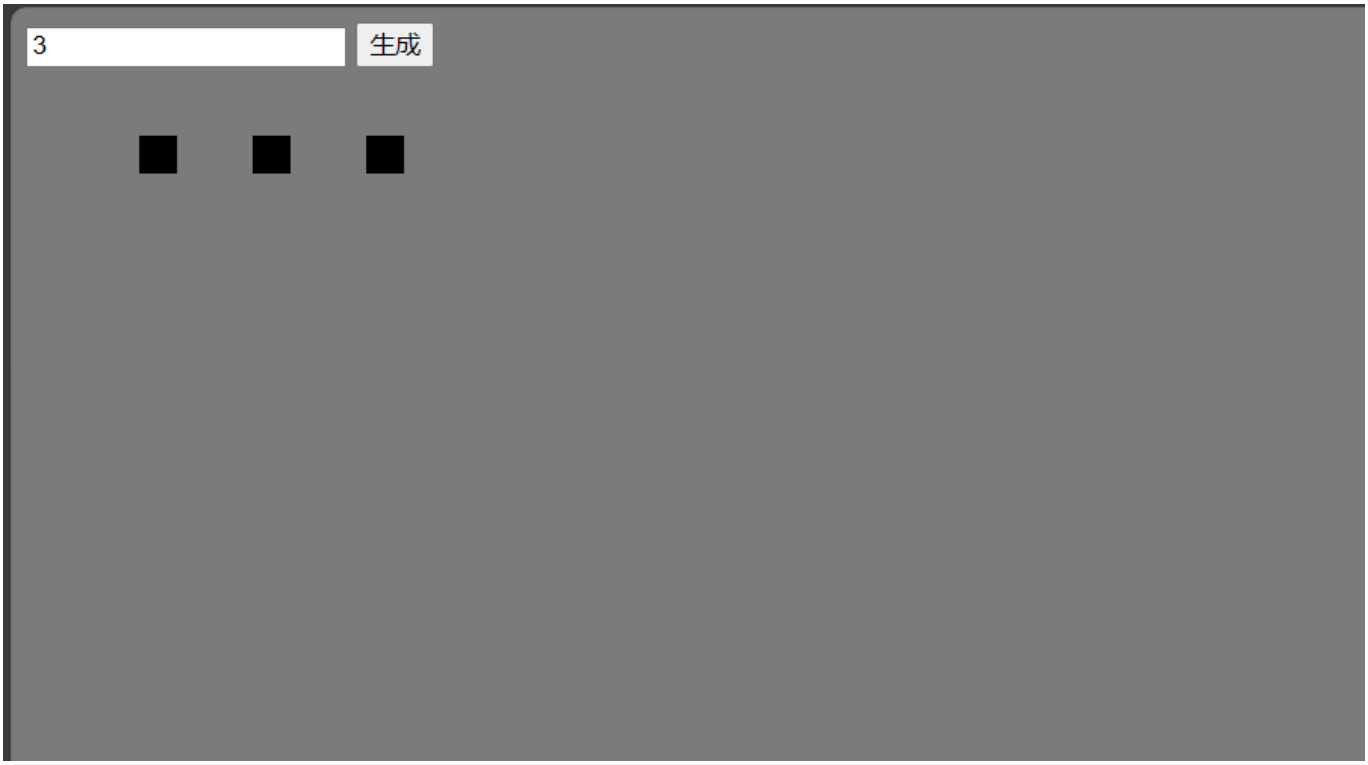
> 文件组织要求：

- 采用外置的JavaScript脚本和CSS层叠样式表
- 最终应该提交一个文件夹，内含一个.html文件，.css文件，一个.js文件。

样例：

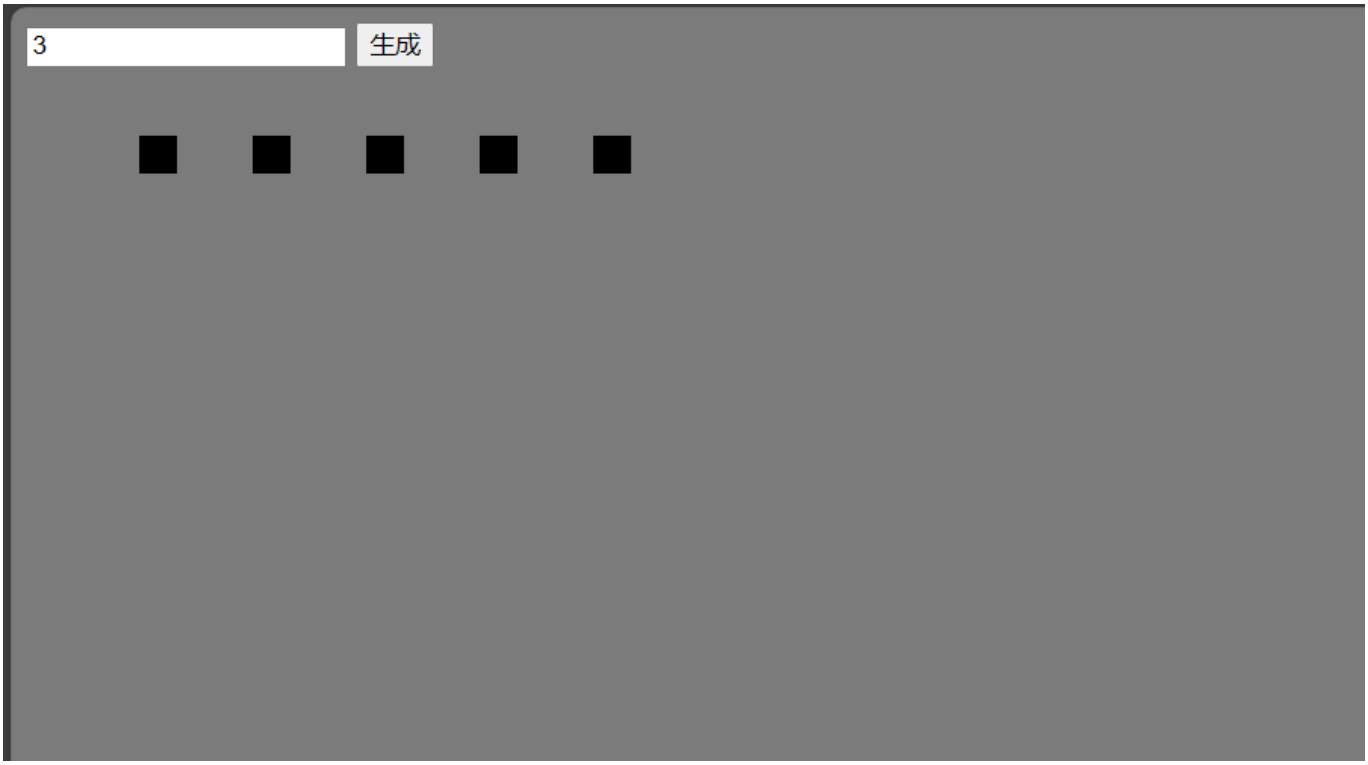
1. 网页刚刚加载

1. 在文本框中输入"3"并点击"生成"按钮



1. 左键单击最左侧的小方块

1. 再次点击"生成"按钮



一些可能有助于完成项目的小提示：

- 小方块的长相和排列方式可以通过修改HTML元素"li"的样式获得
- 左键小方块使其消失的功能，需要在生成小方块时对每个小方块注册"onclick"事件
- 你可能需要学习一些基础的HTML、JS、CSS的语法，可以考虑参考这个网站上的教程

4. 204什么情况？- Backend

根据传统，老东西们会在204门口的电闸上贴上一个便利贴，写着“都别动！动会死！”。因为协会在自闭间有三台服务器，其中一台服务器上运行着EVA File、EVA Wiki等各种重要的网络服务。如果断电了，协会的部分网络服务无法正常运行，也有可能会损伤服务器硬件。但是，如果碰到了突发情况，又没有人在204，如何方便得知204服务器运行的状态呢？

我们可以在另一台较可靠的服务器（如云服务器）上搭建一个简单的服务端，并让204的服务器定时向云服务器发送请求。如果云服务器那端验证是204服务器发送过来的请求，那么就更新时间戳。如果过了一定时间（如：5分钟）没有来自204服务器的请求，那么就说明204的服务器可能出了问题，需要去检查情况了。

现在，就请你根据以上描述搭建一个简单的云端监测服务吧！

你的任务

在本地搭建一个服务端，并完成以下需求：

需求	需求描述	HTTP路由
检验网络连通性	服务端在接收到客户端发来请求后返回成功提示	`[GET]` /ping
更新时间戳	服务端在接收到待检测服务器发送的请求后返回更新情况	`[POST]`/check
查询更新时间	服务端在接收到客户端发送的服务器名称后返回上一次更新时间、以及是否失联	`[POST]`/status

- **通用格式：** 所有的返回值都需要符合以下格式：

```
{
    "code": 0,        // 错误码，非 0 表示失败
    "msg": "",        // 错误描述
    "data":
      {
          ...         // 数据主体中的具体内容
      },
}
```

在后续的返回值格式描述中，我们将**略去通用部分**，只描述数据主体（即 data 部分）的格式。

- `[GET]`/ping **返回值：**

```
{
    "msg": "pong"
}
```

- `[POST]`/check **请求体：**

```
{
    "source": "ZJUEVA204", // 待监测服务器名称，你可以自拟
```

```
}
```

返回体：

```
{
    "isChecked": true,      // 是否更新时间戳
}
```

可能的错误信息：

1. 服务器未在监测列表内

```
{
    "code": 100,
    "msg": "Server not authorized",
    "data":
      {
          "isChecked": false,
          ...
      },
}
```

2. 请求未提供服务器名称

```
{
    "code": 101,
    "msg": "Should have a server name",
    ...
}
```

- [POST]/status **请求体：**

```
{
    "server": "ZJUEVA204", // 待检查服务器名称，你可以自拟
}
```

返回体：

```
{
    "lastTime": "1984/10/01 11:45:14", // 按照"yyyy/mm/dd hh:mm:ss"的格式返回
    "isDisconnected": true,            // 是否失联
```

```
    "hitokoto": "你好呀，祝你满绩每一天"  // （选做）一言 API 返回的语句
}
```

不要求错误处理。

你可以适当减少监测间隔时长以方便调试。

附加题

- 将服务器名称链接数据库，形式不限，数据自拟即可。
- 当访问`/status`路由时从一言API拉取语句数据再发送给客户端，你可能要再自拟一些错误情况处理。

一些需要注意的事

本题不做语言限制，你可以用任何你喜欢的编程语言完成这一题。如果你没有头绪的话，可以试试 Go的Gin框架，或者Nodejs的express包，Python与C#也是一个不错的选择。~~C++也不是不行~~ 服务只需要在本地运行，不必上传至服务器，提交时提交源码即可。

参考资料

- Nodejs
- Express
- The Little Go Book
- Gin
- What is Hypertext Transfer Protocol (HTTP)
- HTTP Tutorial

5.原来是拟合 - Pytorch

在大家写实验报告的时候不可避免的会遇到拟合实验数据的曲线，小钱同学由于不会使用Excel拟合曲线，导致每次实验报告都被打低分。他下定决心要找出能最好拟合实验数据的曲线，于是他想到了神经网络...

你的任务

现在，请你使用pytorch设计并训练一个神经网络，可以拟合实验数据的一系列点，例如符合 `y = 3x ± 0.5` 分布的一系列散列的点，要求：

- 需要提供训练代码和神经网络的模型结构代码，并将训练完成的模型保存为xx.pt的文件附在一起提交。
- 需要提供神经网络在你所选定的一个线性或非线性的函数上的最终训练效果的图片，即数据点和曲线绘制在一张图上（建议使用matplotlib等绘图工具）。

一些可能有助于完成项目的小提示：

- 本次任务并不需要用到卷积（Convolution），池化（Pooling）等操作，需要的仅仅是线性层（Linear）和激活函数。
- pytorch简单入门可以参考这个文章： https://pytorch.ac.cn/tutorials/beginner/basics/intro.html

附加题

1. 训练一个卷积神经网络，数据集为MINST数据集（需自行下载），识别的准确率需要达到80%以上，提供内容同上，将原来训练效果的图片改为准确率的截图。
2. 试试不依赖神经网络, 编写 Python 代码用传统的统计方法拟合实验数据, 并讨论一下这种方法与使用神经网络进行拟合的区别。

6.网络蜘蛛侠 - spider

小钱同学是一个医学专业的小e，很多时候需要搜索文献，但是人工手动搜索文献效率太低了，小钱同学就想着能不能通过一个自动程序来将相关的搜索结果以 **文献标题：文献链接** 保存为为一个.txt的文件中。

你的任务

- 将Nature上的文章的文献标题和文献链接保存下来，比如

```
Fratricide-resistant CD7-CAR T cells in T-ALL :
https://www.nature.com/articles/s41591-024-03228-8

A CAR enhancer increases the activity and persistence of CAR T cells :
https://www.nature.com/articles/s41587-024-02339-4
```

这样两行，搜索的关键词可以自定义，如`CAR-T`。

- 你所需要提交的包括一个.txt文件和相应的源代码（建议使用python）。

一些可能有助于完成项目的小提示：

- 某些网站可能存在反爬虫的策略，为了更好的得到数据，可以使用chromedriver + python中的`selenium`包 来模仿用户行为，selenium相关教程可以参考这个网站。
- Nature的web源代码可能比较多，可以在查看网页源代码的同时使用ctrl + F 来搜索自己感兴趣的的地方（比如网址）。
- 在page source 中获取自己感兴趣的内容（文献标题与文献链接），可以使用 `BeautifulSoup` 包来解析，也可以使用正则表达式来匹配，BeautifulSoup相关教程可以参考这个网站.

附加题

1. 实现从一页相关的搜索结果拓展到10页，将10页中的文献标题：文献链接内容都下载下来。
2. 实现不仅仅只获取文献链接，将文献链接所指的PDF也下载到本地。

7. 今天你加密了吗？ - NGINX

不少 Web 项目为了降低代码复杂度或是便于维护, 会避免在项目源码中支持 HTTPS (SSL/TLS 加密)。为了安全地提供服务，我们一般会使用带有 HTTPS 的反向代理。 你的任务是为知名的 Web 服务器/反向代理软件 NGINX 编写配置文件片段(具体地说，是一个 server 块)，让 NGINX 将访问本机 443 端口的 HTTPS 流量反向代理到 http://localhost:12345 请提交相应的配置文件片段。

附加题：

1. 为你的反向代理配置 HSTS，并将访问本机 80 端口的 HTTP 请求重定向到 443 端口上的 HTTPS 服务器
2. 修改配置文件，禁用已经并不安全的加密/摘要算法

提示：

1. 你不需要提供 SSL 证书，对于测试用途，你可以选择创建一张自签名证书，也可以使用 let's encrypt 等服务(假如你有一个域名)
2. 测试时被反代的域名可以自行选择(比如 zdbk.zju.edu.cn)，但你提交的配置文件中应当正确配置为题目中提到的地址

可能有帮助的教程:
菜鸟教程
官方文档

8.看不懂的字符就是让人觉得很奇怪啦 - misc

王同学在协会自闭间自闭的时候，屏幕上突然出现了一行字母：

```
RVZBe1czMWMwbUVfVG9fekp1RVY0X1QzY2hAPz8/Pz99Ck1ENTo5NDY4NkM1N0U1MTIzNDkyOEIzNDQ5MjRGOTkyRDlDOQ==
```

王同学感到好疑惑，但又感觉结尾的`=`有点眼熟，好像他学过的某种算法。

你的任务：

通过对上面这段字符作适当的处理，你可以得到一段有意义的字符，大致如`ZJUEVA{xxxxxxx}`的形式。请确保最后的结果不包含`?`，处理方法不限，如果你在解题过程中使用了任何工具，请将使用的工具（网站、代码）放到你的解题文档当中（如果是网站的话请截图解题过程，如果是代码的话直接贴到文档里即可）。

提示：

1. 注意结尾的`=`，这是一个很明显的标志。
2. 我会推荐你使用python来解决这一道题。

X. 喜欢您来 - And You

感谢你看到这里！

如果你有什么觉得足够有趣，并且实现难度等同或超过上述题目的项目，我们也非常欢迎你自行提交项目仓库链接与项目说明。

期待你的表现！