OS Lab Mini Project Report

Name: Sahil Kolte

Roll no.: IMT2023066

Title

Design and Development of Course Registration Portal (Academia)

Abstract

The project aims to develop a socket-based, multi-user Course Registration Portal for an academic institution. This system enables Students, Faculty, and Admin users to perform role-specific operations such as course management, enrollment, user management, and secure access using a login system. The server maintains the entire academic database and handles concurrent client requests using system-level constructs like file management, file locking, and semaphores, ensuring data consistency and synchronization.

Objectives

- To implement a role-based academic portal.
- To ensure secure login for Admin, Faculty, and Student roles.
- To provide functionalities for course registration, enrollment, and user management.
- To maintain data consistency using file locking and semaphores.
- To handle concurrent client connections using socket programming.

System Architecture

The system follows a **client-server architecture**:

- **Server**:
  - Maintains all data files (students, faculty, courses, enrollments).
  - Handles multiple client connections using sockets and fork().
  - Uses semaphores and file locks for synchronized access.
  - Manages different user roles and performs operations based on user input.
- **Client**:
  - Connects to the server using sockets.

    - Sends login credentials and menu choices.
    - Receives and displays responses from the server.

Key Functionalities

Admin

- Login with secure credentials.
- Add Student.
- Add Faculty.
- Activate/Deactivate Students.
- Update Student/Faculty Details.

Faculty

- Login with secure credentials.
- Add new Course (with seat limits).
- Remove Course.
- View Enrollments.
- Change Password.

Student

- Login with secure credentials.
- Enroll in a Course (if seats available).
- Unenroll from Course.
- View enrolled Courses.
- Change Password.

Technical Details

Socket Programming

- TCP Sockets implemented in both server.c and client.c.
- Server listens for multiple connections on a defined port using bind(), listen(), accept().

- Each client request is handled via fork() to create a new process.

File Management

- Data stored in:
    - student.txt (format: id name password status)
    - faculty.txt (format: id name password)
    - course.txt (format: id name faculty)
    - enrollment.txt (format: course_id student_id)

File Locking

- fcntl()-based file locks used.
    - **Read lock** for viewing course information.
    - **Write lock** for enrolling/unenrolling to prevent data races.

Semaphores

- Used to synchronize access when multiple processes attempt to read/write to shared resources.

Concurrency Handling

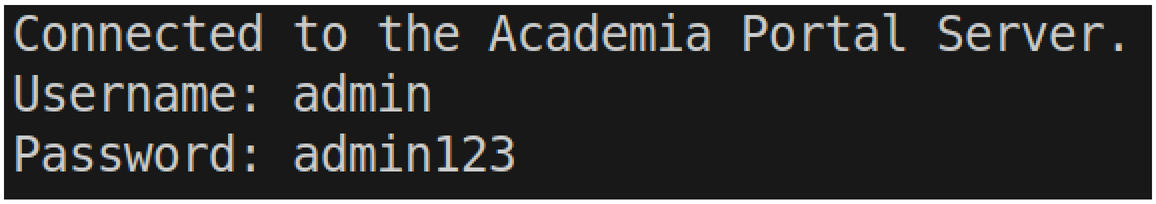
- **Process-based concurrency**: Each client handled in a separate child process.
- **Data Integrity**:
    - Locks used for critical sections.
    - Semaphores prevent simultaneous conflicting access.

Steps to run project:

1) Execute: make all
2) Execute: ./server
3) On a new terminal, execute: ./client

Screenshots of program output

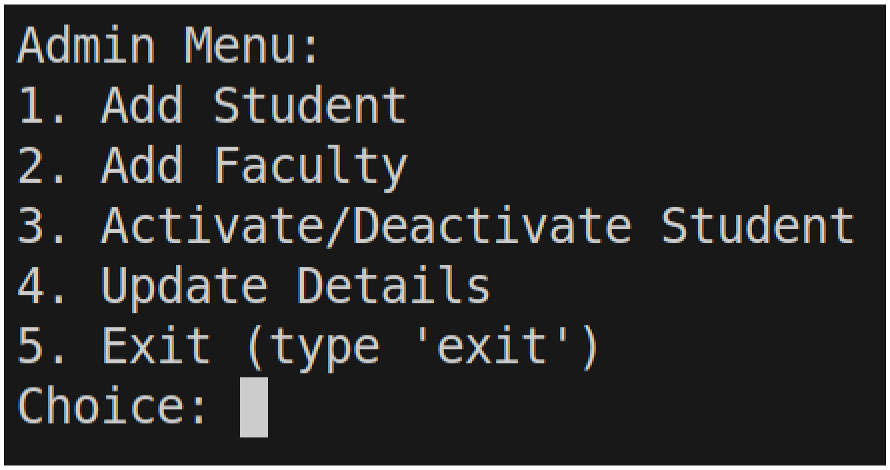
Admin



```
Connected to the Academia Portal Server.
Username: admin
Password: admin123
```

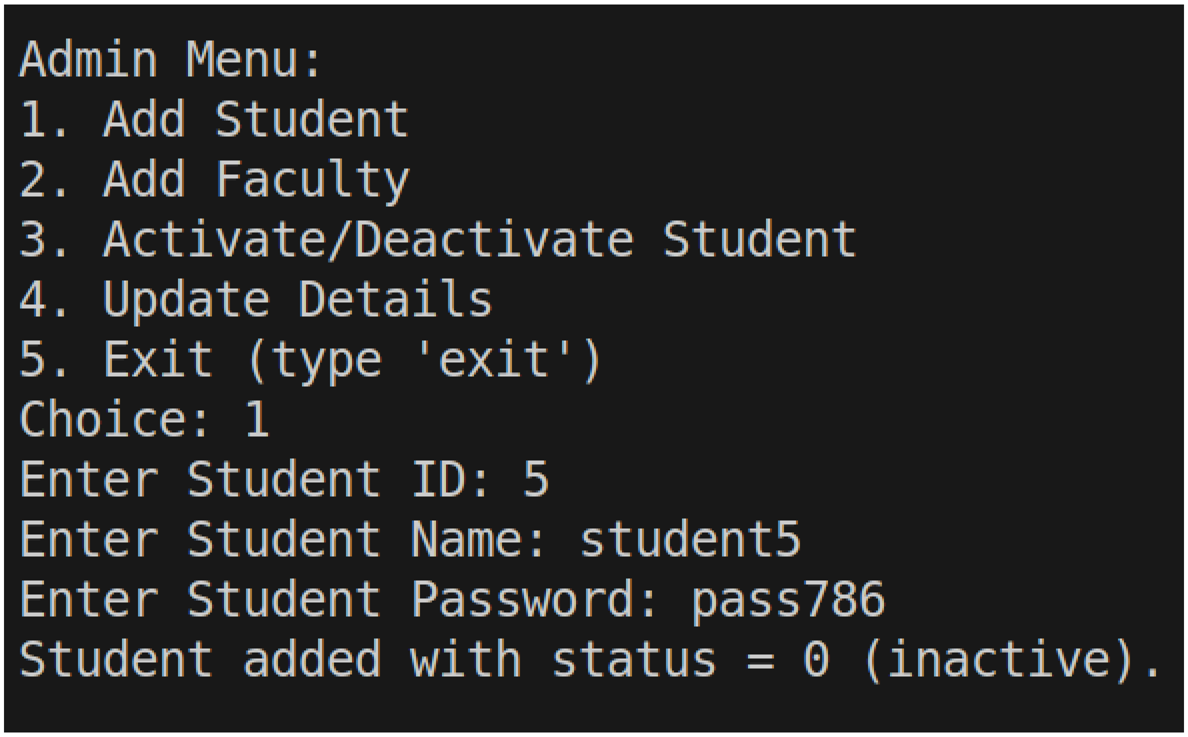
Based on username and password, the program decides whether the user is admin, student or faculty.

Admin menu:



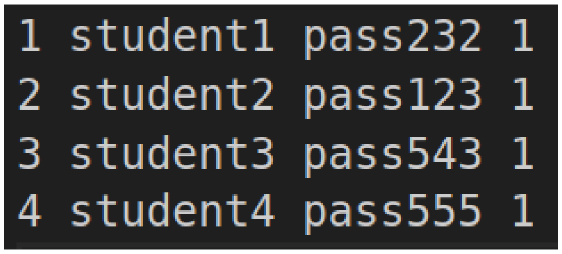
```
Admin Menu:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Details
5. Exit (type 'exit')
Choice:
```

Adding student:

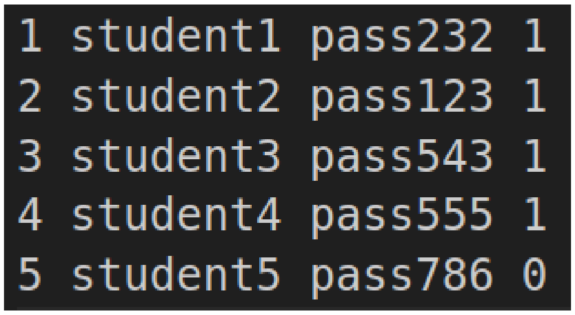


```
Admin Menu:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Details
5. Exit (type 'exit')
Choice: 1
Enter Student ID: 5
Enter Student Name: student5
Enter Student Password: pass786
Student added with status = 0 (inactive).
```

Before and after in student.txt:

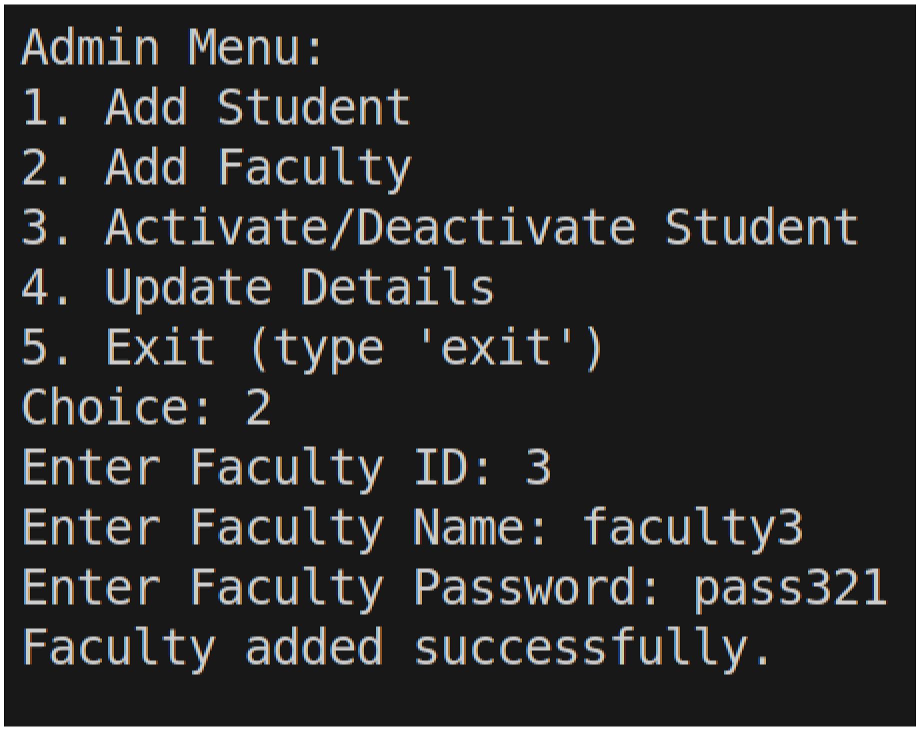


```
1 student1 pass232 1
2 student2 pass123 1
3 student3 pass543 1
4 student4 pass555 1
```



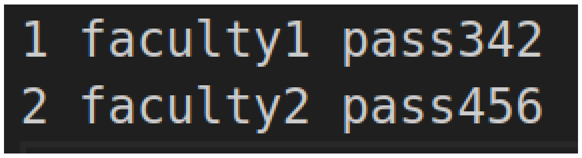
```
1 student1 pass232 1
2 student2 pass123 1
3 student3 pass543 1
4 student4 pass555 1
5 student5 pass786 0
```

Adding faculty:

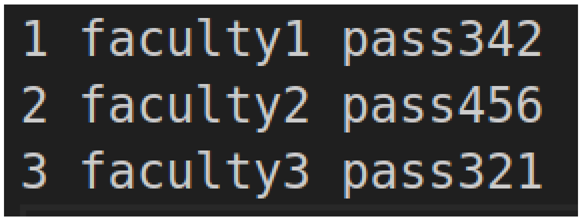


```
Admin Menu:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Details
5. Exit (type 'exit')
Choice: 2
Enter Faculty ID: 3
Enter Faculty Name: faculty3
Enter Faculty Password: pass321
Faculty added successfully.
```

Before and after in faculty.txt:

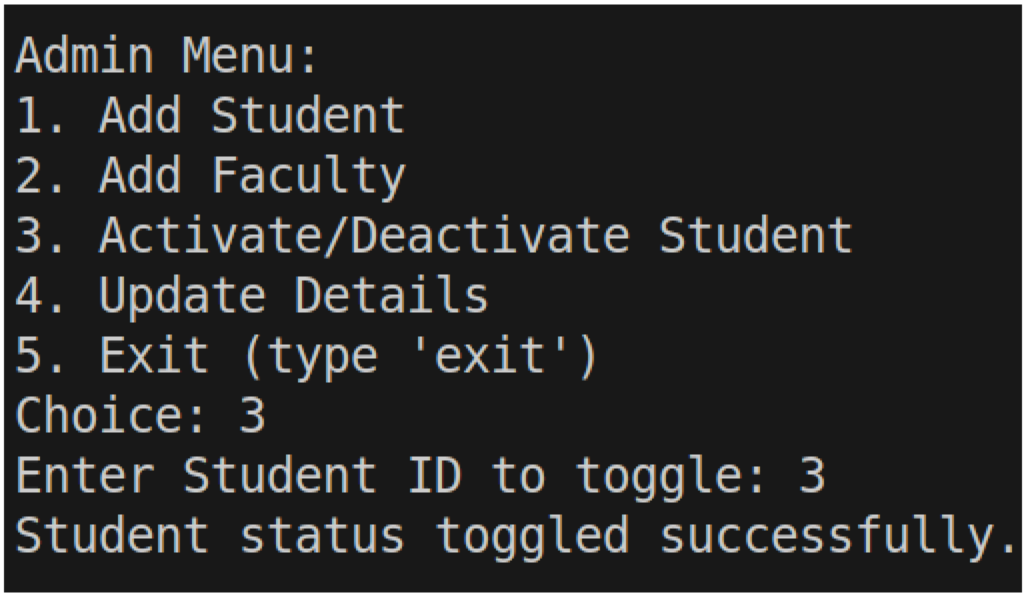


```
1 faculty1 pass342
2 faculty2 pass456
```

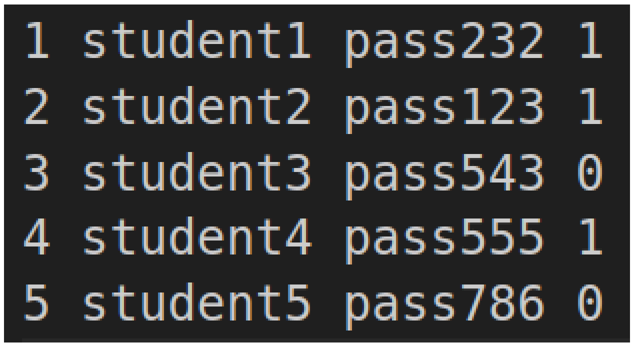


```
1 faculty1 pass342
2 faculty2 pass456
3 faculty3 pass321
```

Toggle student status:

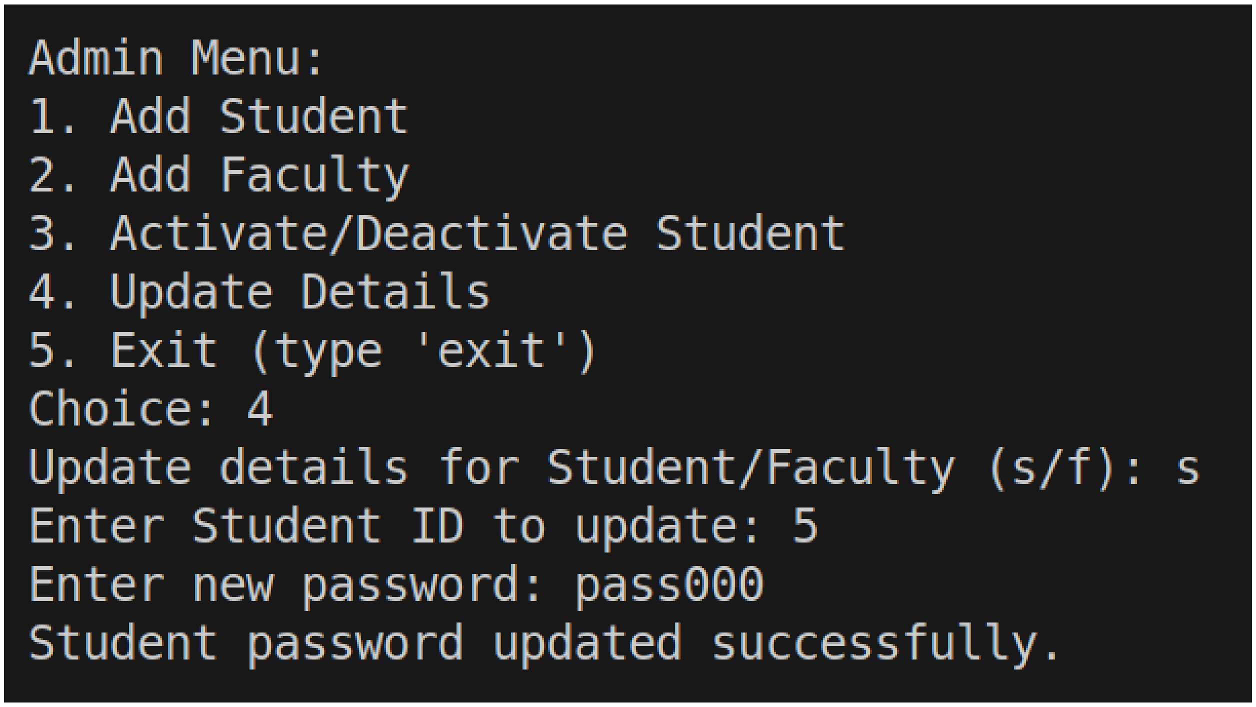


```
Admin Menu:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Details
5. Exit (type 'exit')
Choice: 3
Enter Student ID to toggle: 3
Student status toggled successfully.
```

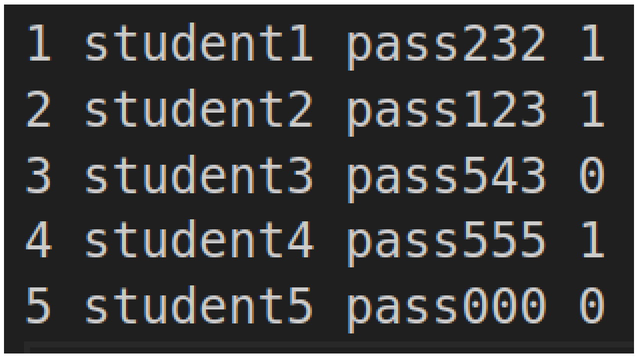


```
1 student1 pass232 1
2 student2 pass123 1
3 student3 pass543 0
4 student4 pass555 1
5 student5 pass786 0
```

Update details:



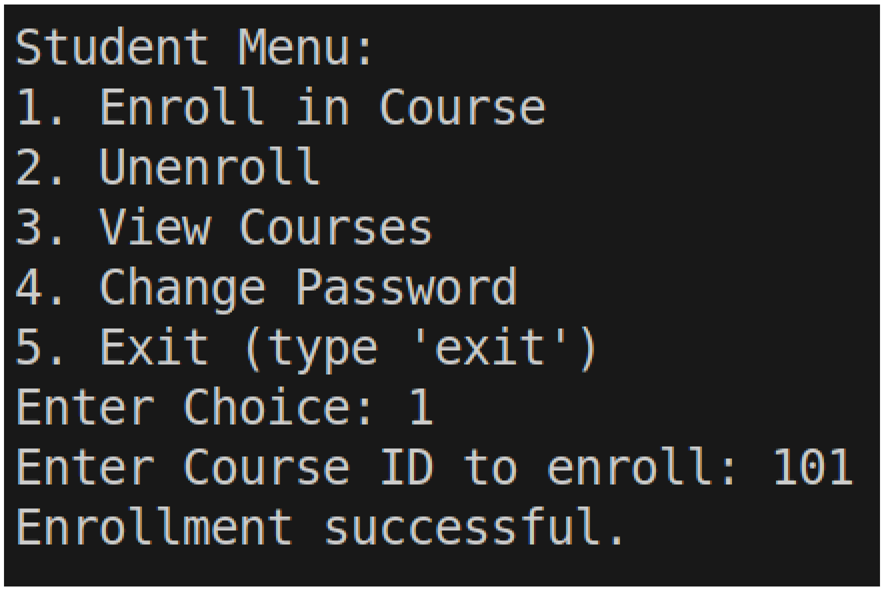
```
Admin Menu:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Details
5. Exit (type 'exit')
Choice: 4
Update details for Student/Faculty (s/f): s
Enter Student ID to update: 5
Enter new password: pass000
Student password updated successfully.
```



```
1 student1 pass232 1
2 student2 pass123 1
3 student3 pass543 0
4 student4 pass555 1
5 student5 pass000 0
```

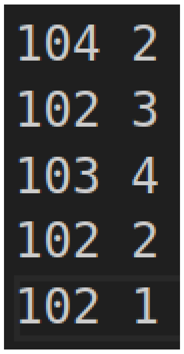
Student

Enroll in course:

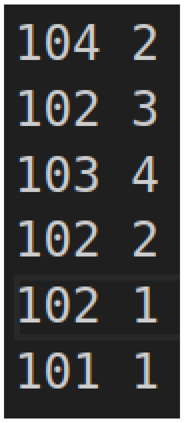


```
Student Menu:
1. Enroll in Course
2. Unenroll
3. View Courses
4. Change Password
5. Exit (type 'exit')
Enter Choice: 1
Enter Course ID to enroll: 101
Enrollment successful.
```

Before and after in enrollment.txt:

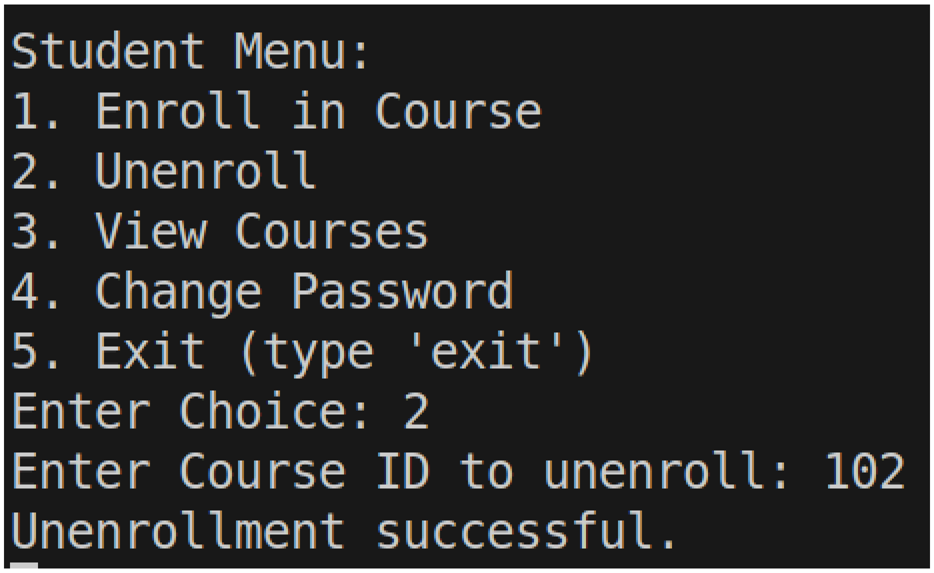


```
104 2
102 3
103 4
102 2
102 1
```



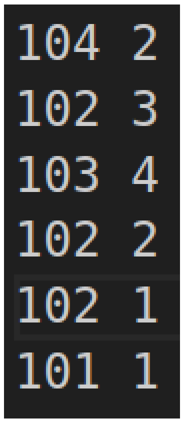
```
104 2
102 3
103 4
102 2
102 1
101 1
```

Unenroll from course:

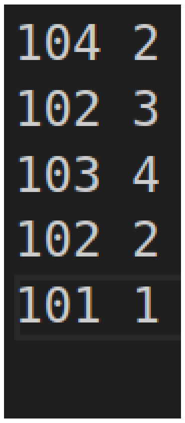


```
Student Menu:
1. Enroll in Course
2. Unenroll
3. View Courses
4. Change Password
5. Exit (type 'exit')
Enter Choice: 2
Enter Course ID to unenroll: 102
Unenrollment successful.
```

Before and after in enrollment.txt:

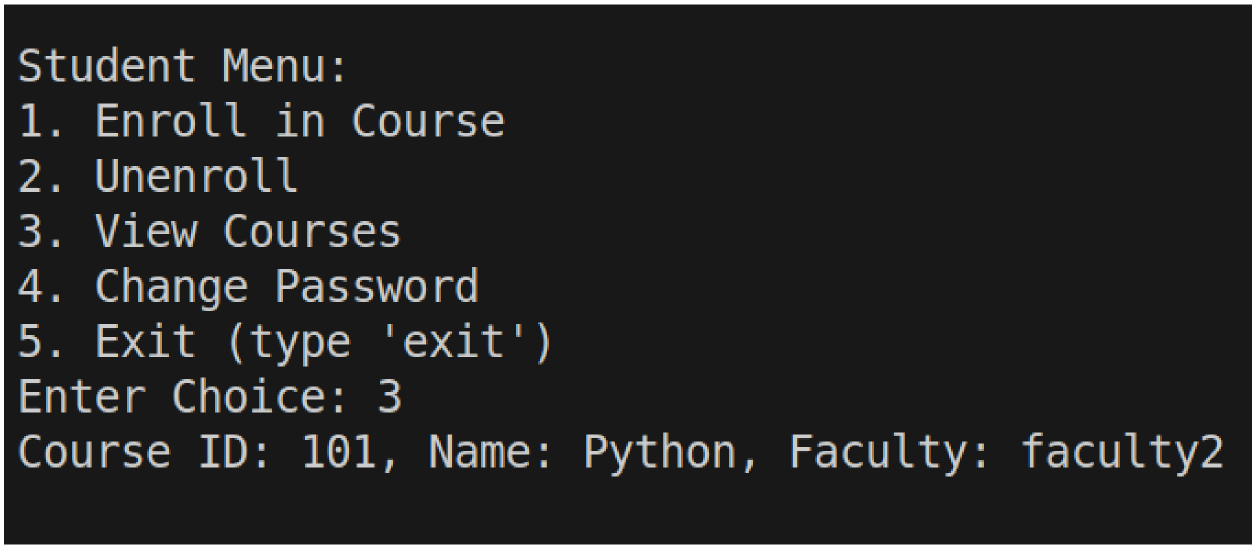


```
104 2
102 3
103 4
102 2
102 1
101 1
```



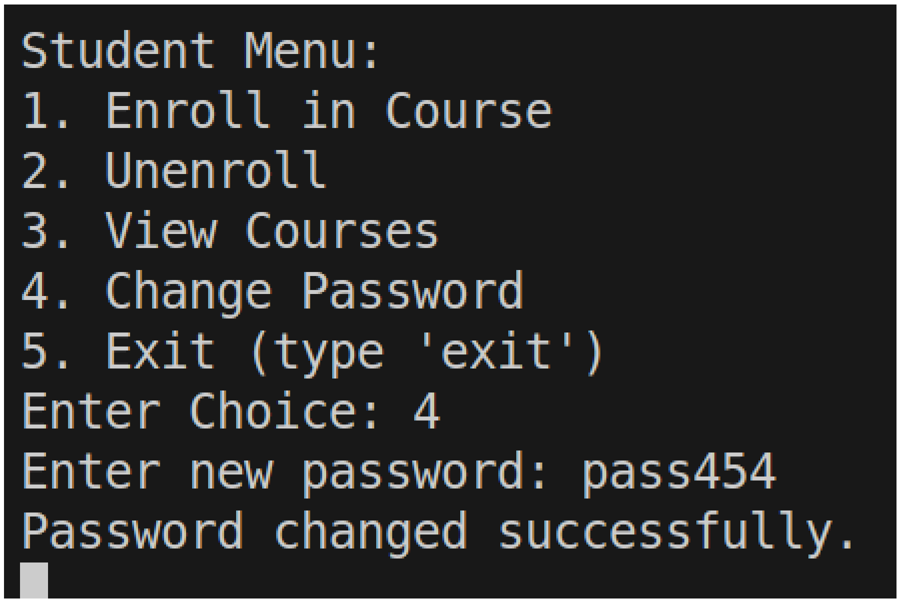
```
104 2
102 3
103 4
102 2
101 1
```

View Courses for currently logged in student:

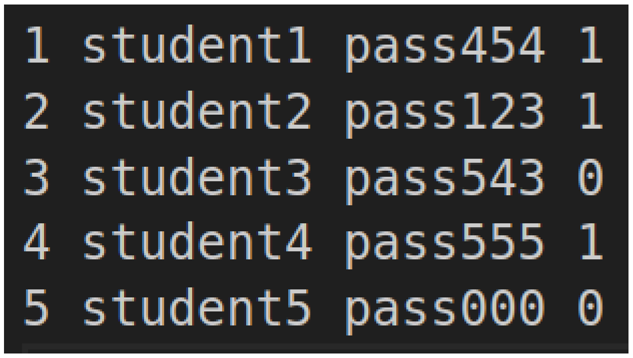


```
Student Menu:
1. Enroll in Course
2. Unenroll
3. View Courses
4. Change Password
5. Exit (type 'exit')
Enter Choice: 3
Course ID: 101, Name: Python, Faculty: faculty2
```

Change password for currently logged in student:



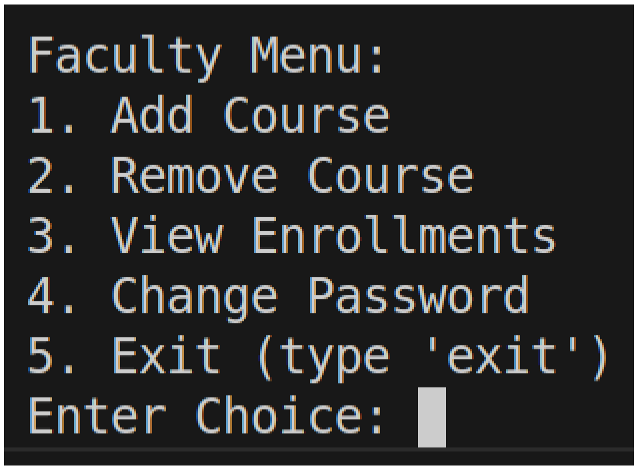
```
Student Menu:
1. Enroll in Course
2. Unenroll
3. View Courses
4. Change Password
5. Exit (type 'exit')
Enter Choice: 4
Enter new password: pass454
Password changed successfully.
```



```
1 student1 pass454 1
2 student2 pass123 1
3 student3 pass543 0
4 student4 pass555 1
5 student5 pass000 0
```

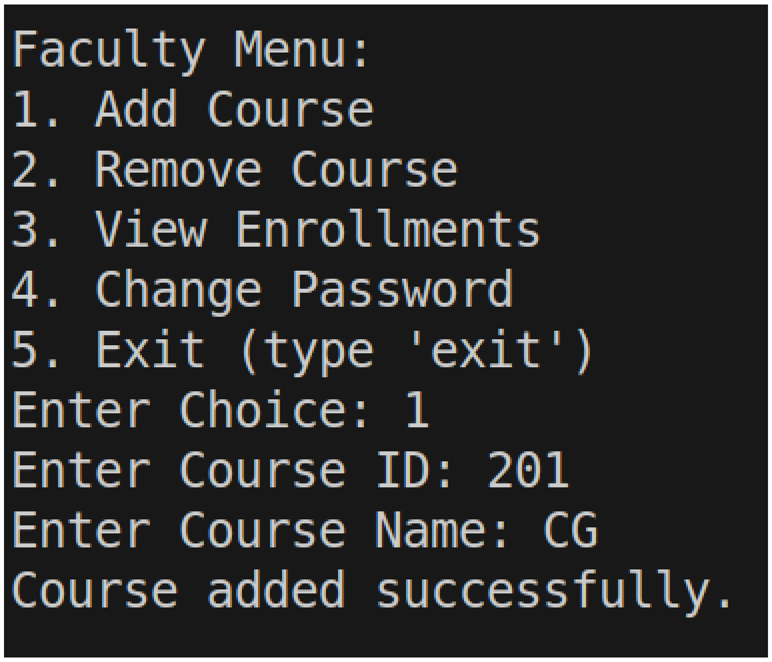
Faculty

Faculty menu:



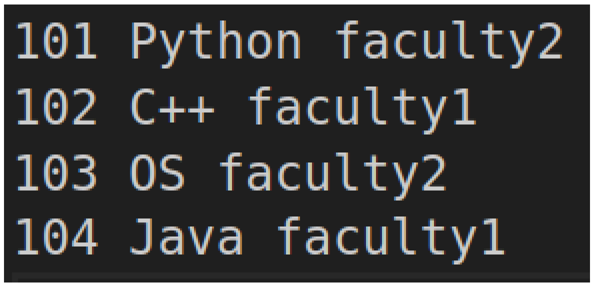
```
Faculty Menu:
1. Add Course
2. Remove Course
3. View Enrollments
4. Change Password
5. Exit (type 'exit')
Enter Choice: 
```

Add Course:

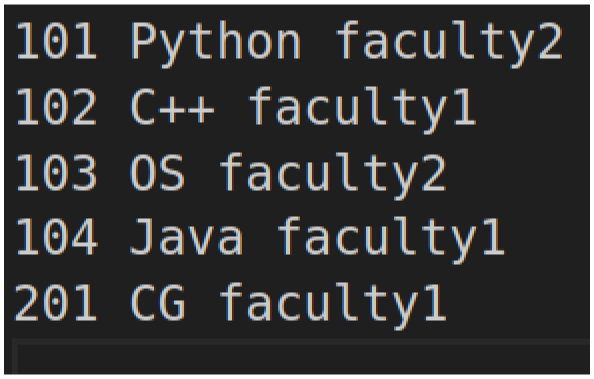


```
Faculty Menu:
1. Add Course
2. Remove Course
3. View Enrollments
4. Change Password
5. Exit (type 'exit')
Enter Choice: 1
Enter Course ID: 201
Enter Course Name: CG
Course added successfully.
```

Before and after in course.txt:

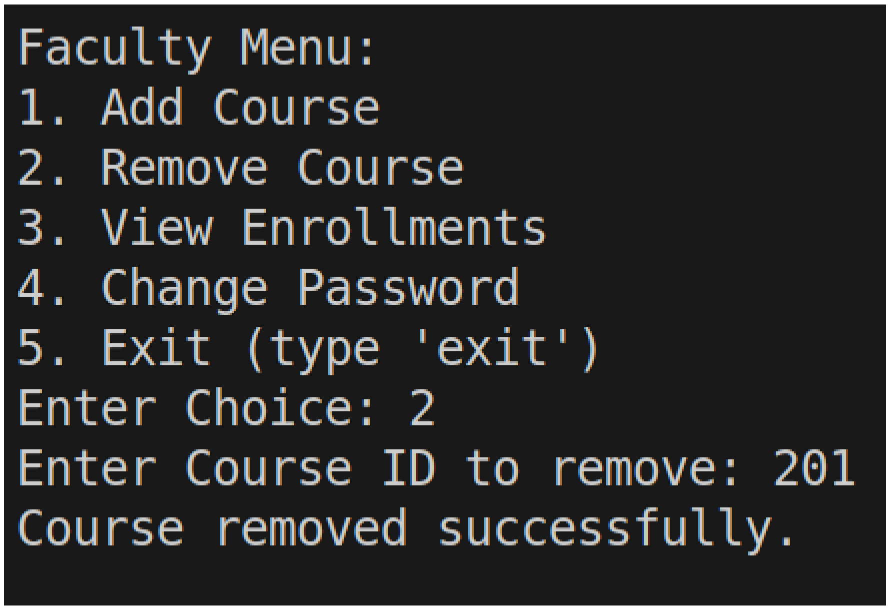


```
101 Python faculty2
102 C++ faculty1
103 OS faculty2
104 Java faculty1
```



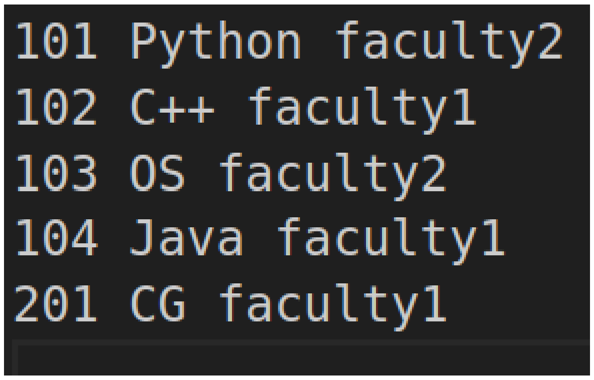
```
101 Python faculty2
102 C++ faculty1
103 OS faculty2
104 Java faculty1
201 CG faculty1
```

Remove course:

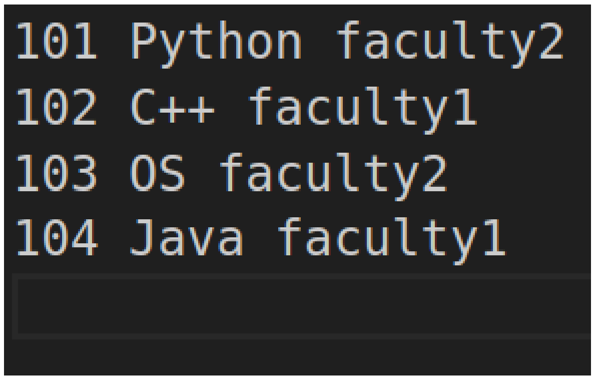


```
Faculty Menu:
1. Add Course
2. Remove Course
3. View Enrollments
4. Change Password
5. Exit (type 'exit')
Enter Choice: 2
Enter Course ID to remove: 201
Course removed successfully.
```

Before and after in course.txt:

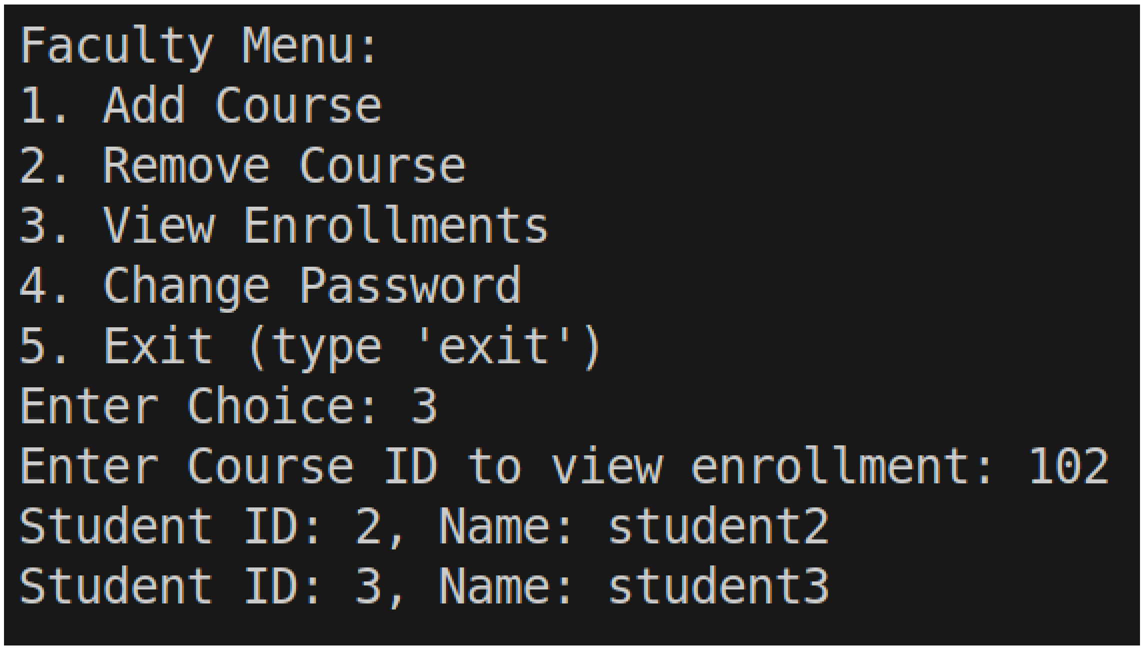


```
101 Python faculty2
102 C++ faculty1
103 OS faculty2
104 Java faculty1
201 CG faculty1
```

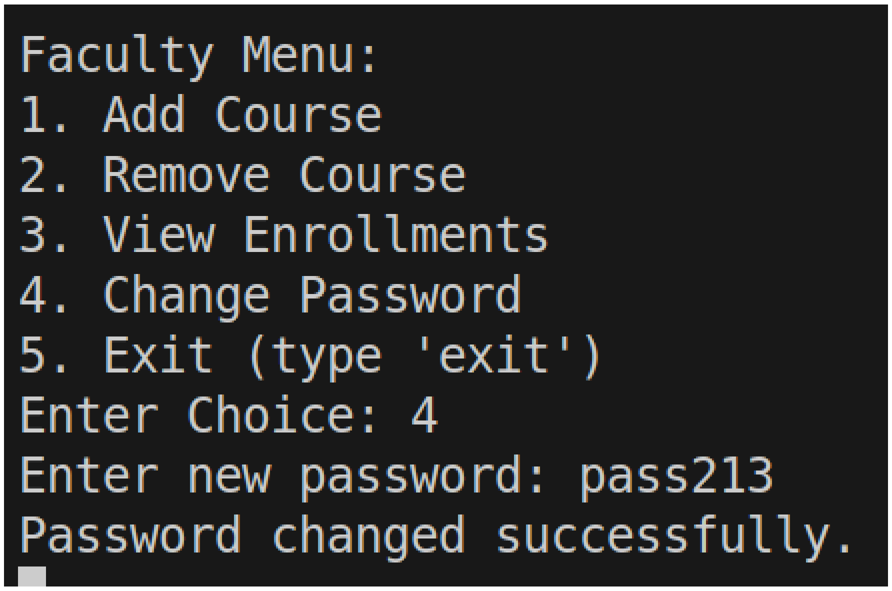


```
101 Python faculty2
102 C++ faculty1
103 OS faculty2
104 Java faculty1
```

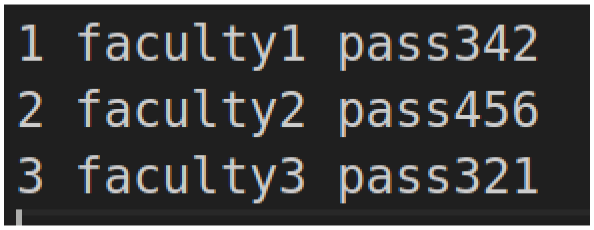
View enrollments for a specified course and provided by the logged in faculty:

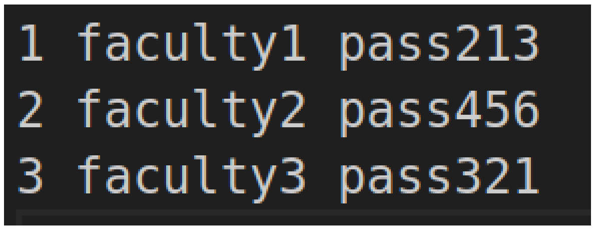

```
Faculty Menu:
1. Add Course
2. Remove Course
3. View Enrollments
4. Change Password
5. Exit (type 'exit')
Enter Choice: 3
Enter Course ID to view enrollment: 102
Student ID: 2, Name: student2
Student ID: 3, Name: student3
```

Change password:


```
Faculty Menu:
1. Add Course
2. Remove Course
3. View Enrollments
4. Change Password
5. Exit (type 'exit')
Enter Choice: 4
Enter new password: pass213
Password changed successfully.
```

Before and after in faculty.txt:


```
1 faculty1 pass342
2 faculty2 pass456
3 faculty3 pass321
```


```
1 faculty1 pass213
2 faculty2 pass456
3 faculty3 pass321
```