PAPER NAME	AUTHOR
Raghav Garg	Raghav Garg
WORD COUNT	CHARACTER COUNT
7421 Words	53862 Characters
PAGE COUNT	FILE SIZE
47 Pages	188.4KB
SUBMISSION DATE	REPORT DATE
Nov 27, 2023 4:00 PM GMT+5:30	Nov 27, 2023 4:01 PM GMT+5:30

● 7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 5% Internet database
- Crossref database
- 4% Submitted Works database
- 0% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 14 words)
- Cited material

Project Report

CollegeTheka

Manan(E22CSEU0527)
Raghav Garg(E22CSEU0521)
Kislaya(E22CSEU0541)

A report submitted in part fulfilment of the degree of

BTech in Computer Science

Supervisor: Dr. Debajyoti Ghosh

Department of Computer Science
Bennett University, Greater Noida

November 17, 2023

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 7419

Student Name: Manan(E22CSEU0527)

Raghav Garg(E22CSEU0521)

Kislaya(E22CSEU041)

Date of Submission:17 November 2023

Signature:

Abstract

The CollegeTheka project stands as a groundbreaking solution tailored to transform and streamline administrative processes within educational institutions in India. This integrated system is intricately designed to enhance the efficiency and effectiveness of various administrative tasks conducted in a college environment. It encompasses student enrollment, academic tracking, grade management, and the handling of academic dues.

The CollegeTheka system offers a user-friendly interface catering to administrators, teachers, and students, fostering seamless communication and collaboration among these crucial stakeholders. Its key features include robust student record management, comprehensive course management, real-time academic tracking, examination management, and the automated generation of detailed academic reports. The system places a strong emphasis on data security and integrity, incorporating a secure authentication system.

Implemented in Virtual Studio Code, the technological backbone of CollegeTheka is fortified with C++ and Data Structures and Algorithms (DSA) for programmatic functionalities. The robust back-end system, fortified with encryption and role-based access control, serves as the foundation for ensuring the confidentiality and security of sensitive information.

By harnessing modern technologies, CollegeTheka significantly reduces manual efforts, minimizes errors, and enhances overall productivity. The system's implementation is poised to augment decision-making processes through real-time data analytics and reporting capabilities. This project stands at the forefront of the evolution in college management systems in India, aiming to usher in an era where administrative tasks are seamlessly automated, enabling educational institutions to dedicate more resources to providing quality education and fostering a conducive learning environment.

The academic environment is dynamic, with myriad challenges ranging from enrollment procedures to academics tracking and examination management. CollegeTheka addresses these complexities by providing an integrated platform that automates routine tasks. The robust student record management feature ensures a centralized repository of student information, facilitating efficient enrollment processes. This leads to a reduction in paperwork, faster processing times, and a more streamlined admission experience.

Course management is a cornerstone of the CollegeTheka system, enabling administrators to effortlessly organize and update course offerings. This feature not only provides clarity to students regarding available courses but also aids in resource allocation and scheduling. The academic tracking module is designed to monitor student progress in real-time, allowing for timely interventions and support. It enables teachers and administrators to identify academic trends and implement strategies for improvement.

The examination management feature ensures a seamless and organized approach to conducting exams. From scheduling to result processing, this module automates various aspects of the examination process, reducing the burden on administrative staff

and minimizing the scope for errors. Automated generation of academic reports enhances transparency and provides stakeholders with detailed insights into student performance.

The secure authentication system is integral to the CollegeTheka project, safeguarding the confidentiality and integrity of sensitive information. Role-based access control ensures that only authorized personnel can access specific data, adding an additional layer of security. Encryption protocols are employed to protect data during transmission and storage, aligning the system with modern data protection standards.

The technological underpinning of CollegeTheka, implemented in Virtual Studio Code, showcases a commitment to utilizing contemporary tools for software development. The choice of C++ and DSA highlights a focus on efficiency and performance, crucial aspects in handling large datasets and complex algorithms inherent in a college management system.

The CollegeTheka project represents a paradigm shift in college management systems in India. By automating routine tasks, enhancing data security, and providing advanced analytics, it addresses the challenges faced by educational institutions in the modern era. The user-friendly interface and robust technological framework ensure that CollegeTheka is not just a system but a transformative tool for administrators, teachers, and students. As educational institutions in India continue to evolve, CollegeTheka stands as a beacon, guiding them towards a future where administrative processes are seamlessly integrated, allowing for a renewed focus on providing quality education and fostering a conducive learning environment.

Link of Github:

https://github.com/manangoel456/CollegeTheka_Sem3_cpp.git

Table of Contents

Chapter 1: **Introduction**

In the ever-evolving scenario of education, proficient management of administrative processes within a college environment is not just a requirement but a critical factor in determining the success of the institution. With a surge in student enrollment, the complexities of academic tracking, the nuances of examination management, and the imperative need for financial oversight, the demand for a comprehensive and integrated College Management System has never been more pronounced. This project, aptly titled "CollegeTheka," aims to address these challenges by offering a robust, user-friendly, and technologically advanced solution to redefine the way colleges manage their administrative tasks.

As higher education institutions grapple with an increasing influx of students, the manual administrative processes that were once sufficient are now proving to be inadequate. These processes are not only time-consuming but are also prone to errors, leading to inefficiencies that can have a cascading effect on the overall functioning of an institution. The CollegeTheka project emerges as a response to this paradigm shift, acknowledging the evolving needs of educational institutions and striving to revolutionize the way they handle administrative workflows.

a. Significance of CollegeTheka

College Management Systems serve as crucial components in the efficient functioning of educational institutions. By automating various administrative processes, these systems alleviate the workload on staff, reduce errors, and enhance overall productivity. The importance of a robust College Management System is particularly highlighted considering the diverse tasks that colleges handle on a daily basis.

From the intricate process of student enrollment to the meticulous tracking of academic progress, effective course management, and the orchestration of examinations, colleges navigate a complex web of responsibilities. The implementation of a comprehensive College Management System, such as CollegeTheka, not only promises to simplify these tasks but also aims to elevate them to a level of efficiency and sophistication that aligns with the contemporary demands of the Indian educational landscape.

The rationale behind the CollegeTheka project is deeply rooted in the need for a dynamic, adaptable, and user-centric solution to address the challenges faced by colleges in managing their administrative workflows. Traditional methods of manual record-keeping and task execution have become outdated and are prone to errors, leading to inefficiencies that can impact the overall functioning of an institution.

Through the development of CollegeTheka, our goal is to create a system that not only automates routine administrative tasks but also introduces features that empower administrators, teachers, and students. The project aims to bring about a paradigm shift in the way Indian colleges approach management systems, fostering an environment where technology is harnessed to augment decision-making processes and enhance the overall educational experience.

b. Scope And Objectives

The scope of the CollegeTheka project is vast, covering a wide range of functionalities crucial for the efficient administration of a college. The primary objectives include:

1. Efficient Student Enrolment:

The project aims to simplify and expedite the student enrollment process by providing a centralized and user-friendly platform for both administrators and prospective students. This reduces paperwork and enhances the overall experience of students during the admission process.

2. Comprehensive Academics Tracking:

Introducing a real-time tracking system to monitor and analyze the academic progress of students is a core objective. This feature enables timely interventions and personalized support, allowing teachers and administrators to identify areas of improvement and provide necessary guidance.

3. Streamlined Grade Management:

Automating the process of grade management is pivotal for transparency, accuracy, and timely dissemination of academic results. CollegeTheka seeks to minimize manual effort in this area, ensuring that grading processes are efficient and error-free.

4. Effective Examination Management:

The creation of a module that handles all aspects of examination management, from scheduling to result processing, is a key component. This automation not only reduces the burden on administrative staff but also minimizes the scope for errors in the examination process.

5. Automated Financial Oversight:

Managing academic dues and financial transactions seamlessly is critical for the financial health of an institution. CollegeTheka aims to provide administrators with a robust tool to monitor and control the financial aspects of the institution, ensuring accountability and transparency.

6. User-Friendly Interface:

Developing an intuitive and user-friendly interface is fundamental to the success of CollegeTheka. The project recognizes the importance of clear communication and collaboration among stakeholders and endeavors to create an interface that caters to the diverse needs of administrators, teachers, and students.

7. Data Security and Integrity:

Implementing robust security measures, including secure authentication systems, encryption, and role-based access control, is integral to the project. Safeguarding sensitive information is a top priority, and CollegeTheka aims to set high standards in data security and integrity.

The technological foundation of the CollegeTheka project is as critical as its functional components. Developed in Virtual Studio Code, the project leverages the power of C++ and Data Structures and Algorithms (DSA) to ensure efficiency, performance, and scalability. The incorporation of these technologies reflects a commitment to creating a system capable of handling the complexities inherent in college management.

Virtual Studio Code, a lightweight yet powerful source code editor, serves as the development environment for CollegeTheka. This choice is informed by the versatility and extensibility of the platform, offering an environment conducive to collaborative and efficient coding practices.

C++, a high-performance programming language, is selected for its ability to handle complex algorithms and large datasets efficiently. This choice is particularly relevant in the context of a college management system, where the processing of substantial amounts of data is inherent.

Data Structures and Algorithms (DSA) form the backbone of the project's functionality. The selection of DSA reflects a commitment to optimizing the efficiency and performance of the system, especially in scenarios where rapid data retrieval and manipulation are paramount.

The integration of these technologies is not arbitrary; rather, it represents a strategic approach to building a College Management System that is not only functional but also adaptable to the dynamic needs of educational institutions.

This comprehensive project report is structured to provide a detailed understanding of the CollegeTheka project, covering various aspects from the literature review to the methodology, functional features, technological framework, challenges faced, and potential future enhancements.

Following this introduction, the literature review section will delve into existing college management systems, analyzing their features, successes, and shortcomings. This critical examination sets the stage for understanding the unique contributions and innovations brought about by CollegeTheka.

The methodology section will outline the approach taken in the development of CollegeTheka. This will include insights into the software development life cycle, the rationale behind the selection of specific technologies, and the design choices made to ensure the project's success. The methodology provides a roadmap for understanding the intricacies of transforming an idea into a functional and efficient College Management System.

Subsequent sections of the report will dive into the functional aspects of CollegeTheka. Each section will focus on a specific module of the system, including student enrollment, academics tracking, grade management, examination management, financial oversight, and the user interface. These sections will not only describe the features but will also provide insights into the implementation details, potential challenges, and the anticipated impact on administrative processes within a college setting.

The technological framework, highlighted in the introduction, will be elucidated further in a dedicated section. This will provide a deeper understanding of the programming languages, tools, and security measures incorporated in the project. The technological framework section serves as a bridge between the functional aspects of CollegeTheka and the underlying architecture that powers its capabilities.

Chapter 2: **Problem Definition & Objectives**

A. Problem Statement

The "CollegeTheka" project addresses the intricate challenges inherent in college administration by proposing the development of a sophisticated College Management System. This system is designed to streamline and enhance administrative, student, and faculty operations within the college environment. The fundamental goal is to establish a secure and role-specific login mechanism for administrators, students, and faculty members. Administrators will have the capability to securely authenticate and perform essential administrative tasks, such as managing student and faculty records, overseeing course information, generating reports, and handling fee-related activities. Students, upon logging in, will gain access to a range of personal and academic information, including grades, attendance records, course registration features, and communication channels with both faculty and administrators. Faculty members, too, will benefit from secure logins, enabling them to manage student records, upload course materials, record grades, and communicate effectively with their students.

From a technical standpoint, the project necessitates the implementation of a robust relational database management system (RDBMS). This database will be meticulously designed to represent crucial entities like students, faculty, courses, and administrative details. The integration of Create, Read, Update, and Delete (CRUD) operations will facilitate seamless interactions with the database. The project will be developed using the C++ programming language, leveraging appropriate data structures and algorithms to optimize information processing efficiency. The user interface, whether command-line or graphical, will prioritize user-friendliness and intuitive design. Security considerations include the incorporation of strong authentication mechanisms, data encryption for sensitive information, and access controls to safeguard against unauthorized access.

Testing and quality assurance protocols will be implemented rigorously, encompassing unit and integration testing phases to ensure the accuracy of data, robust security measures, and system reliability. To provide comprehensive guidance and reference, the project documentation will include detailed information about the system architecture, database schema, code documentation, and user manuals. In summary, the "CollegeTheka" project aims to revolutionize college administration by delivering a centralized system that not only enhances operational efficiency and data accuracy but also fosters improved communication within the college community. The meticulous detailing of both functional and technical requirements will serve as a blueprint for the development process, ensuring the successful realization of the project's objectives.

B. Objectives

The primary objectives of the "CollegeTheka" project are to improve the efficiency and effectiveness of college administration through the development of an advanced College Management System. The main focus is on establishing secure authentication mechanisms for administrators, students, and faculty members, each with specific roles and permissions. For administrators, the goal is to streamline record management tasks, including the addition, updating, and deletion of student and faculty records. Additionally, the system will facilitate efficient course management, allowing administrators to add, modify, and delete course information and generate comprehensive reports. The project places a strong emphasis on fee tracking and management, ensuring accurate and effective handling of fee-related transactions.

For students, the objective is to create a secure login system that enables access to personal and academic information, including grades, attendance records, and exam results. The system will also facilitate course registration and provide students with access to course schedules and examination details. Communication channels between students, faculty, and administrators will be enhanced within the system, incorporating features such as messaging, announcements, and notifications. Faculty members will benefit from secure authentication mechanisms and tools to manage student grades, attendance records, and course materials efficiently. The project aims to empower faculty by providing a centralized platform for uploading, managing, and updating course materials, assignments, and announcements.

To ensure the security and integrity of the system, a reliable Relational Database Management System (RDBMS) will be utilized, accompanied by a normalized database schema representing entities such as students, faculty, courses, and administrative details. Robust authentication mechanisms, data encryption protocols, and access controls will be implemented to safeguard sensitive information and prevent unauthorized access. The development will prioritize a user-friendly interface, whether through the command line or a graphical interface, fostering consistency and ease of use. Comprehensive testing, including unit and integration testing, will be conducted to guarantee data accuracy, security measures, and system reliability. The documentation will be thorough, covering the system architecture, database schema, code documentation, and user manuals, providing a valuable reference for future maintenance and improvements. In conclusion, the "CollegeTheka" project aims to achieve a holistic transformation of college administration by fulfilling specific objectives related to efficiency, information access, communication, security, and user experience.

Chapter 3: **Proposed Work/Methodology**

The planned work and methodology for the "CollegeTheka" project involve a systematic and phased approach to ensure the successful development of a robust College Management System. The project will start with a thorough analysis of the specific requirements and objectives identified for administrators, students, and faculty members. This initial phase will include detailed discussions with stakeholders to gather insights into their needs and expectations. Following this, the system architecture will be meticulously designed, taking into account the utilization of a reliable Relational Database Management System (RDBMS) and a normalized database schema to efficiently store and manage data.

The development phase will use the C++ programming language, leveraging appropriate data structures and algorithms for efficient information processing. A modular design approach will be adopted to break down the system into manageable components, facilitating easier understanding, maintenance, and future scalability. The user interface, whether command-line or graphical, will be crafted with a focus on user-friendliness and intuitive design, promoting a positive user experience.

Security measures will be a paramount consideration throughout the development process. Strong authentication mechanisms, data encryption protocols, and access controls will be implemented to safeguard sensitive information and prevent unauthorized access. Thorough testing will be conducted at both unit and integration levels to ensure the accuracy of data, robustness of security measures, and overall system reliability. Quality assurance processes will be applied to address any identified issues and refine the system for optimal performance.

Comprehensive documentation will accompany each phase of the development, covering the system architecture, database schema, code documentation, and user manuals. This documentation will serve as a valuable resource for future maintenance, updates, and improvements. Continuous communication and feedback loops with stakeholders will be maintained throughout the development process to ensure alignment with expectations and requirements. The proposed methodology aims to deliver a seamlessly functioning College Management System that not only meets the specified objectives but also exceeds the expectations of administrators, students, and faculty members.

Chapter 4: **Data Structure Used**

In the development of the "CollegeTheka" project, we will use various data structures to efficiently organize and manage different types of information. Selecting appropriate data structures is crucial for optimizing the performance of the system. Here are some key data structures that may be used in different aspects of the project:

Arrays and Lists:

Purpose: Arrays and linked lists may be used to store and manage student and faculty records.
Usage: Arrays can be employed for fixed-size collections, while linked lists can accommodate dynamic growth and removal of records.

Trees (Binary Search Trees, AVL Trees):

Purpose: Binary Search Trees (BST) or AVL Trees can be used for efficient storage and retrieval of sorted data, such as student or faculty information.
Usage: These data structures facilitate quick search, insertion, and deletion operations, which are essential for managing records.

Hash Tables:

Purpose: Hash tables can be used for quick access to specific records based on unique identifiers, like student IDs or faculty IDs.
Usage: Hashing can provide constant time complexity for certain operations, enhancing the efficiency of data retrieval.

Queues and Stacks:

Purpose: Queues and stacks can be employed for managing tasks related to system processes, notifications, or communication channels.
Usage: Queues can handle tasks in a first-come-first-serve manner, while stacks can be useful for managing communication history.

Strings:

Purpose: Strings will be used for storing and processing textual information, such as names, course titles, or any other text-based data.
Usage: String manipulation operations will be applied for tasks like search, comparison, and display of information.

The specific choice of data structures will depend on the requirements of each module within the "CollegeTheka" project. Efficient data structures are crucial for optimizing time and space complexity, ensuring the smooth functioning of the system. The goal is to strike a balance between the nature of the data and the operations performed on that data, choosing structures that best suit the system's needs..

Chapter 5: **Language and Tools**

Programming Language:

The "CollegeTheka" project will be developed using the C++ programming language. C++ was chosen for its versatility, efficiency, and powerful features that make it well-suited for system-level programming. Its ability to handle low-level operations and manage memory directly provides a significant advantage, particularly when dealing with data structures and algorithms, essential components of the College Management System. C++ also supports object-oriented programming, allowing for the creation of modular and reusable code, which is crucial for the development of a complex and multifaceted system like CollegeTheka. Additionally, the language's widespread use and community support ensure a wealth of resources and libraries that can be leveraged during the development process.

Database Management System (DBMS):

For efficient data storage and retrieval, the project will use a relational database management system (RDBMS). Among the potential choices, databases like MySQL, PostgreSQL, or SQLite could be considered based on specific project requirements. The relational model of these databases aligns well with the structured nature of the data in a college management system, where relationships between entities such as students, faculty, and courses need to be maintained. A well-designed database will contribute to the overall performance, security, and scalability of the system.

Integrated Development Environment (IDE) and Version Control:

An integrated development environment like Visual Studio Code, Eclipse, or Code::Blocks will be used for coding, debugging, and project management. These environments provide a user-friendly interface, code highlighting, and debugging tools that enhance the development workflow. Version control, possibly using Git, will be implemented to track changes, manage collaboration among team members, and ensure a systematic approach to code management.

User Interface (UI) Design:

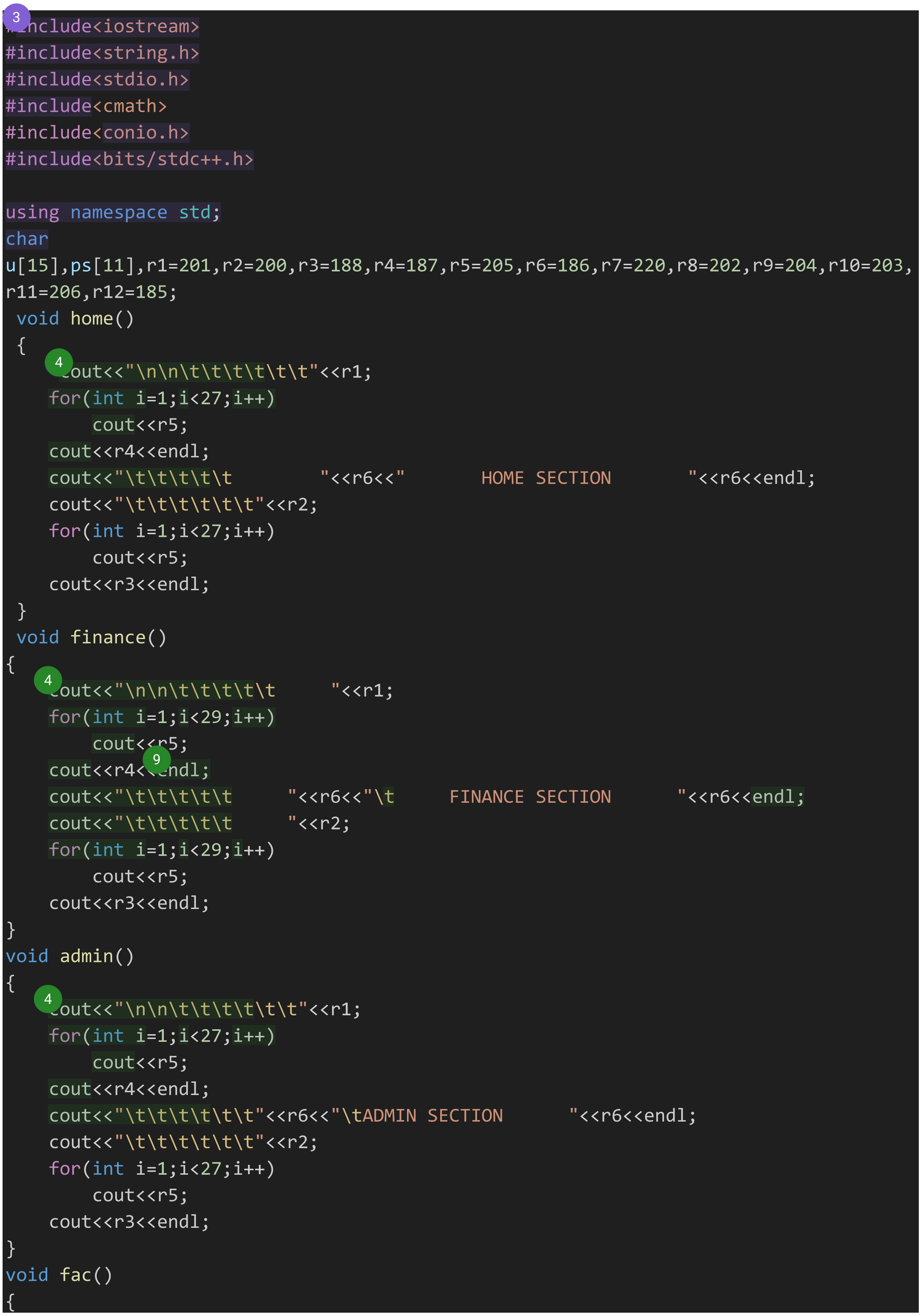
The user interface will be designed with a focus on simplicity, intuitiveness, and functionality. Depending on the project requirements and team preferences, a combination of command-line interfaces (CLI) and graphical user interfaces (GUI) may be implemented. Tools such as Qt for C++ GUI development or libraries like ncurses for command-line interfaces could be considered. The chosen tools will enable the creation of an interface that meets the diverse needs of administrators, students, and faculty members while providing a seamless and user-friendly experience.

Security Measures:

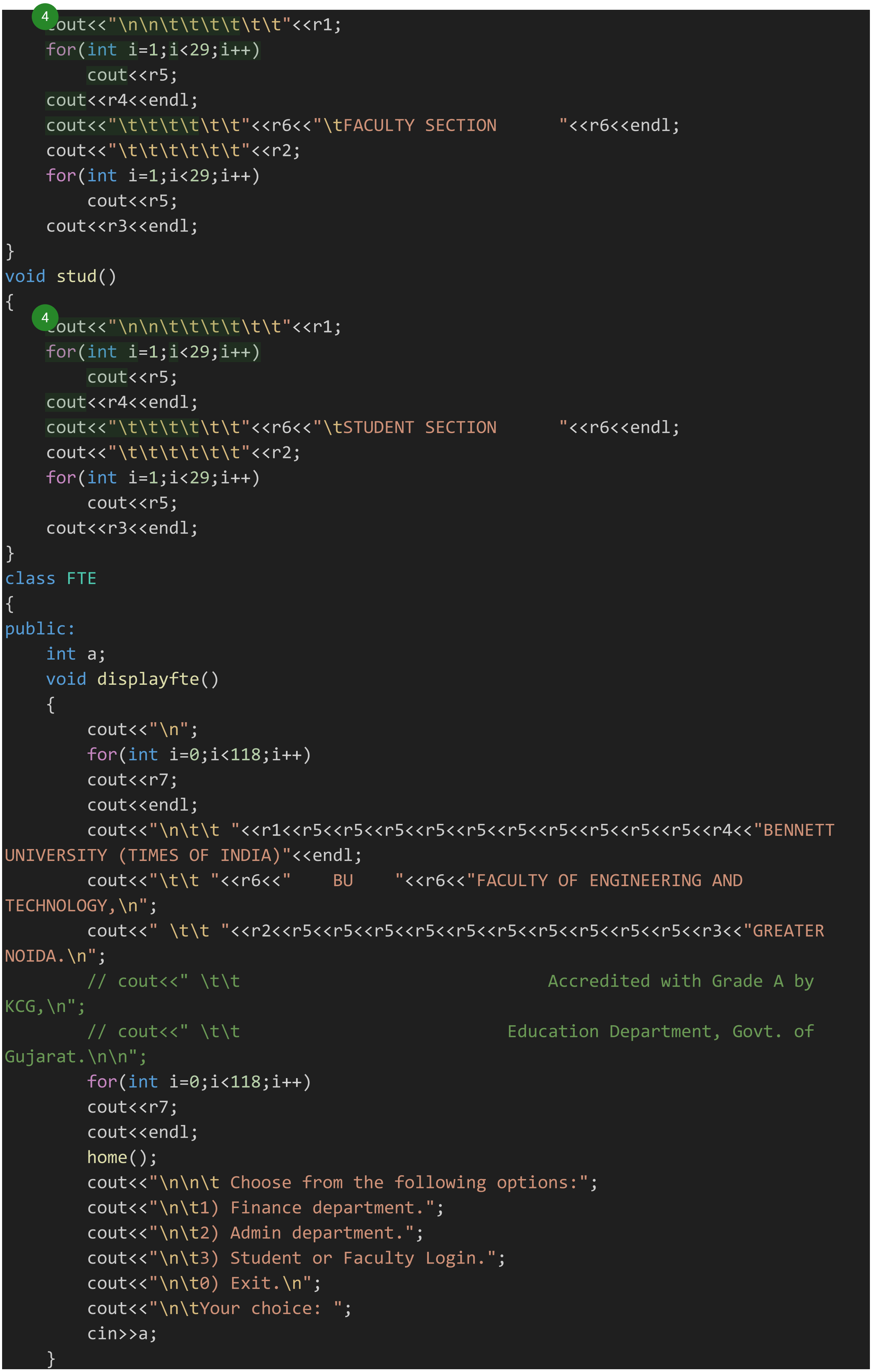
To ensure the security of user data and system integrity, the project will implement encryption protocols and access controls. Libraries such as OpenSSL for cryptographic functions and mechanisms for secure socket layer (SSL) implementation may be considered. Additionally, standard security practices and protocols will be followed during the development process to mitigate potential vulnerabilities and ensure a secure College Management System.

In summary, the "CollegeTheka" project will leverage the power of the C++ programming language, a relational database management system, and appropriate development tools to create a robust, efficient, and user-friendly system for college administration. The selected tools align with the project's goals of ensuring scalability, security, and maintainability throughout the development lifecycle.

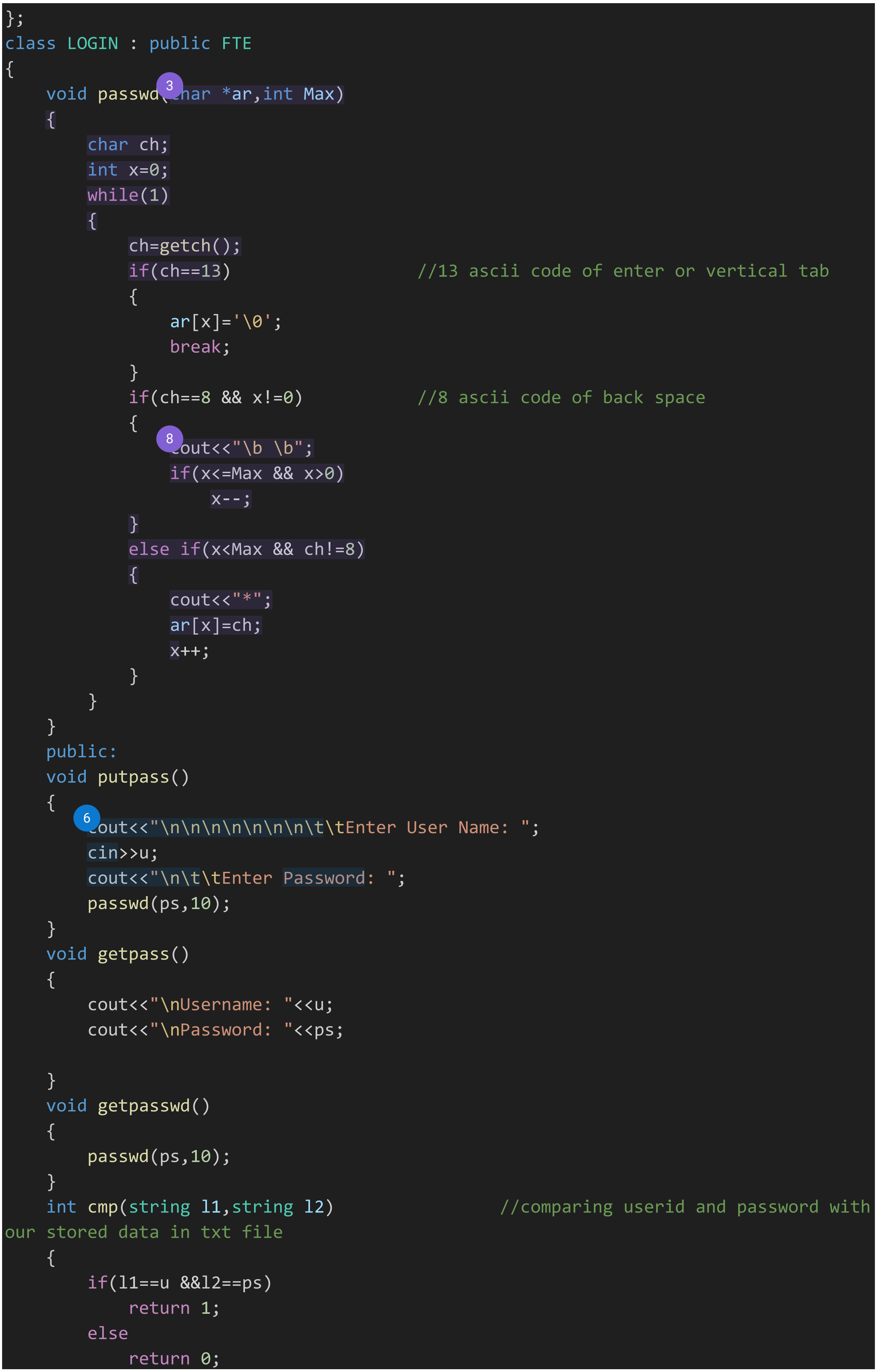
Chapter 6: **Source Code**



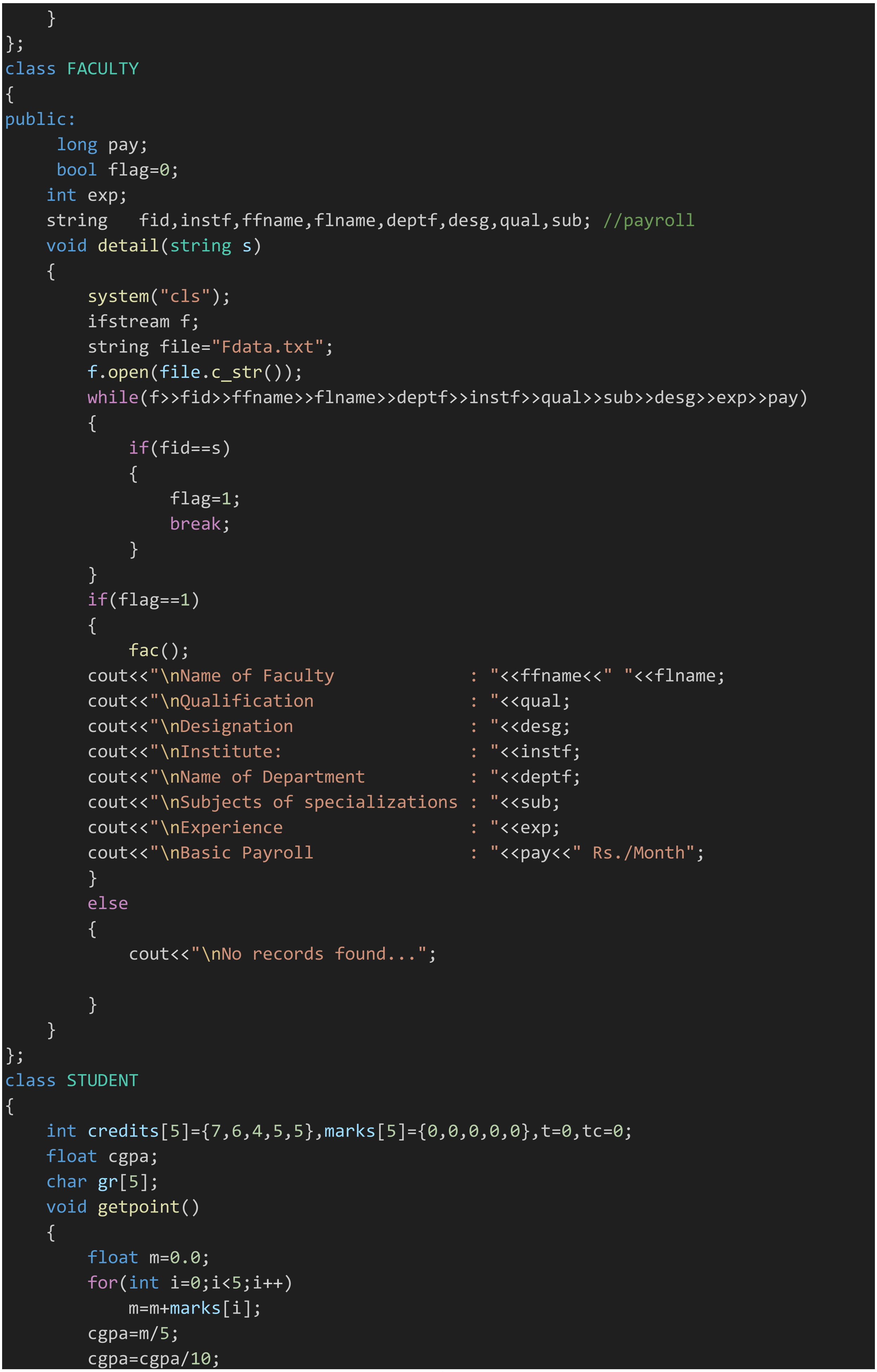
```
#include<iostream>
#include<string.h>
#include<stdio.h>
#include<cmath>
#include<conio.h>
#include<bits/stdc++.h>

using namespace std;
char
u[15],ps[11],r1=201,r2=200,r3=188,r4=187,r5=205,r6=186,r7=220,r8=202,r9=204,r10=203,
r11=206,r12=185;
 void home()
 {
     cout<<"\n\n\t\t\t\t\t\t\t"<<r1;
    for(int i=1;i<27;i++)
        cout<<r5;
    cout<<r4<<endl;
    cout<<"\t\t\t\t\t        "<<r6<<"       HOME SECTION       "<<r6<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r2;
    for(int i=1;i<27;i++)
        cout<<r5;
    cout<<r3<<endl;
 }
 void finance()
{
    cout<<"\n\n\t\t\t\t\t     "<<r1;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r4<<endl;
    cout<<"\t\t\t\t\t     "<<r6<<"\t     FINANCE SECTION      "<<r6<<endl;
    cout<<"\t\t\t\t\t     "<<r2;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r3<<endl;
}
void admin()
{
    cout<<"\n\n\t\t\t\t\t\t\t"<<r1;
    for(int i=1;i<27;i++)
        cout<<r5;
    cout<<r4<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r6<<"\tADMIN SECTION      "<<r6<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r2;
    for(int i=1;i<27;i++)
        cout<<r5;
    cout<<r3<<endl;
}
void fac()
{
```



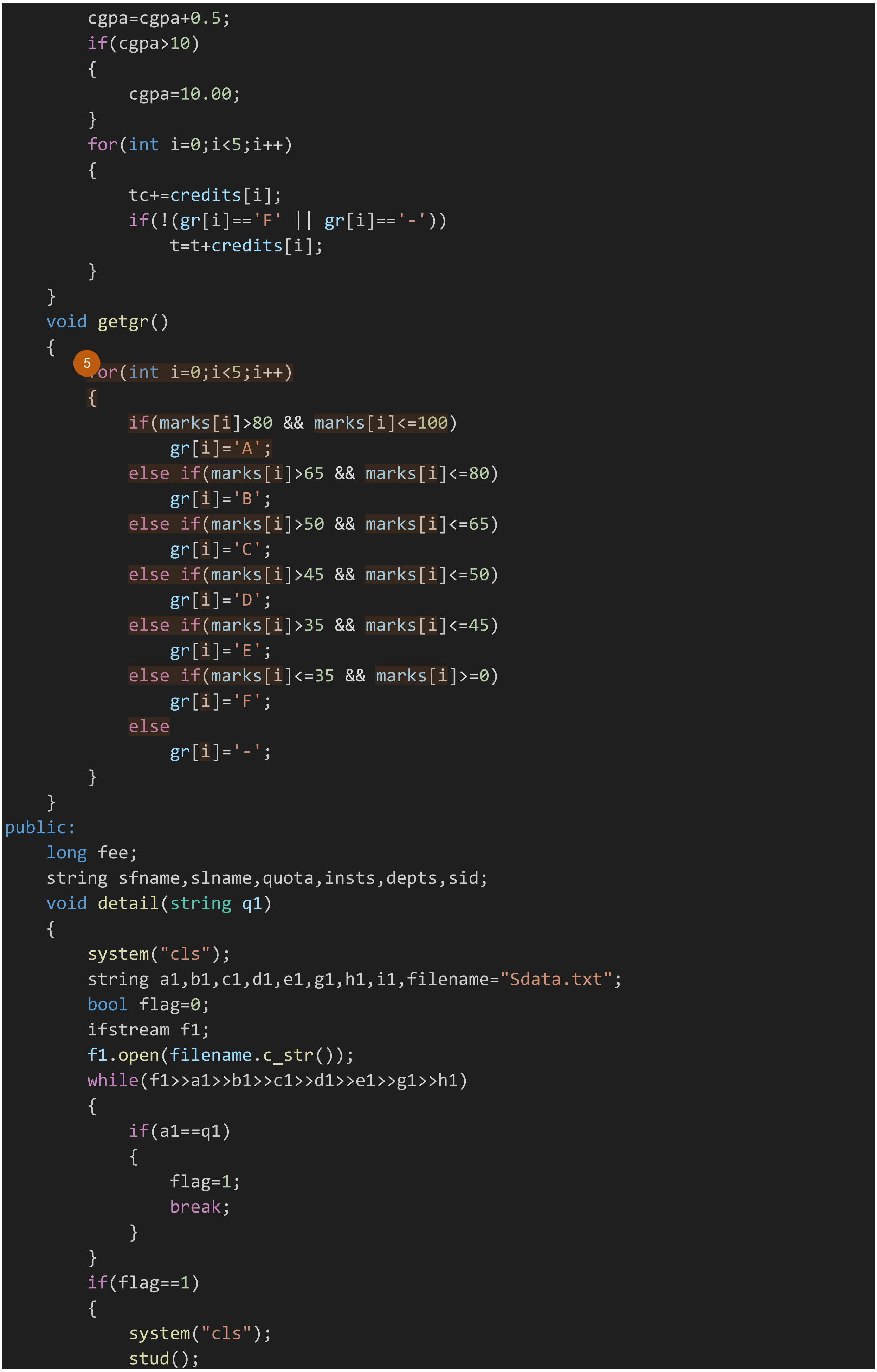
```
    cout<<"\n\n\t\t\t\t\t\t"<<r1;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r4<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r6<<"\tFACULTY SECTION      "<<r6<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r2;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r3<<endl;
}
void stud()
{
    cout<<"\n\n\t\t\t\t\t\t"<<r1;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r4<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r6<<"\tSTUDENT SECTION      "<<r6<<endl;
    cout<<"\t\t\t\t\t\t\t"<<r2;
    for(int i=1;i<29;i++)
        cout<<r5;
    cout<<r3<<endl;
}
class FTE
{
public:
    int a;
    void displayfte()
    {
        cout<<"\n";
        for(int i=0;i<118;i++)
        cout<<r7;
        cout<<endl;
        cout<<"\n\t\t "<<r1<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r4<<"BENNETT
UNIVERSITY (TIMES OF INDIA)"<<endl;
        cout<<"\t\t "<<r6<<"    BU    "<<r6<<"FACULTY OF ENGINEERING AND
TECHNOLOGY,\n";
        cout<<" \t\t "<<r2<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r3<<"GREATER
NOIDA.\n";
        // cout<<" \t\t                              Accredited with Grade A by
KCG,\n";
        // cout<<" \t\t                          Education Department, Govt. of
Gujarat.\n\n";
        for(int i=0;i<118;i++)
        cout<<r7;
        cout<<endl;
        home();
        cout<<"\n\n\t Choose from the following options:";
        cout<<"\n\t1) Finance department.";
        cout<<"\n\t2) Admin department.";
        cout<<"\n\t3) Student or Faculty Login.";
        cout<<"\n\t0) Exit.\n";
        cout<<"\n\tYour choice: ";
        cin>>a;
    }
```



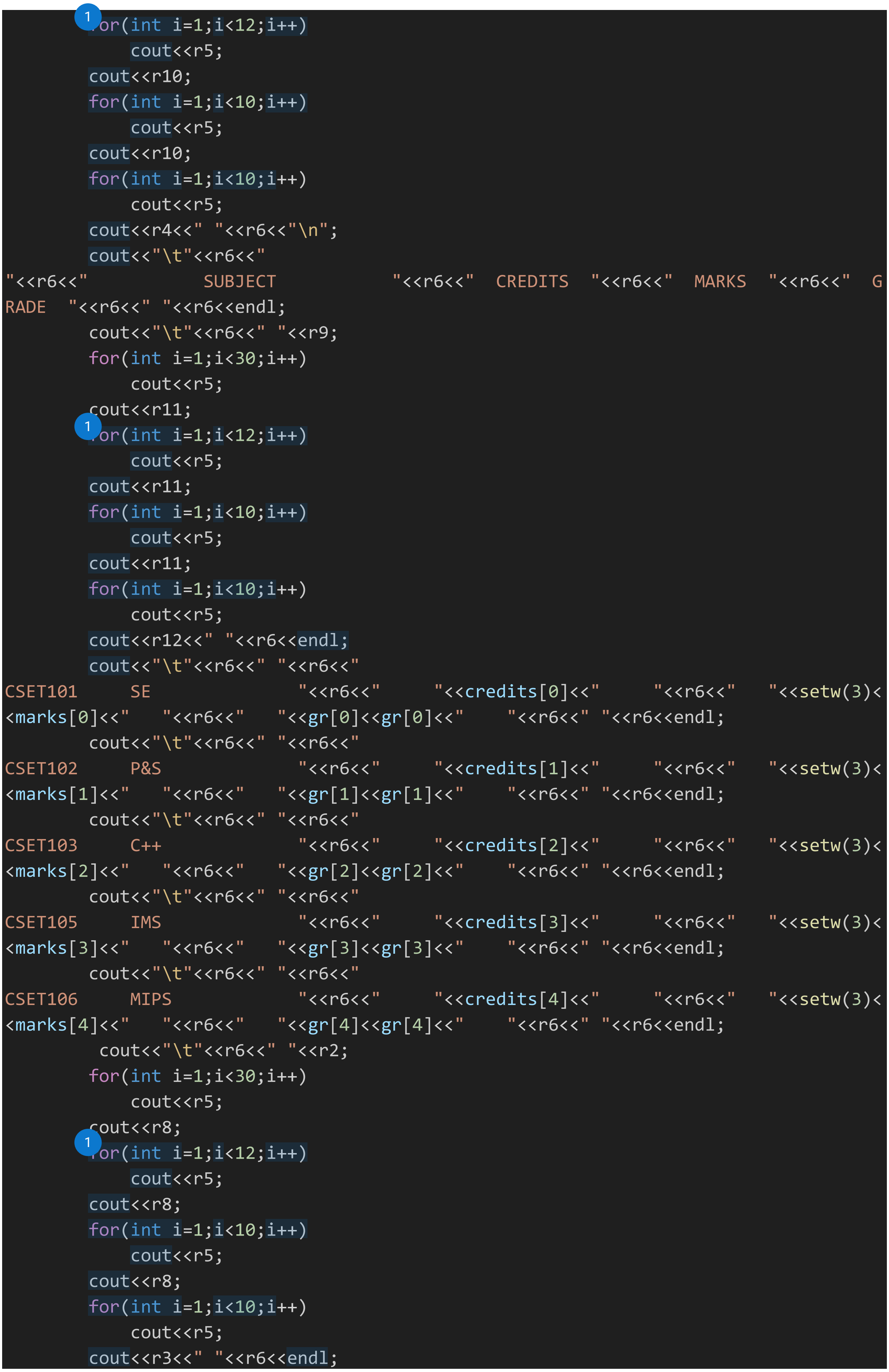
```
};
class LOGIN : public FTE
{
    void passwd(char *ar,int Max)
    {
        char ch;
        int x=0;
        while(1)
        {
            ch=getch();
            if(ch==13)                  //13 ascii code of enter or vertical tab
            {
                ar[x]='\0';
                break;
            }
            if(ch==8 && x!=0)           //8 ascii code of back space
            {
                cout<<"\b \b";
                if(x<=Max && x>0)
                    x--;
            }
            else if(x<Max && ch!=8)
            {
                cout<<"*";
                ar[x]=ch;
                x++;
            }
        }
    }
    public:
    void putpass()
    {
        cout<<"\n\n\n\n\n\n\n\t\tEnter User Name: ";
        cin>>u;
        cout<<"\n\t\tEnter Password: ";
        passwd(ps,10);
    }
    void getpass()
    {
        cout<<"\nUsername: "<<u;
        cout<<"\nPassword: "<<ps;

    }
    void getpasswd()
    {
        passwd(ps,10);
    }
    int cmp(string l1,string l2)                //comparing userid and password with
our stored data in txt file
    {
        if(l1==u &&l2==ps)
            return 1;
        else
            return 0;
```



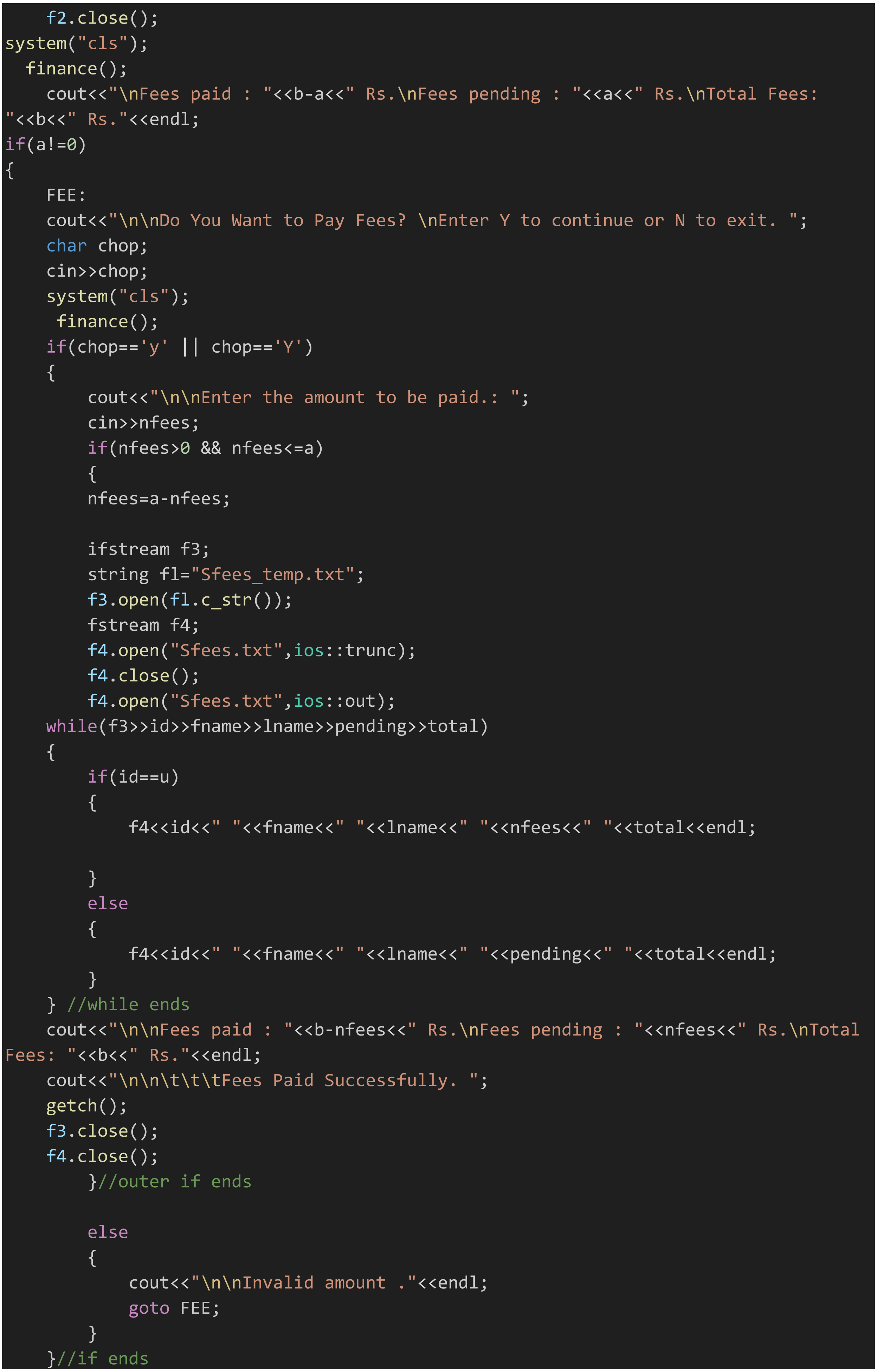
```
    }
};
class FACULTY
{
public:
     long pay;
     bool flag=0;
    int exp;
    string   fid,instf,ffname,flname,deptf,desg,qual,sub; //payroll
    void detail(string s)
    {
        system("cls");
        ifstream f;
        string file="Fdata.txt";
        f.open(file.c_str());
        while(f>>fid>>ffname>>flname>>deptf>>instf>>qual>>sub>>desg>>exp>>pay)
        {
            if(fid==s)
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        {
            fac();
        cout<<"\nName of Faculty            : "<<ffname<<" "<<flname;
        cout<<"\nQualification              : "<<qual;
        cout<<"\nDesignation                : "<<desg;
        cout<<"\nInstitute:                 : "<<instf;
        cout<<"\nName of Department         : "<<deptf;
        cout<<"\nSubjects of specializations : "<<sub;
        cout<<"\nExperience                 : "<<exp;
        cout<<"\nBasic Payroll              : "<<pay<<" Rs./Month";
        }
        else
        {
            cout<<"\nNo records found...";

        }
    }
};
class STUDENT
{
    int credits[5]={7,6,4,5,5},marks[5]={0,0,0,0,0},t=0,tc=0;
    float cgpa;
    char gr[5];
    void getpoint()
    {
        float m=0.0;
        for(int i=0;i<5;i++)
            m=m+marks[i];
        cgpa=m/5;
        cgpa=cgpa/10;
```



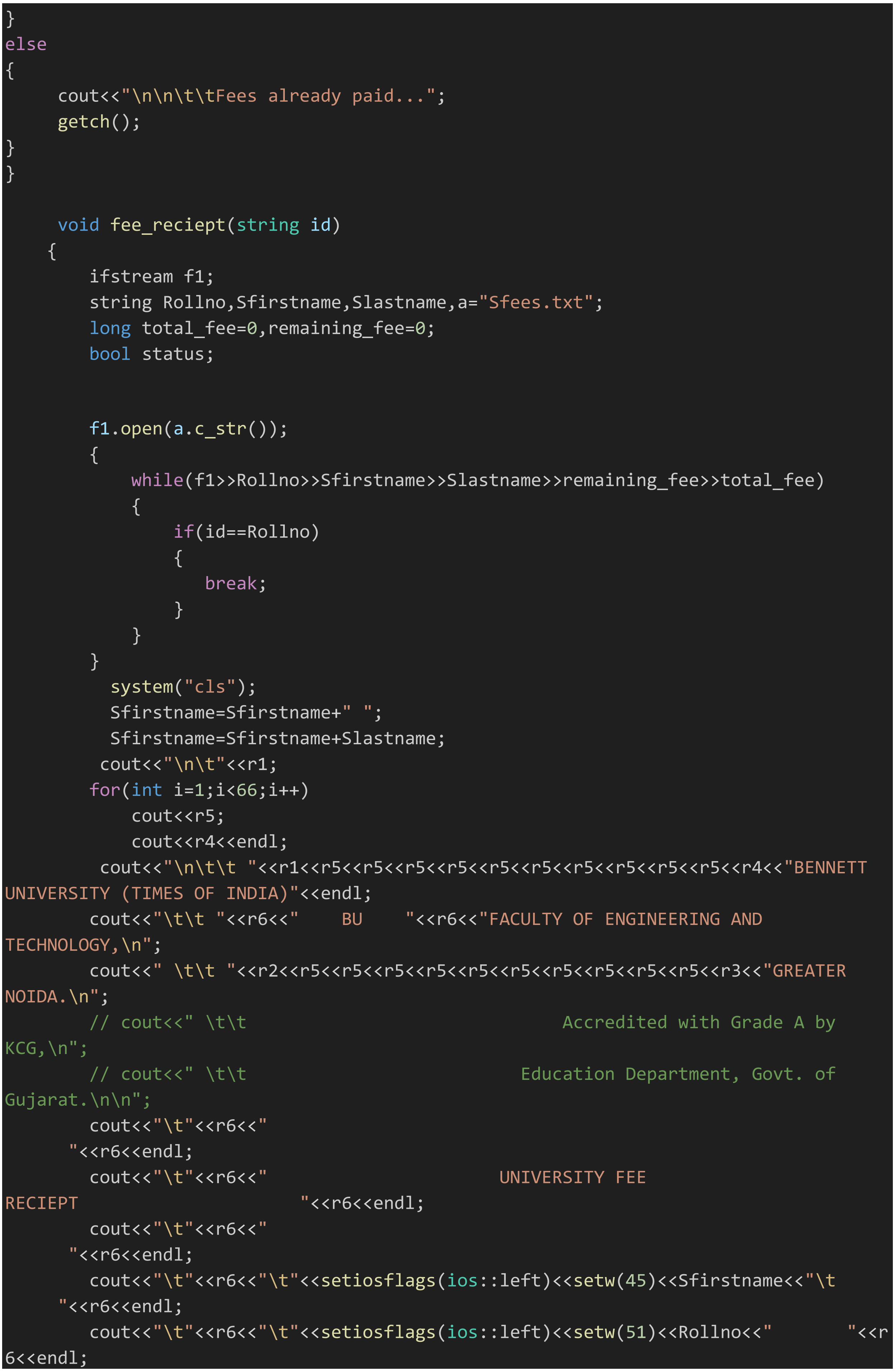
```
        cgpa=cgpa+0.5;
        if(cgpa>10)
        {
            cgpa=10.00;
        }
        for(int i=0;i<5;i++)
        {
            tc+=credits[i];
            if(!(gr[i]=='F' || gr[i]=='-'))
                t=t+credits[i];
        }
    }
    void getgr()
    {
        for(int i=0;i<5;i++)
        {
            if(marks[i]>80 && marks[i]<=100)
                gr[i]='A';
            else if(marks[i]>65 && marks[i]<=80)
                gr[i]='B';
            else if(marks[i]>50 && marks[i]<=65)
                gr[i]='C';
            else if(marks[i]>45 && marks[i]<=50)
                gr[i]='D';
            else if(marks[i]>35 && marks[i]<=45)
                gr[i]='E';
            else if(marks[i]<=35 && marks[i]>=0)
                gr[i]='F';
            else
                gr[i]='-';
        }
    }
public:
    long fee;
    string sfname,slname,quota,insts,depts,sid;
    void detail(string q1)
    {
        system("cls");
        string a1,b1,c1,d1,e1,g1,h1,i1,filename="Sdata.txt";
        bool flag=0;
        ifstream f1;
        f1.open(filename.c_str());
        while(f1>>a1>>b1>>c1>>d1>>e1>>g1>>h1)
        {
            if(a1==q1)
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        {
            system("cls");
            stud();
            cout<<"\n\n\n\tName of Student   : "<<b1<<" "<<c1;
            cout<<"\n\tStudent ID        : "<<a1;
            cout<<"\n\tDepartment        : "<<e1;
            cout<<"\n\tInstitute         : "<<d1;
            cout<<"\n\tFees per year     : "<<g1;
            cout<<"\n\tQuota of Admission: "<<h1;
            getch();
        }
        else
        {
            cout<<"\nNo records found...";
        }
    }
    void getmarks(string id)
    {

        ifstream f2,f1;
        string a,s2="Sresult.txt";
        int b,c,d,e,f;

        f1.open(s2.c_str());
        while(f1>>a>>b>>c>>d>>e>>f)
               {
                   if(a==id)
                   {
                       marks[0]=b;
                       marks[1]=c;
                       marks[2]=d;
                       marks[3]=e;
                       marks[4]=f;
                   }
               }

    }
    void result(string id)
    {
        fstream f1,f2;
        bool flag=0;
        string a1,a,c,d,e,k,b,l,m,b1;
        int f,g,h,i,j;
        a1="Sdata.txt";
        b1="Sresult.txt";
        f2.open(b1.c_str());
        while(f2>>k>>f>>g>>h>>i>>j)
        {
            if(k==id)
            {
                marks[0]=f;
                marks[1]=g;
                marks[2]=h;
                marks[3]=i;
                marks[4]=j;
                flag=0;
                break;
            }
            else
               flag=1;
        }
        if(flag==1)
        {
            cout<<"Result not declared..";
        }
        else
        {
        f1.open(a1.c_str());
        while(f1>>a>>b>>c>>d>>e>>l>>m)
        {
            if(a==id)
            {
                sfname=b;
                slname=c;
                break;
            }
        }
        system("cls");
        getgr();
        getpoint();
        string name=sfname+" ";
        name+=slname;
        getmarks(id);
        cout<<"\n\t"<<r1;
        for(int i=1;i<66;i++)
            cout<<r5;
            cout<<r4<<endl;
        cout<<"\n\t\t "<<r1<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r4<<"BENNETT
UNIVERSITY (TIMES OF INDIA)"<<endl;
        cout<<"\t\t "<<r6<<"    BU     "<<r6<<"FACULTY OF ENGINEERING AND
TECHNOLOGY,\n";
        cout<<" \t\t "<<r2<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r3<<" GREATER
NOIDA.\n";
        // cout<<" \t\t                              Accredited with Grade A by
KCG,\n";
        // cout<<" \t\t                          Education Department, Govt. of
Gujarat.\n\n";
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r6<<"\t\t\t UNIVERSITY EXAM RESULT \t\t  "<<r6<<endl;
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r6<<"\t"<<setiosflags(ios::left)<<setw(55)<<name<<"   "<<r6<<end
l;
        cout<<"\t"<<r6<<"\t"<<setiosflags(ios::left)<<setw(55)<<id<<"   "<<r6<<endl;
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r1;
        for(int i=1;i<30;i++)
            cout<<r5;
        cout<<r10;
```



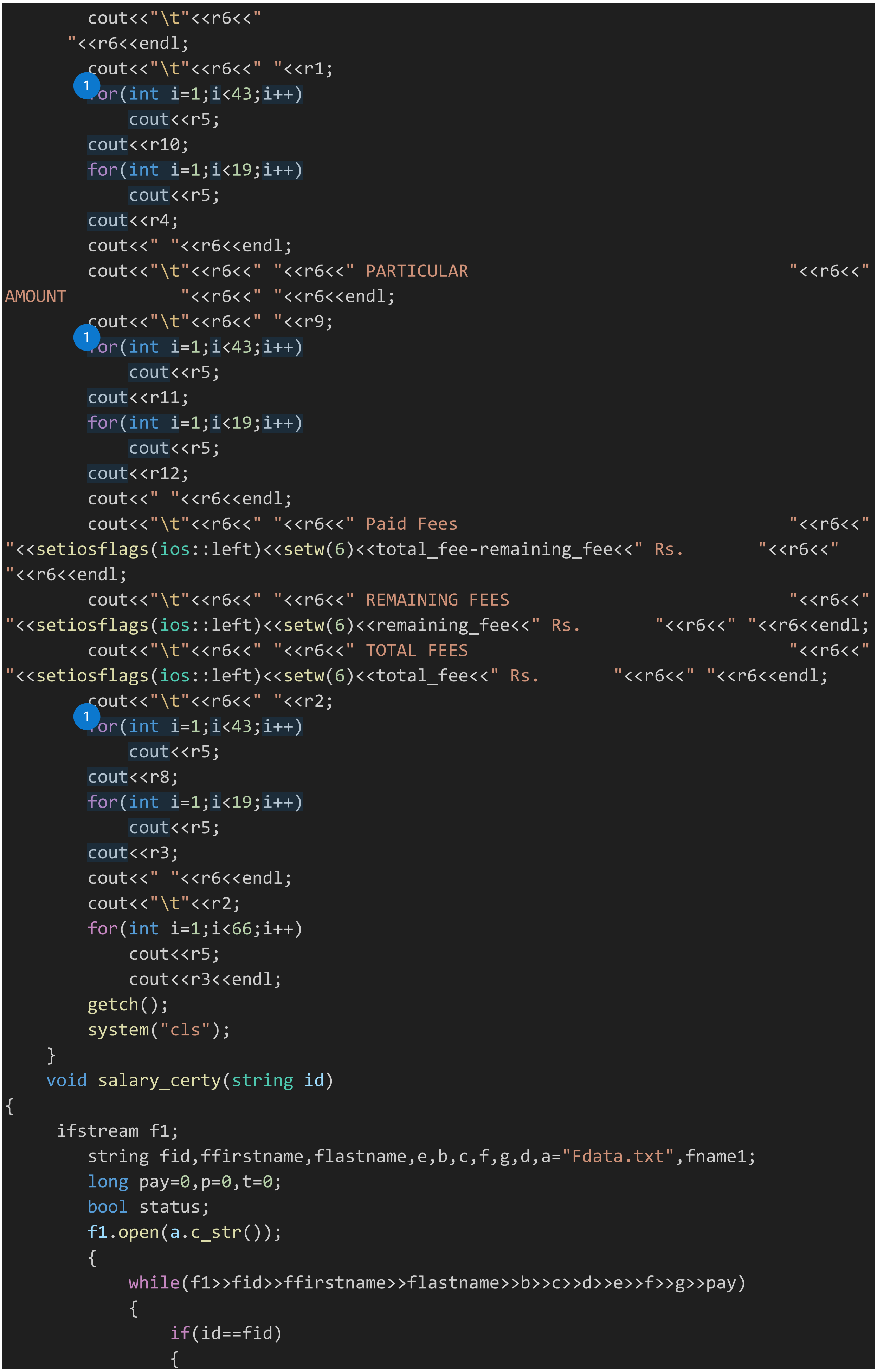
```
        for(int i=1;i<12;i++)
            cout<<r5;
        cout<<r10;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r10;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r4<<" "<<r6<<"\n";
        cout<<"\t"<<r6<<"
"<<r6<<"           SUBJECT           "<<r6<<"  CREDITS  "<<r6<<"  MARKS  "<<r6<<"  G
RADE  "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r9;
        for(int i=1;i<30;i++)
            cout<<r5;
        cout<<r11;
        for(int i=1;i<12;i++)
            cout<<r5;
        cout<<r11;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r11;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r12<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<"
CSET101     SE              "<<r6<<"     "<<credits[0]<<"     "<<r6<<"   "<<setw(3)<
<marks[0]<<"   "<<r6<<"   "<<gr[0]<<gr[0]<<"    "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<"
CSET102     P&S             "<<r6<<"     "<<credits[1]<<"     "<<r6<<"   "<<setw(3)<
<marks[1]<<"   "<<r6<<"   "<<gr[1]<<gr[1]<<"    "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<"
CSET103     C++             "<<r6<<"     "<<credits[2]<<"     "<<r6<<"   "<<setw(3)<
<marks[2]<<"   "<<r6<<"   "<<gr[2]<<gr[2]<<"    "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<"
CSET105     IMS             "<<r6<<"     "<<credits[3]<<"     "<<r6<<"   "<<setw(3)<
<marks[3]<<"   "<<r6<<"   "<<gr[3]<<gr[3]<<"    "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<"
CSET106     MIPS            "<<r6<<"     "<<credits[4]<<"     "<<r6<<"   "<<setw(3)<
<marks[4]<<"   "<<r6<<"   "<<gr[4]<<gr[4]<<"    "<<r6<<" "<<r6<<endl;
         cout<<"\t"<<r6<<" "<<r2;
        for(int i=1;i<30;i++)
            cout<<r5;
        cout<<r8;
        for(int i=1;i<12;i++)
            cout<<r5;
        cout<<r8;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r8;
        for(int i=1;i<10;i++)
            cout<<r5;
        cout<<r3<<" "<<r6<<endl;
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r6<<"\tCGPA :
"<<fixed<<setw(5)<<setprecision(2)<<cgpa<<"
     "<<r6<<endl;
        cout<<"\t"<<r6<<"\tTotal credits :
"<<setw(2)<<tc<<"                                        "<<r6<<endl;
        cout<<"\t"<<r6<<"\tTotal credits earned :
"<<setw(2)<<t<<"                                 "<<r6<<endl;
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r2;
        for(int i=1;i<66;i++)
            cout<<r5;

        cout<<r3<<endl;
            getch();
            system("cls");
        }
    }

};
class FINANCE : public FTE , public STUDENT , public FACULTY
{
public:
      void modify_fees()
{
    string c,d,e;
    long nfees;
    string file="Sfees.txt";
    string fname,lname,id;
    long total,a,b,pending;
    ifstream f1;
    f1.open(file.c_str());
    fstream f2;
    f2.open("Sfees_temp.txt",ios::trunc);
    f2.close();
    f2.open("Sfees_temp.txt",ios::out);
//     cout<<"\nEnter userID: ";
//    cin>>u;
    while(f1>>id>>fname>>lname>>pending>>total)
    {
        f2<<id<<" "<<fname<<" "<<lname<<" "<<pending<<" "<<total<<endl;
        if(id==u)
        {
            c=id;
            d=fname;
            e=lname;
            a=pending;
            b=total;
        }
    }

        f1.close();
```



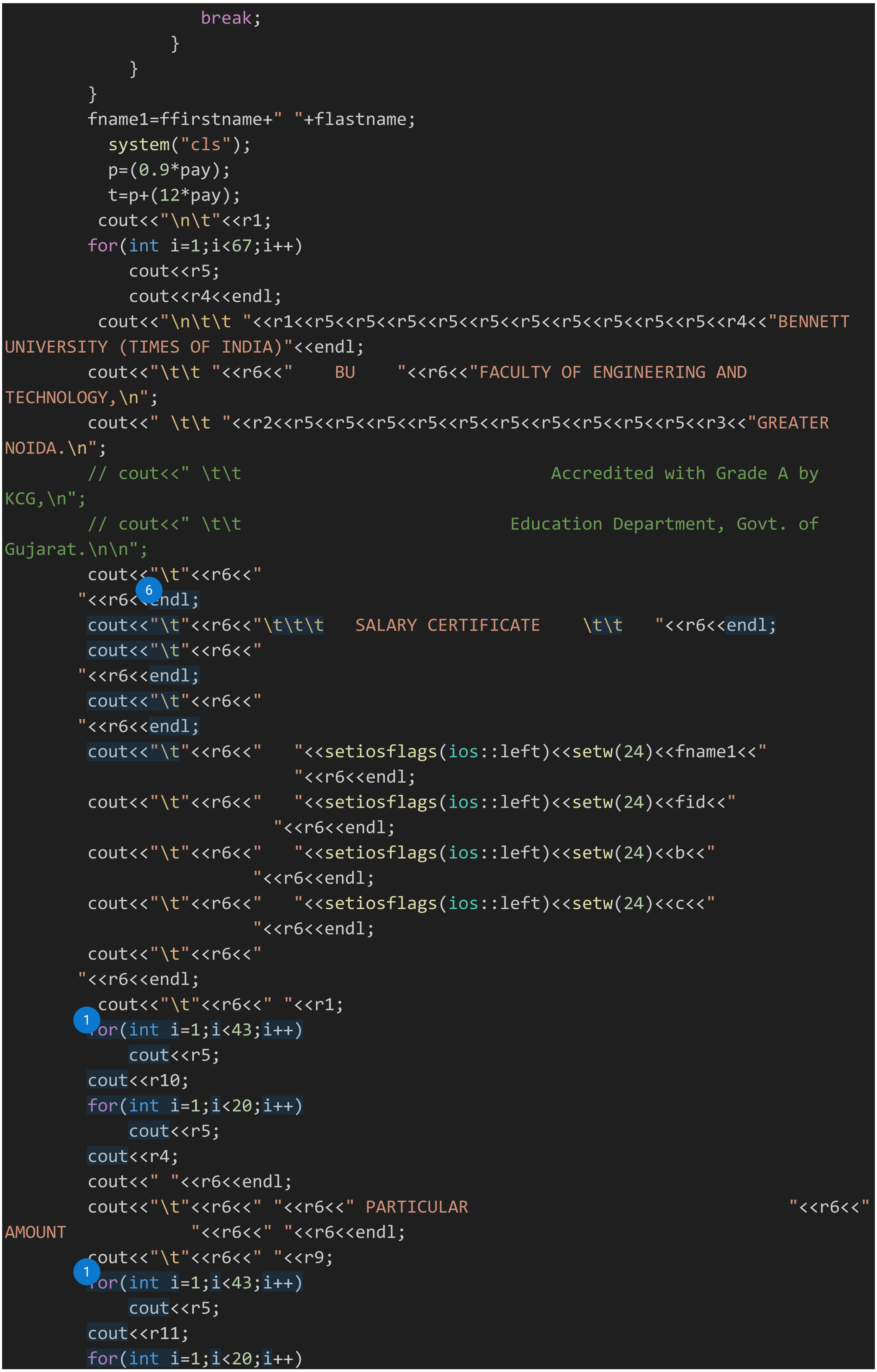
```
    f2.close();
system("cls");
  finance();
    cout<<"\nFees paid : "<<b-a<<" Rs.\nFees pending : "<<a<<" Rs.\nTotal Fees: 
"<<b<<" Rs."<<endl;
if(a!=0)
{
    FEE:
    cout<<"\n\nDo You Want to Pay Fees? \nEnter Y to continue or N to exit. ";
    char chop;
    cin>>chop;
    system("cls");
     finance();
    if(chop=='y' || chop=='Y')
    {
        cout<<"\n\nEnter the amount to be paid.: ";
        cin>>nfees;
        if(nfees>0 && nfees<=a)
        {
        nfees=a-nfees;

        ifstream f3;
        string fl="Sfees_temp.txt";
        f3.open(fl.c_str());
        fstream f4;
        f4.open("Sfees.txt",ios::trunc);
        f4.close();
        f4.open("Sfees.txt",ios::out);
    while(f3>>id>>fname>>lname>>pending>>total)
    {
        if(id==u)
        {
            f4<<id<<" "<<fname<<" "<<lname<<" "<<nfees<<" "<<total<<endl;

        }
        else
        {
            f4<<id<<" "<<fname<<" "<<lname<<" "<<pending<<" "<<total<<endl;
        }
    } //while ends
    cout<<"\n\nFees paid : "<<b-nfees<<" Rs.\nFees pending : "<<nfees<<" Rs.\nTotal 
Fees: "<<b<<" Rs."<<endl;
    cout<<"\n\n\t\t\tFees Paid Successfully. ";
    getch();
    f3.close();
    f4.close();
        }//outer if ends

        else
        {
            cout<<"\n\nInvalid amount ."<<endl;
            goto FEE;
        }
    }//if ends
```



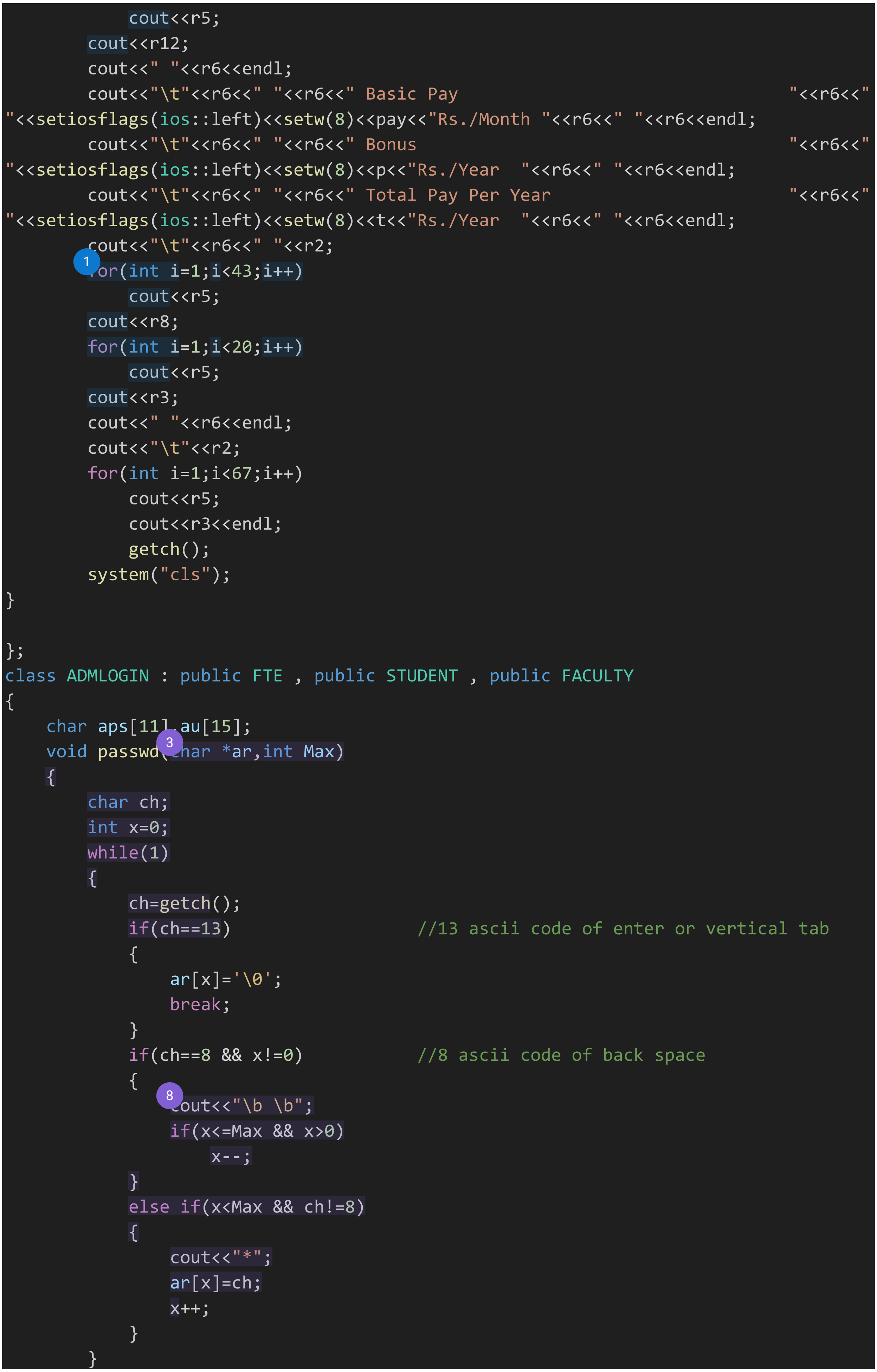
```
}
else
{
     cout<<"\n\n\t\tFees already paid...";
     getch();
}
}

     void fee_reciept(string id)
    {
        ifstream f1;
        string Rollno,Sfirstname,Slastname,a="Sfees.txt";
        long total_fee=0,remaining_fee=0;
        bool status;


        f1.open(a.c_str());
        {
            while(f1>>Rollno>>Sfirstname>>Slastname>>remaining_fee>>total_fee)
            {
                if(id==Rollno)
                {
                   break;
                }
            }
        }
          system("cls");
          Sfirstname=Sfirstname+" ";
          Sfirstname=Sfirstname+Slastname;
         cout<<"\n\t"<<r1;
        for(int i=1;i<66;i++)
            cout<<r5;
            cout<<r4<<endl;
         cout<<"\n\t\t "<<r1<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r4<<"BENNETT 
UNIVERSITY (TIMES OF INDIA)"<<endl;
        cout<<"\t\t "<<r6<<"    BU    "<<r6<<"FACULTY OF ENGINEERING AND 
TECHNOLOGY,\n";
        cout<<" \t\t "<<r2<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r3<<"GREATER 
NOIDA.\n";
        // cout<<" \t\t                             Accredited with Grade A by 
KCG,\n";
        // cout<<" \t\t                         Education Department, Govt. of 
Gujarat.\n\n";
        cout<<"\t"<<r6<<"                                                       
      "<<r6<<endl;
        cout<<"\t"<<r6<<"                      UNIVERSITY FEE 
RECIEPT                     "<<r6<<endl;
        cout<<"\t"<<r6<<"                                                       
      "<<r6<<endl;
        cout<<"\t"<<r6<<"\t"<<setiosflags(ios::left)<<setw(45)<<Sfirstname<<"\t 
     "<<r6<<endl;
        cout<<"\t"<<r6<<"\t"<<setiosflags(ios::left)<<setw(51)<<Rollno<<"       "<<r
6<<endl;
```



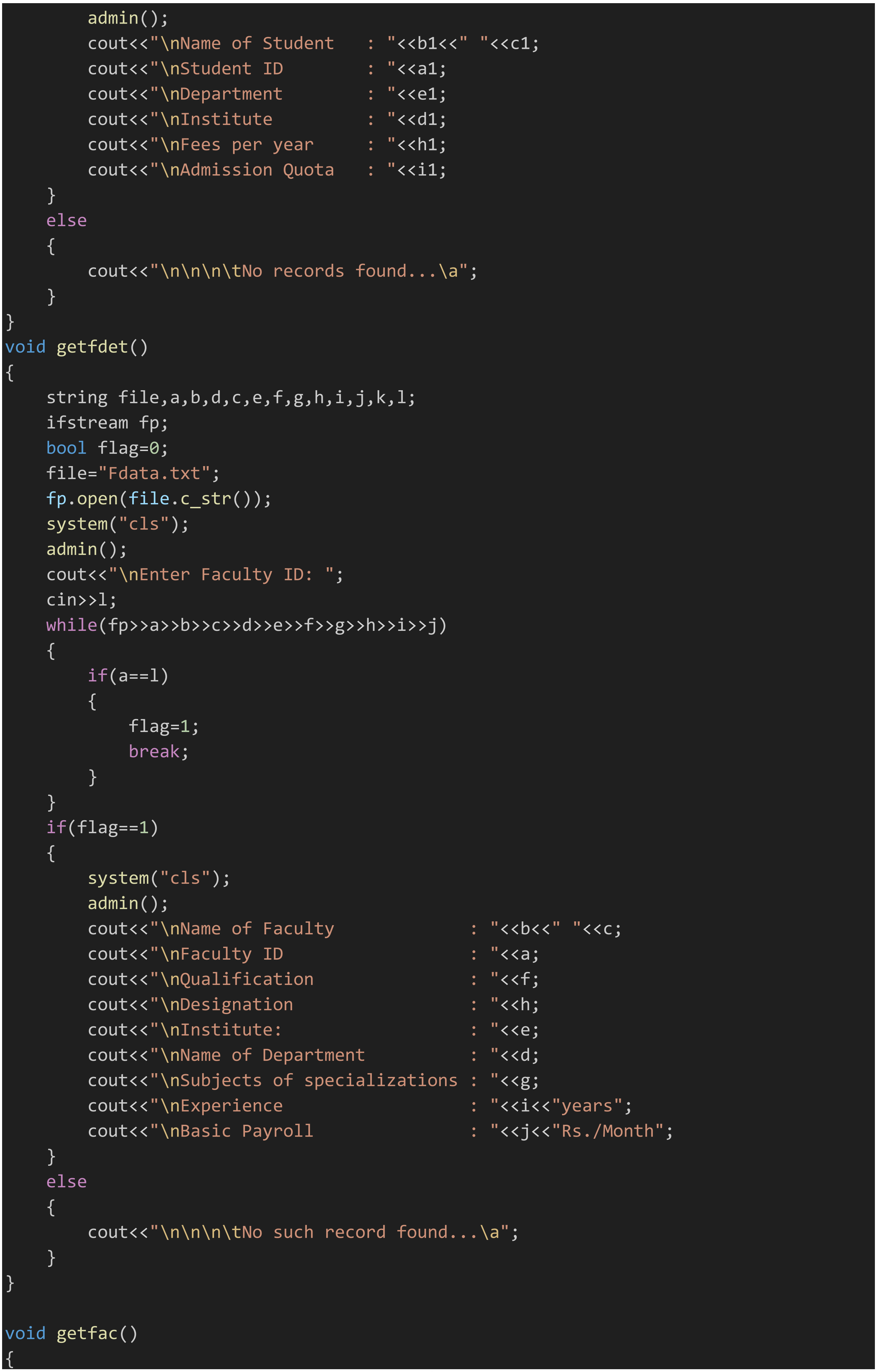
```
        cout<<"\t"<<r6<<"
      "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r1;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r10;
        for(int i=1;i<19;i++)
            cout<<r5;
        cout<<r4;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" PARTICULAR                               "<<r6<<"
AMOUNT          "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r9;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r11;
        for(int i=1;i<19;i++)
            cout<<r5;
        cout<<r12;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" Paid Fees                                "<<r6<<"
"<<setiosflags(ios::left)<<setw(6)<<total_fee-remaining_fee<<" Rs.       "<<r6<<"
"<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" REMAINING FEES                           "<<r6<<"
"<<setiosflags(ios::left)<<setw(6)<<remaining_fee<<" Rs.       "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" TOTAL FEES                               "<<r6<<"
"<<setiosflags(ios::left)<<setw(6)<<total_fee<<" Rs.       "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r2;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r8;
        for(int i=1;i<19;i++)
            cout<<r5;
        cout<<r3;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r2;
        for(int i=1;i<66;i++)
            cout<<r5;
            cout<<r3<<endl;
        getch();
        system("cls");
    }
    void salary_certy(string id)
{
     ifstream f1;
        string fid,ffirstname,flastname,e,b,c,f,g,d,a="Fdata.txt",fname1;
        long pay=0,p=0,t=0;
        bool status;
        f1.open(a.c_str());
        {
            while(f1>>fid>>ffirstname>>flastname>>b>>c>>d>>e>>f>>g>>pay)
            {
                if(id==fid)
                {
```



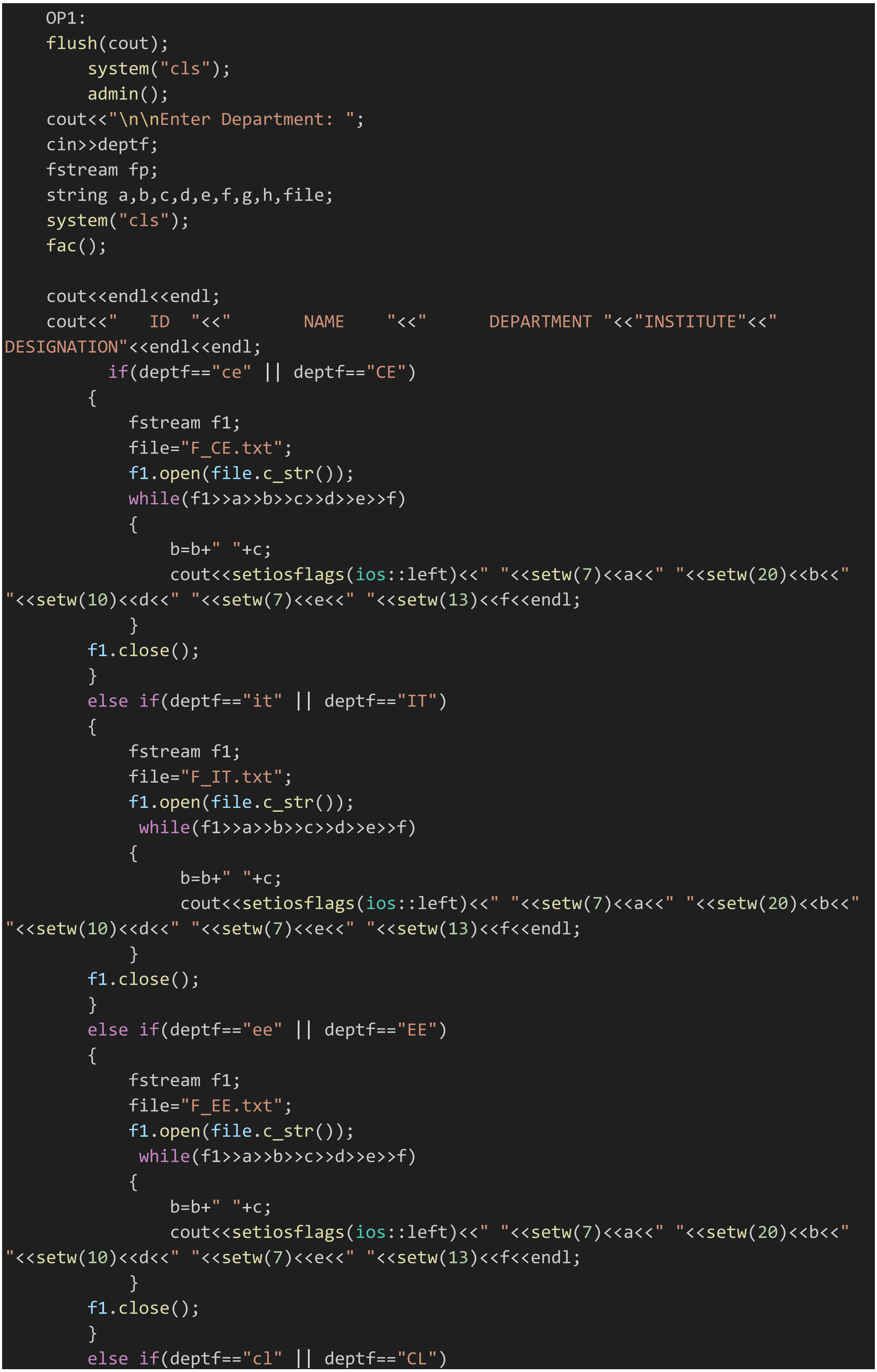
```
                   break;
                }
            }
        }
        fname1=ffirstname+" "+flastname;
          system("cls");
          p=(0.9*pay);
          t=p+(12*pay);
         cout<<"\n\t"<<r1;
        for(int i=1;i<67;i++)
            cout<<r5;
            cout<<r4<<endl;
         cout<<"\n\t\t "<<r1<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r4<<"BENNETT
UNIVERSITY (TIMES OF INDIA)"<<endl;
        cout<<"\t\t "<<r6<<"    BU    "<<r6<<"FACULTY OF ENGINEERING AND
TECHNOLOGY,\n";
        cout<<" \t\t "<<r2<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r5<<r3<<"GREATER
NOIDA.\n";
        // cout<<" \t\t                              Accredited with Grade A by
KCG,\n";
        // cout<<" \t\t                          Education Department, Govt. of
Gujarat.\n\n";
        cout<<"\t"<<r6<<"
       "<<r6<<endl;
        cout<<"\t"<<r6<<"\t\t\t   SALARY CERTIFICATE    \t\t   "<<r6<<endl;
        cout<<"\t"<<r6<<"
       "<<r6<<endl;
        cout<<"\t"<<r6<<"
       "<<r6<<endl;
        cout<<"\t"<<r6<<"   "<<setiosflags(ios::left)<<setw(24)<<fname1<<"
                            "<<r6<<endl;
        cout<<"\t"<<r6<<"   "<<setiosflags(ios::left)<<setw(24)<<fid<<"
                          "<<r6<<endl;
        cout<<"\t"<<r6<<"   "<<setiosflags(ios::left)<<setw(24)<<b<<"
                        "<<r6<<endl;
        cout<<"\t"<<r6<<"   "<<setiosflags(ios::left)<<setw(24)<<c<<"
                        "<<r6<<endl;
        cout<<"\t"<<r6<<"
       "<<r6<<endl;
         cout<<"\t"<<r6<<" "<<r1;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r10;
        for(int i=1;i<20;i++)
            cout<<r5;
        cout<<r4;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" PARTICULAR                               "<<r6<<"
AMOUNT            "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r9;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r11;
        for(int i=1;i<20;i++)
```



```
            cout<<r5;
        cout<<r12;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" Basic Pay                                "<<r6<<"
"<<setiosflags(ios::left)<<setw(8)<<pay<<"Rs./Month "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" Bonus                                    "<<r6<<"
"<<setiosflags(ios::left)<<setw(8)<<p<<"Rs./Year  "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r6<<" Total Pay Per Year                       "<<r6<<"
"<<setiosflags(ios::left)<<setw(8)<<t<<"Rs./Year  "<<r6<<" "<<r6<<endl;
        cout<<"\t"<<r6<<" "<<r2;
        for(int i=1;i<43;i++)
            cout<<r5;
        cout<<r8;
        for(int i=1;i<20;i++)
            cout<<r5;
        cout<<r3;
        cout<<" "<<r6<<endl;
        cout<<"\t"<<r2;
        for(int i=1;i<67;i++)
            cout<<r5;
            cout<<r3<<endl;
            getch();
        system("cls");
}

};
class ADMLOGIN : public FTE , public STUDENT , public FACULTY
{
    char aps[11],au[15];
    void passwd(char *ar,int Max)
    {
        char ch;
        int x=0;
        while(1)
        {
            ch=getch();
            if(ch==13)                  //13 ascii code of enter or vertical tab
            {
                ar[x]='\0';
                break;
            }
            if(ch==8 && x!=0)           //8 ascii code of back space
            {
                cout<<"\b \b";
                if(x<=Max && x>0)
                    x--;
            }
            else if(x<Max && ch!=8)
            {
                cout<<"*";
                ar[x]=ch;
                x++;
            }
        }
    }

    void gethod()
    {
        fstream fp;
        bool flag=1;
        string file="Fdata.txt",a,b,c,d,e,f,g,h,i,j;
        fp.open(file.c_str());
        system("cls");
        fac();
        cout<<"List of hod's :\n\n";
        resetiosflags(ios::right);
        cout<<"   ID  "<<"     NAME    "<<"     DEPARTMENT     "<<"
INSTITUTE"<<endl<<endl;
        while(fp>>a>>b>>c>>d>>e>>f>>g>>h>>i>>j)
        {
            if(h=="hod" || h=="HOD")
            {
                flag=0;
                b=b+" "+c;
                cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(12)<<d<<setw(11)<<e<<endl;
            }
        }
        if(flag==1)
        {
            cout<<"\n\n\n\tNo records found...\a";
        }
        getch();
        system("cls");
    }
    void getsdet()
    {
    string q1,a1,b1,c1,d1,e1,g1,h1,i1,filename="Sdata.txt";
    system("cls");
    admin();
    cout<<"\nEnter the Student ID: ";
    cin>>q1;


    bool flag=0;
    ifstream f1;
    f1.open(filename.c_str());
    while(f1>>a1>>b1>>c1>>d1>>e1>>h1>>i1)
    {
        if(a1==q1)
         {
             flag=1;
             break;
         }
    }
    if(flag==1)
    {
        system("cls");
```

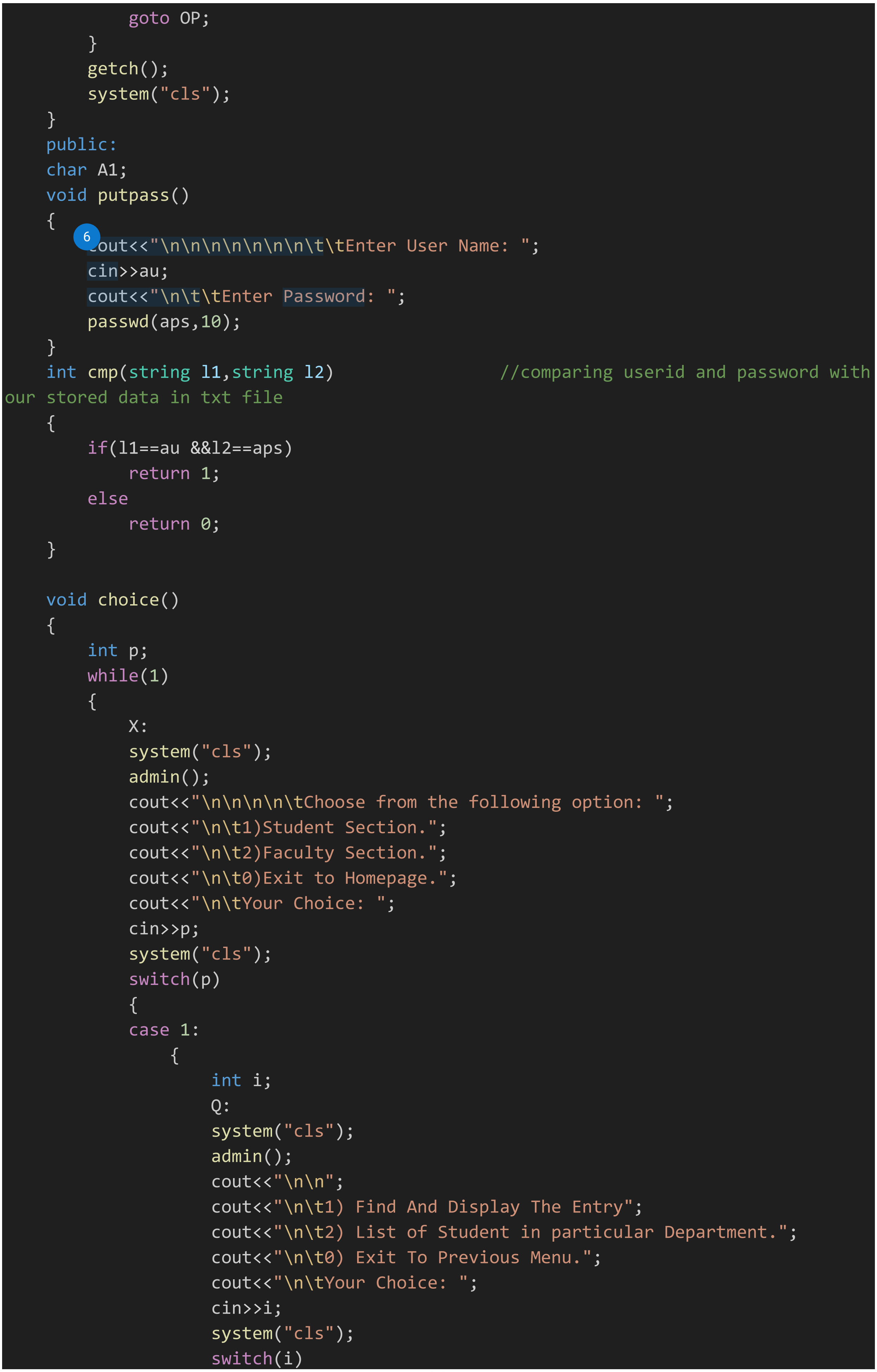


```
        admin();
        cout<<"\nName of Student   : "<<b1<<" "<<c1;
        cout<<"\nStudent ID        : "<<a1;
        cout<<"\nDepartment        : "<<e1;
        cout<<"\nInstitute         : "<<d1;
        cout<<"\nFees per year     : "<<h1;
        cout<<"\nAdmission Quota   : "<<i1;
    }
    else
    {
        cout<<"\n\n\n\tNo records found...\a";
    }
}
void getfdet()
{
    string file,a,b,d,c,e,f,g,h,i,j,k,l;
    ifstream fp;
    bool flag=0;
    file="Fdata.txt";
    fp.open(file.c_str());
    system("cls");
    admin();
    cout<<"\nEnter Faculty ID: ";
    cin>>l;
    while(fp>>a>>b>>c>>d>>e>>f>>g>>h>>i>>j)
    {
        if(a==l)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        system("cls");
        admin();
        cout<<"\nName of Faculty            : "<<b<<" "<<c;
        cout<<"\nFaculty ID                 : "<<a;
        cout<<"\nQualification              : "<<f;
        cout<<"\nDesignation                : "<<h;
        cout<<"\nInstitute:                 : "<<e;
        cout<<"\nName of Department         : "<<d;
        cout<<"\nSubjects of specializations : "<<g;
        cout<<"\nExperience                 : "<<i<<"years";
        cout<<"\nBasic Payroll              : "<<j<<"Rs./Month";
    }
    else
    {
        cout<<"\n\n\n\tNo such record found...\a";
    }
}

void getfac()
{
```

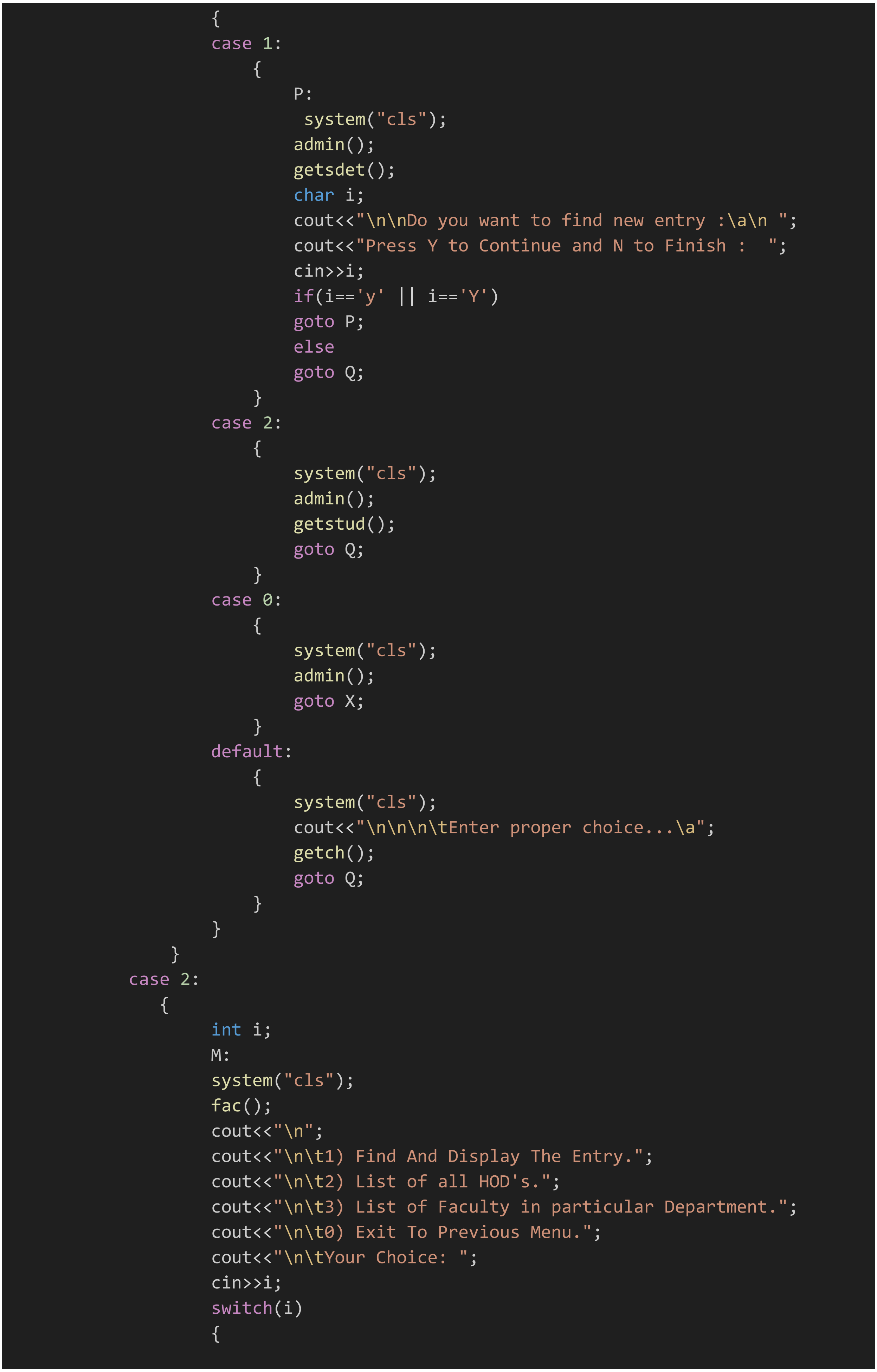


```
    OP1:
    flush(cout);
        system("cls");
        admin();
    cout<<"\n\nEnter Department: ";
    cin>>deptf;
    fstream fp;
    string a,b,c,d,e,f,g,h,file;
    system("cls");
    fac();

    cout<<endl<<endl;
    cout<<"   ID  "<<"       NAME    "<<"      DEPARTMENT "<<"INSTITUTE"<<"
DESIGNATION"<<endl<<endl;
          if(deptf=="ce" || deptf=="CE")
        {
            fstream f1;
            file="F_CE.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e>>f)
            {
                b=b+" "+c;
                cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(10)<<d<<" "<<setw(7)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
        else if(deptf=="it" || deptf=="IT")
        {
            fstream f1;
            file="F_IT.txt";
            f1.open(file.c_str());
             while(f1>>a>>b>>c>>d>>e>>f)
            {
                 b=b+" "+c;
                 cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(10)<<d<<" "<<setw(7)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
        else if(deptf=="ee" || deptf=="EE")
        {
            fstream f1;
            file="F_EE.txt";
            f1.open(file.c_str());
             while(f1>>a>>b>>c>>d>>e>>f)
            {
                b=b+" "+c;
                cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(10)<<d<<" "<<setw(7)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
        else if(deptf=="cl" || deptf=="CL")
        {
            fstream f1;
            file="F_CL.txt";
            f1.open(file.c_str());
             while(f1>>a>>b>>c>>d>>e>>f)
            {
                 b=b+" "+c;
                 cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(10)<<d<<" "<<setw(7)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
        else if(deptf=="me" || deptf=="ME")
        {
            fstream f1;
         file="F_ME.txt";
         f1.open(file.c_str());
             while(f1>>a>>b>>c>>d>>e>>f)
            {
                 b=b+" "+c;
                 cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(10)<<d<<" "<<setw(7)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
        else if(deptf=="depstar" || deptf=="DEPSTAR" )
        {
            fstream f1;
         file="F_DEP.txt";
         f1.open(file.c_str());
             while(f1>>a>>b>>c>>d>>e>>f)
            {
                 b=b+" "+c;
                 cout<<setiosflags(ios::left)<<" "<<setw(7)<<a<<" "<<setw(20)<<b<<"
"<<setw(9)<<d<<" "<<setw(8)<<e<<" "<<setw(13)<<f<<endl;
            }
        f1.close();
        }
         else
        {
            system("cls");
            cout<<"\n\n\n\tEnter proper department...\a";
            getch();
            system("cls");
            goto OP1;
        }
        getch();
        system("cls");
}
void getstud()
{
    OP:
        system("cls");
        admin();
    cout<<"\n\nEnter Department: ";
    cin>>depts;
    fstream fp;
    string a,b,c,d,e,file;
    system("cls");
    stud();
    cout<<endl<<endl;
    cout<<"   ID  "<<"     NAME    "<<"      INSTITUE "<<" DEPARTMENT"<<endl<<endl;
     if(depts=="ce" || depts=="CE")
        {
            fstream f1;
            file="S_CE.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
                b=b+" "+c;
                cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(18)<<b<<"
"<<setw(10)<<d<<setw(11)<<e<<endl;
            }
            f1.close();
        }
        else if(depts=="it" || depts=="IT")
        {
            fstream f1;
            file="S_IT.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
               cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else if(depts=="ee" || depts=="EE")
        {
            fstream f1;
            file="S_EE.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
                cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else if(depts=="cl" || depts=="CL")
        {
            fstream f1;
            file="S_CL.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
               cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
```

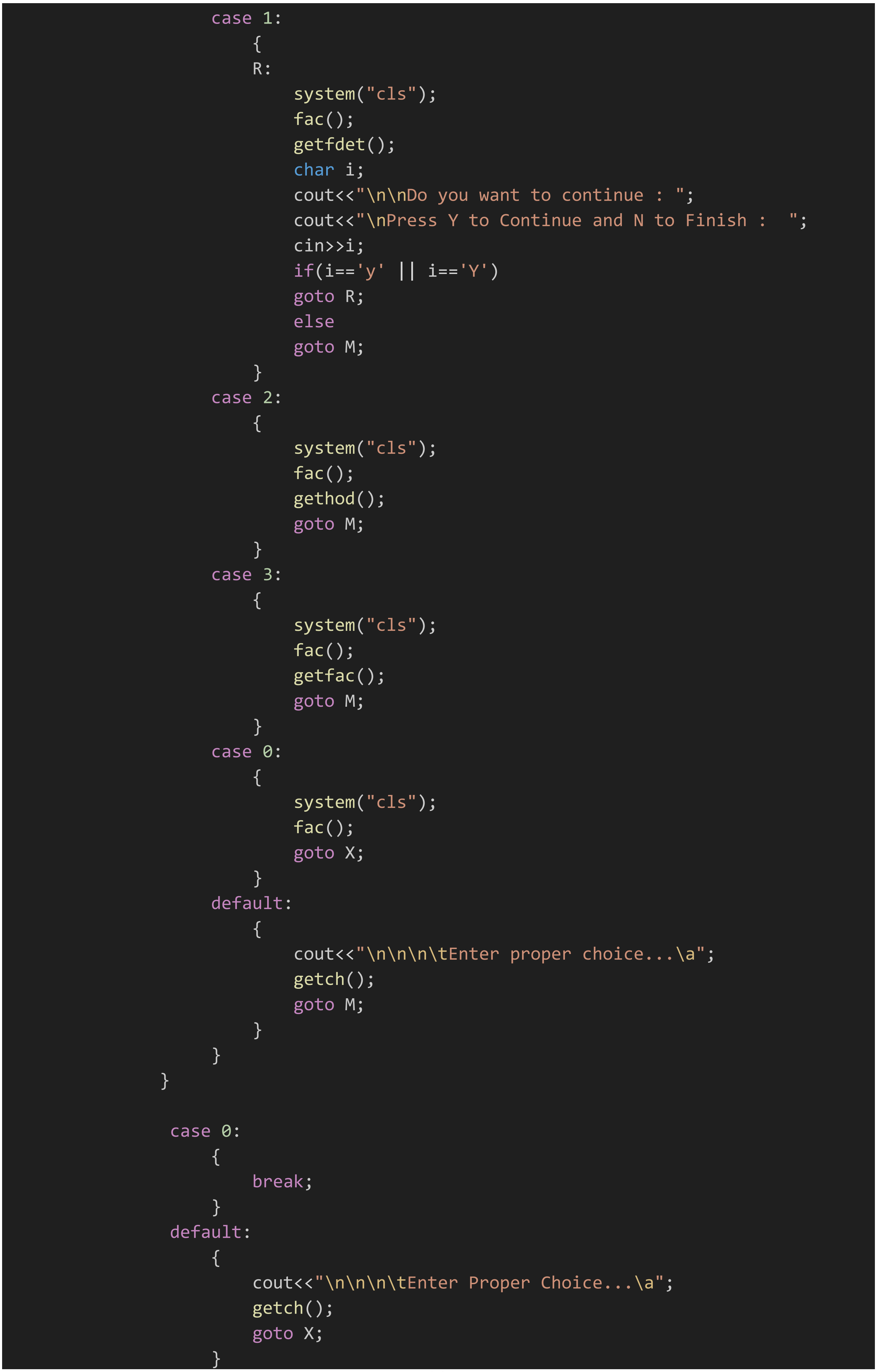
```
            }
        f1.close();
        }
        else if(depts=="me" || depts=="ME")
        {
            fstream f1;
            file="S_ME.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
                cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else if(depts=="dce" || depts=="DCE")
        {
            fstream f1;
            file="S_DCE.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
               cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else if(depts=="dcs" || depts=="DCS")
        {
            fstream f1;
            file="S_DCS.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
                cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else if(depts=="dit" || depts=="DIT")
        {
            fstream f1;
            file="S_DIT.txt";
            f1.open(file.c_str());
            while(f1>>a>>b>>c>>d>>e)
            {
                cout<<setiosflags(ios::left)<<setw(7)<<a<<" "<<setw(10)<<b<<"
"<<setw(10)<<c<<" "<<setw(10)<<d<<setw(11)<<e<<endl;
            }
        f1.close();
        }
        else
        {
            cout<<"Enter proper department...";
```



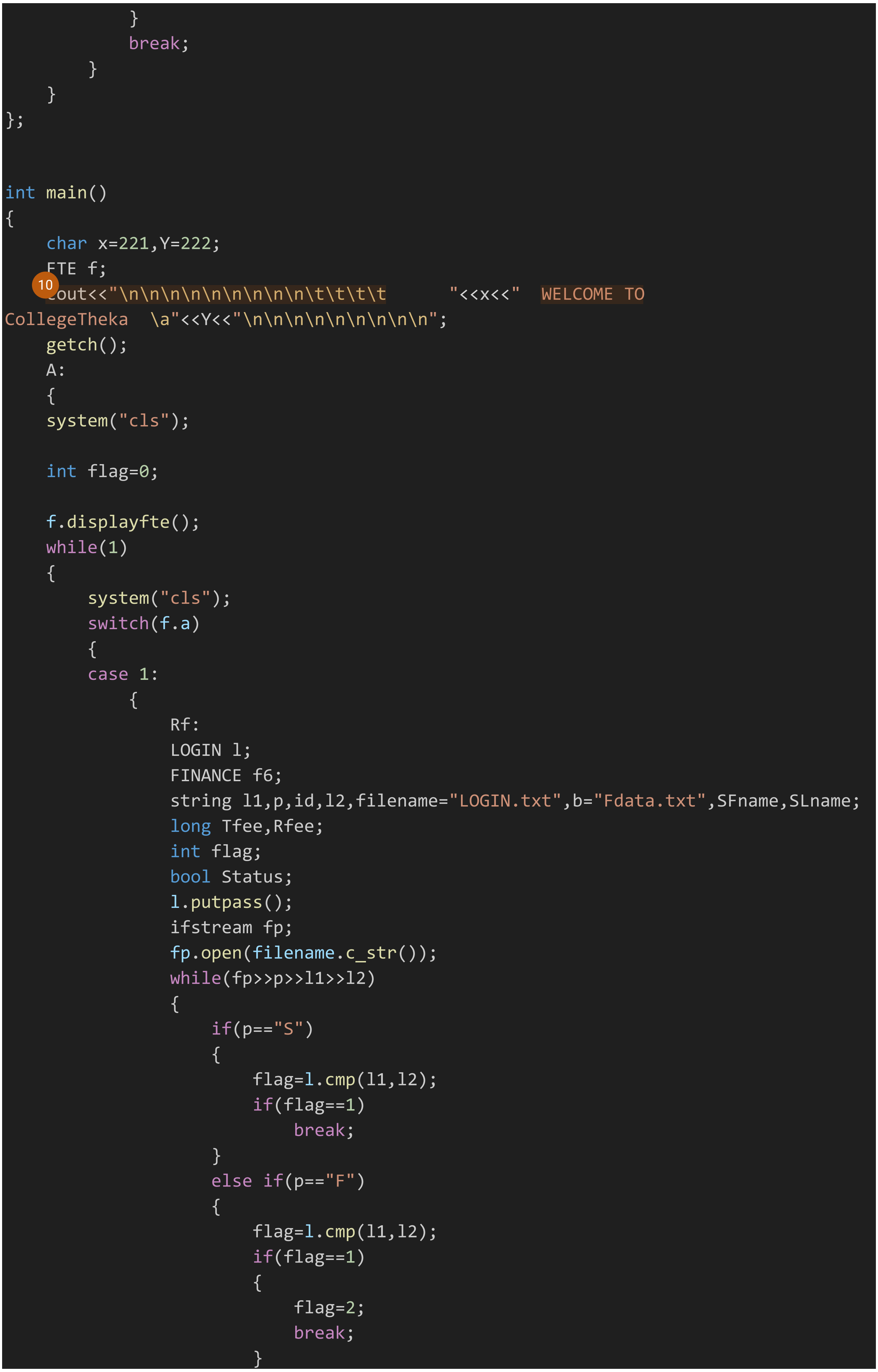
```
            goto OP;
        }
        getch();
        system("cls");
    }
    public:
    char A1;
    void putpass()
    {
        cout<<"\n\n\n\n\n\n\n\t\tEnter User Name: ";
        cin>>au;
        cout<<"\n\t\tEnter Password: ";
        passwd(aps,10);
    }
    int cmp(string l1,string l2)                //comparing userid and password with
our stored data in txt file
    {
        if(l1==au &&l2==aps)
            return 1;
        else
            return 0;
    }

    void choice()
    {
        int p;
        while(1)
        {
            X:
            system("cls");
            admin();
            cout<<"\n\n\n\n\tChoose from the following option: ";
            cout<<"\n\t1)Student Section.";
            cout<<"\n\t2)Faculty Section.";
            cout<<"\n\t0)Exit to Homepage.";
            cout<<"\n\tYour Choice: ";
            cin>>p;
            system("cls");
            switch(p)
            {
            case 1:
                {
                    int i;
                    Q:
                    system("cls");
                    admin();
                    cout<<"\n\n";
                    cout<<"\n\t1) Find And Display The Entry";
                    cout<<"\n\t2) List of Student in particular Department.";
                    cout<<"\n\t0) Exit To Previous Menu.";
                    cout<<"\n\tYour Choice: ";
                    cin>>i;
                    system("cls");
                    switch(i)
```



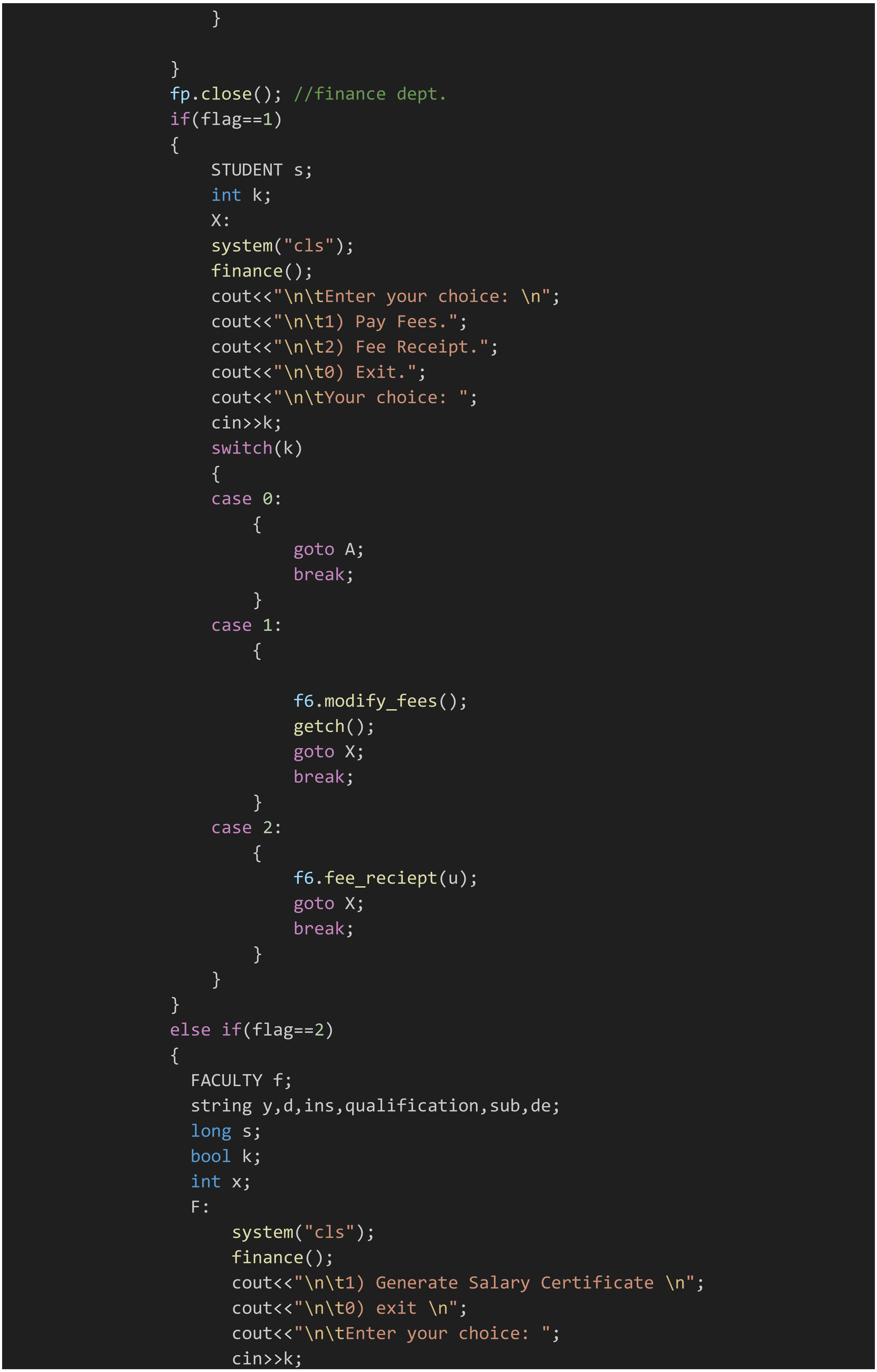
```
                {
                case 1:
                    {
                        P:
                         system("cls");
                        admin();
                        getsdet();
                        char i;
                        cout<<"\n\nDo you want to find new entry :\a\n ";
                        cout<<"Press Y to Continue and N to Finish :  ";
                        cin>>i;
                        if(i=='y' || i=='Y')
                        goto P;
                        else
                        goto Q;
                    }
                case 2:
                    {
                        system("cls");
                        admin();
                        getstud();
                        goto Q;
                    }
                case 0:
                    {
                        system("cls");
                        admin();
                        goto X;
                    }
                default:
                    {
                        system("cls");
                        cout<<"\n\n\n\tEnter proper choice...\a";
                        getch();
                        goto Q;
                    }
                }
            }
        case 2:
           {
                int i;
                M:
                system("cls");
                fac();
                cout<<"\n";
                cout<<"\n\t1) Find And Display The Entry.";
                cout<<"\n\t2) List of all HOD's.";
                cout<<"\n\t3) List of Faculty in particular Department.";
                cout<<"\n\t0) Exit To Previous Menu.";
                cout<<"\n\tYour Choice: ";
                cin>>i;
                switch(i)
                {
```



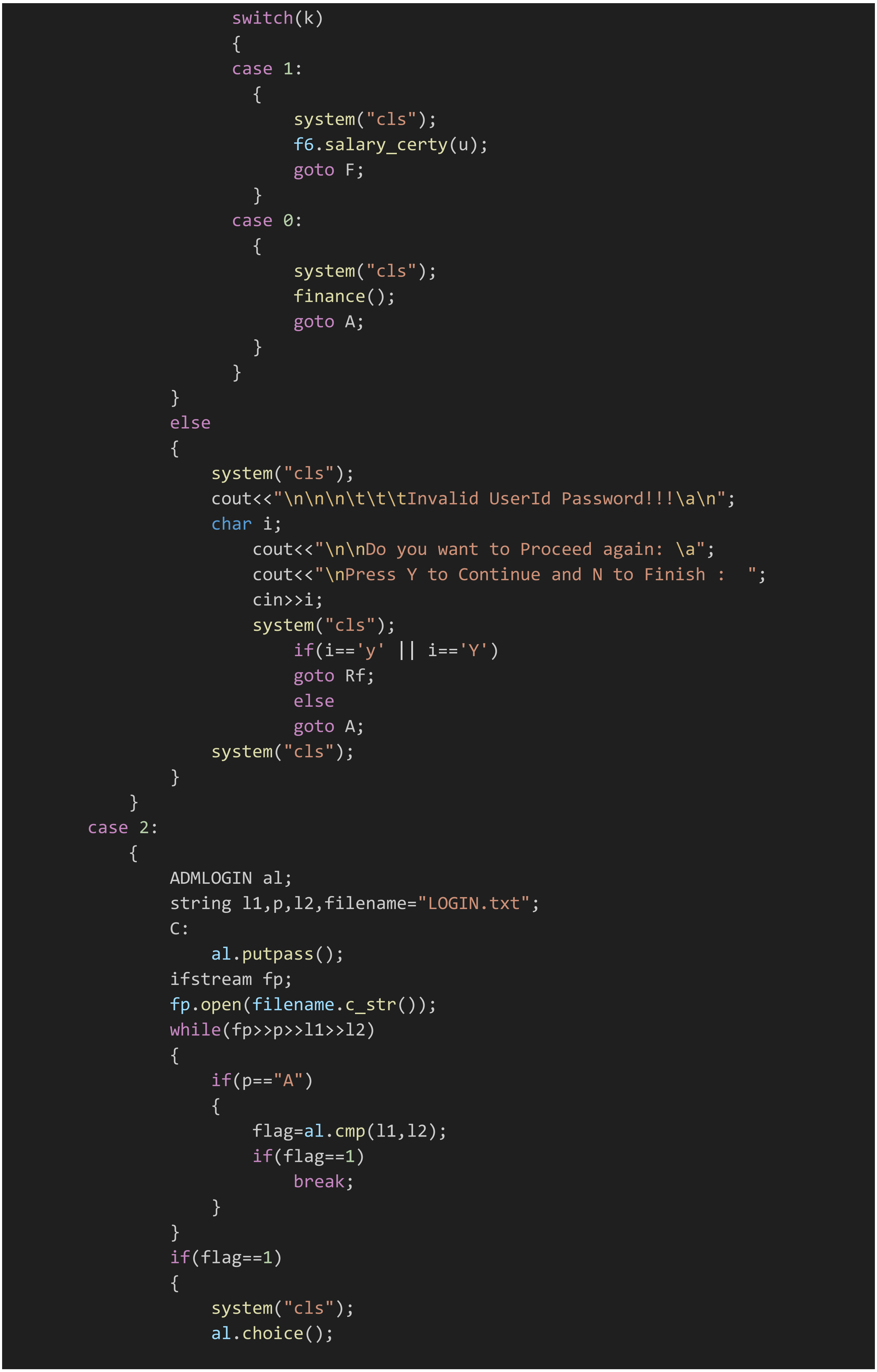
```
     case 1:
         {
         R:
             system("cls");
             fac();
             getfdet();
             char i;
             cout<<"\n\nDo you want to continue : ";
             cout<<"\nPress Y to Continue and N to Finish :  ";
             cin>>i;
             if(i=='y' || i=='Y')
             goto R;
             else
             goto M;
         }
     case 2:
         {
             system("cls");
             fac();
             gethod();
             goto M;
         }
     case 3:
         {
             system("cls");
             fac();
             getfac();
             goto M;
         }
     case 0:
         {
             system("cls");
             fac();
             goto X;
         }
     default:
         {
             cout<<"\n\n\n\tEnter proper choice...\a";
             getch();
             goto M;
         }
     }
}

 case 0:
     {
         break;
     }
 default:
     {
         cout<<"\n\n\n\tEnter Proper Choice...\a";
         getch();
         goto X;
     }
```



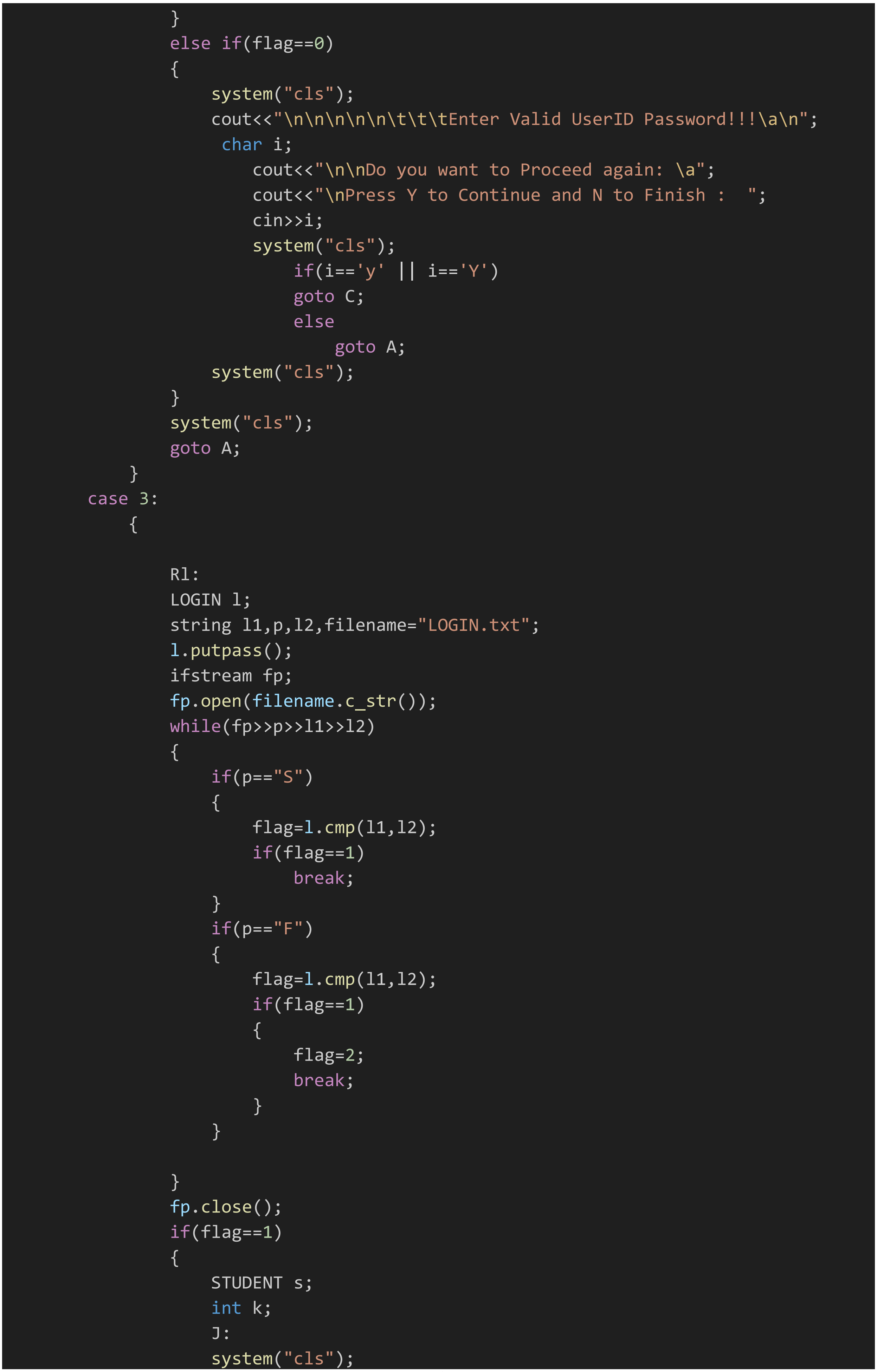
```
            }
            break;
        }
    }
};


int main()
{
    char x=221,Y=222;
    FTE f;
    cout<<"\n\n\n\n\n\n\n\n\n\t\t\t\t       "<<x<<"  WELCOME TO
CollegeTheka  \a"<<Y<<"\n\n\n\n\n\n\n\n\n";
    getch();
    A:
    {
    system("cls");

    int flag=0;

    f.displayfte();
    while(1)
    {
        system("cls");
        switch(f.a)
        {
        case 1:
            {
                Rf:
                LOGIN l;
                FINANCE f6;
                string l1,p,id,l2,filename="LOGIN.txt",b="Fdata.txt",SFname,SLname;
                long Tfee,Rfee;
                int flag;
                bool Status;
                l.putpass();
                ifstream fp;
                fp.open(filename.c_str());
                while(fp>>p>>l1>>l2)
                {
                    if(p=="S")
                    {
                        flag=l.cmp(l1,l2);
                        if(flag==1)
                            break;
                    }
                    else if(p=="F")
                    {
                        flag=l.cmp(l1,l2);
                        if(flag==1)
                        {
                            flag=2;
                            break;
                        }
```



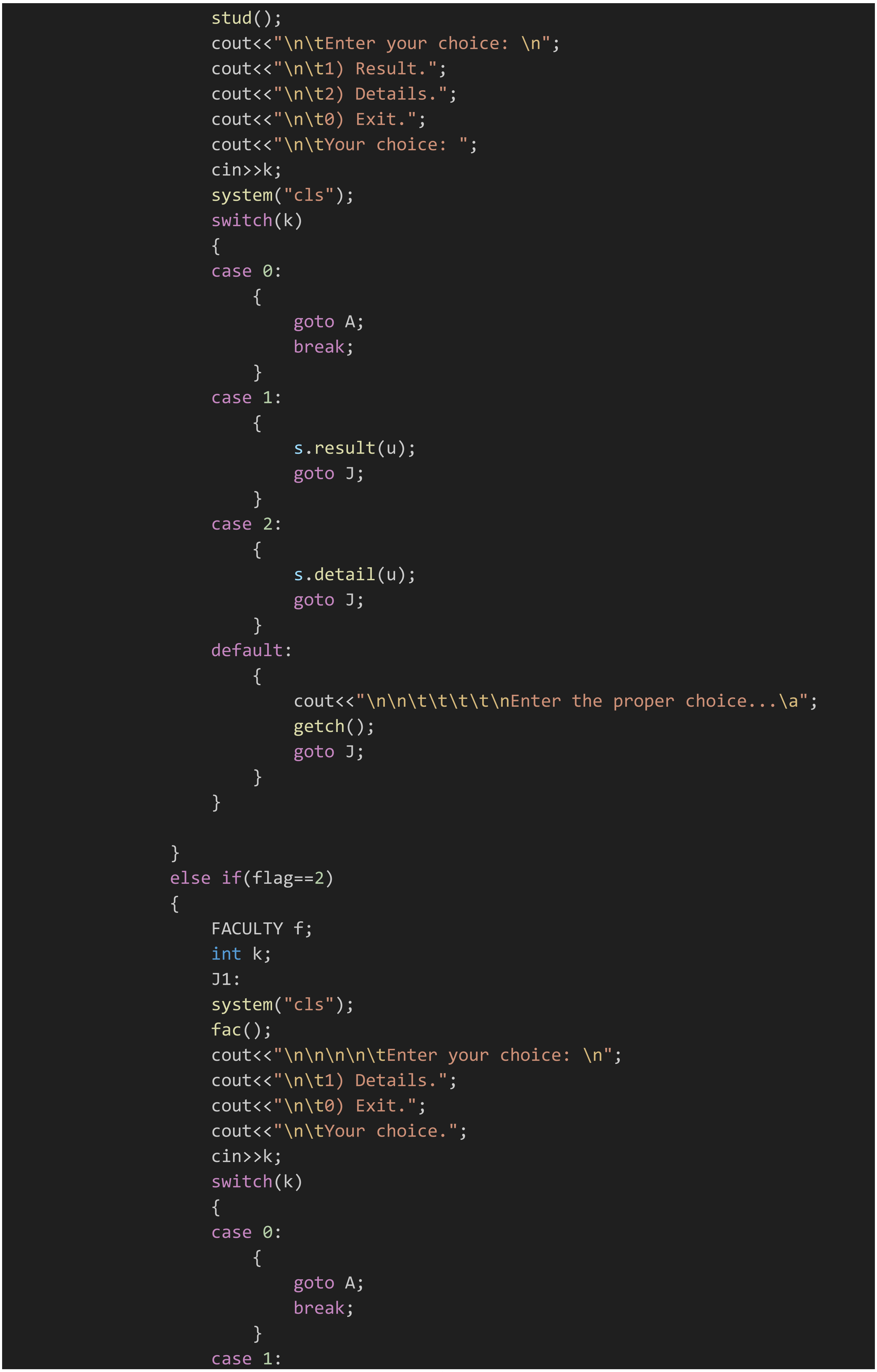
```
    }

}
fp.close(); //finance dept.
if(flag==1)
{
    STUDENT s;
    int k;
    X:
    system("cls");
    finance();
    cout<<"\n\tEnter your choice: \n";
    cout<<"\n\t1) Pay Fees.";
    cout<<"\n\t2) Fee Receipt.";
    cout<<"\n\t0) Exit.";
    cout<<"\n\tYour choice: ";
    cin>>k;
    switch(k)
    {
    case 0:
        {
            goto A;
            break;
        }
    case 1:
        {

            f6.modify_fees();
            getch();
            goto X;
            break;
        }
    case 2:
        {
            f6.fee_reciept(u);
            goto X;
            break;
        }
    }
}
else if(flag==2)
{
  FACULTY f;
  string y,d,ins,qualification,sub,de;
  long s;
  bool k;
  int x;
  F:
      system("cls");
      finance();
      cout<<"\n\t1) Generate Salary Certificate \n";
      cout<<"\n\t0) exit \n";
      cout<<"\n\tEnter your choice: ";
      cin>>k;
```



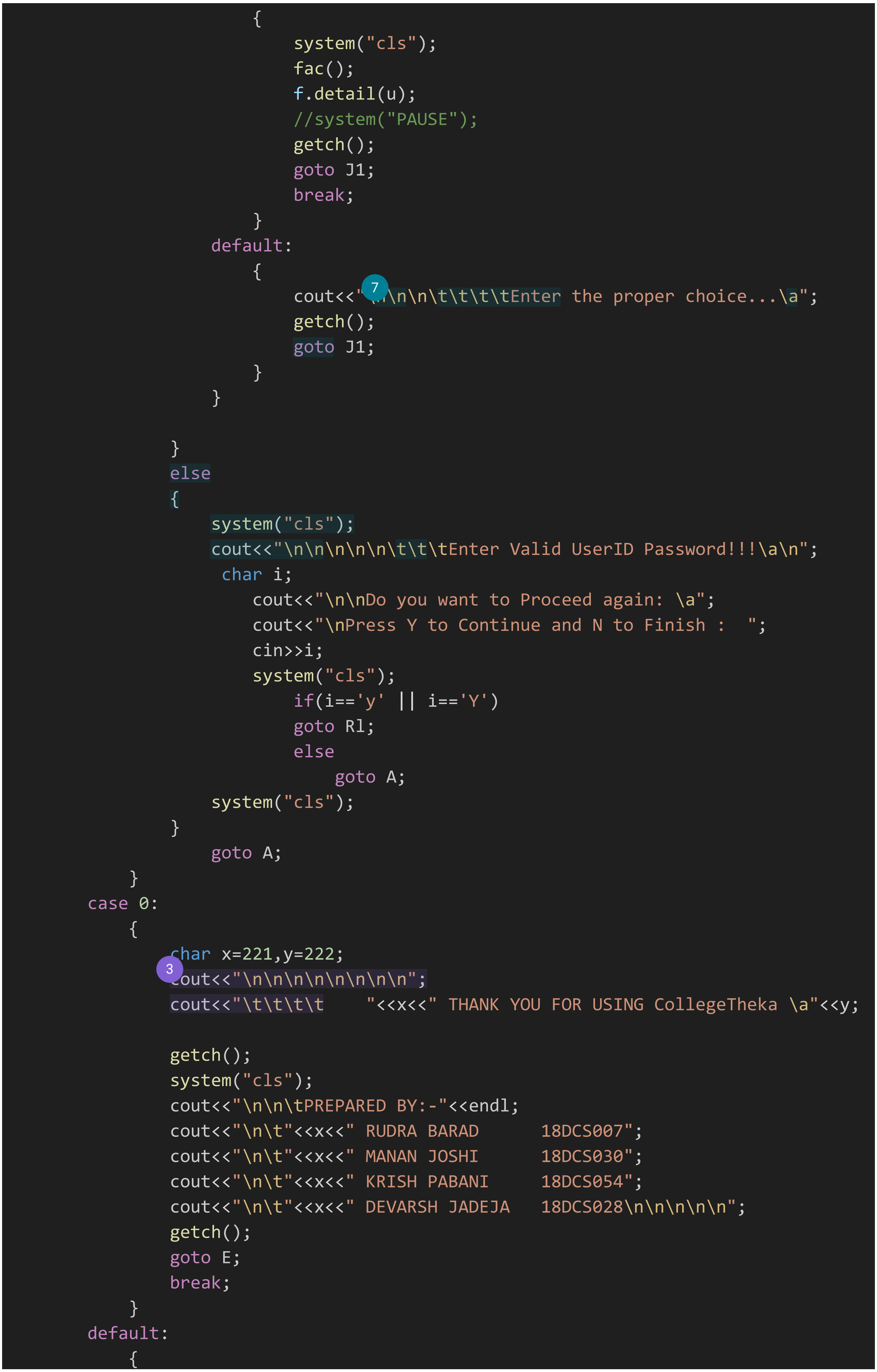
```
                      switch(k)
                      {
                      case 1:
                        {
                            system("cls");
                            f6.salary_certy(u);
                            goto F;
                        }
                      case 0:
                        {
                            system("cls");
                            finance();
                            goto A;
                        }
                      }
                }
                else
                {
                    system("cls");
                    cout<<"\n\n\n\t\t\tInvalid UserId Password!!!\a\n";
                    char i;
                        cout<<"\n\nDo you want to Proceed again: \a";
                        cout<<"\nPress Y to Continue and N to Finish :  ";
                        cin>>i;
                        system("cls");
                            if(i=='y' || i=='Y')
                            goto Rf;
                            else
                            goto A;
                    system("cls");
                }
            }
        case 2:
            {
                ADMLOGIN al;
                string l1,p,l2,filename="LOGIN.txt";
                C:
                    al.putpass();
                ifstream fp;
                fp.open(filename.c_str());
                while(fp>>p>>l1>>l2)
                {
                    if(p=="A")
                    {
                        flag=al.cmp(l1,l2);
                        if(flag==1)
                            break;
                    }
                }
                if(flag==1)
                {
                    system("cls");
                    al.choice();
```



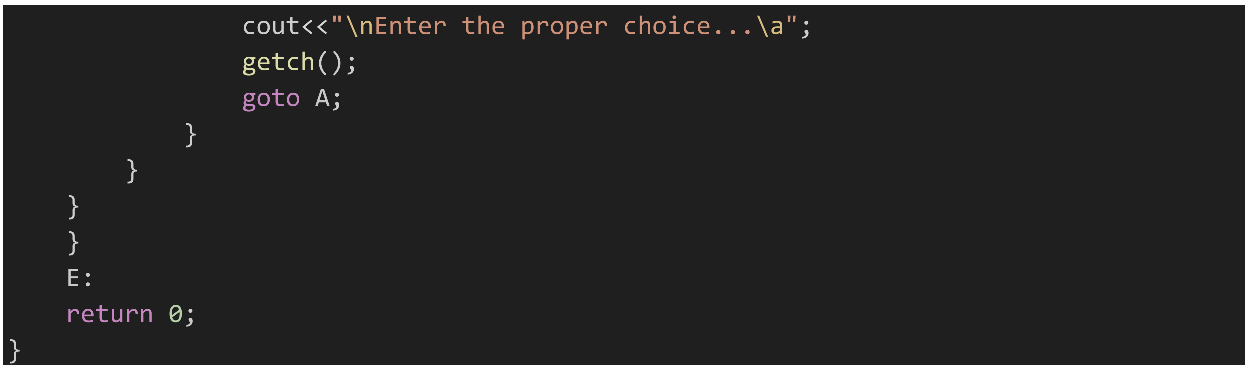
```
                }
                else if(flag==0)
                {
                    system("cls");
                    cout<<"\n\n\n\n\n\t\t\tEnter Valid UserID Password!!!\a\n";
                     char i;
                        cout<<"\n\nDo you want to Proceed again: \a";
                        cout<<"\nPress Y to Continue and N to Finish :  ";
                        cin>>i;
                        system("cls");
                            if(i=='y' || i=='Y')
                            goto C;
                            else
                                goto A;
                    system("cls");
                }
                system("cls");
                goto A;
            }
        case 3:
            {

                Rl:
                LOGIN l;
                string l1,p,l2,filename="LOGIN.txt";
                l.putpass();
                ifstream fp;
                fp.open(filename.c_str());
                while(fp>>p>>l1>>l2)
                {
                    if(p=="S")
                    {
                        flag=l.cmp(l1,l2);
                        if(flag==1)
                            break;
                    }
                    if(p=="F")
                    {
                        flag=l.cmp(l1,l2);
                        if(flag==1)
                        {
                            flag=2;
                            break;
                        }
                    }

                }
                fp.close();
                if(flag==1)
                {
                    STUDENT s;
                    int k;
                    J:
                    system("cls");
```



```
    stud();
    cout<<"\n\tEnter your choice: \n";
    cout<<"\n\t1) Result.";
    cout<<"\n\t2) Details.";
    cout<<"\n\t0) Exit.";
    cout<<"\n\tYour choice: ";
    cin>>k;
    system("cls");
    switch(k)
    {
    case 0:
        {
            goto A;
            break;
        }
    case 1:
        {
            s.result(u);
            goto J;
        }
    case 2:
        {
            s.detail(u);
            goto J;
        }
    default:
        {
            cout<<"\n\n\t\t\t\t\nEnter the proper choice...\a";
            getch();
            goto J;
        }
    }

}
else if(flag==2)
{
    FACULTY f;
    int k;
    J1:
    system("cls");
    fac();
    cout<<"\n\n\n\n\tEnter your choice: \n";
    cout<<"\n\t1) Details.";
    cout<<"\n\t0) Exit.";
    cout<<"\n\tYour choice.";
    cin>>k;
    switch(k)
    {
    case 0:
        {
            goto A;
            break;
        }
    case 1:
```



```
                {
                    system("cls");
                    fac();
                    f.detail(u);
                    //system("PAUSE");
                    getch();
                    goto J1;
                    break;
                }
            default:
                {
                    cout<<"\n\n\n\t\t\t\tEnter the proper choice...\a";
                    getch();
                    goto J1;
                }
            }

        }
        else
        {
            system("cls");
            cout<<"\n\n\n\n\n\t\t\tEnter Valid UserID Password!!!\a\n";
             char i;
                cout<<"\n\nDo you want to Proceed again: \a";
                cout<<"\nPress Y to Continue and N to Finish :  ";
                cin>>i;
                system("cls");
                    if(i=='y' || i=='Y')
                    goto Rl;
                    else
                        goto A;
            system("cls");
        }
            goto A;
    }
case 0:
    {
        char x=221,y=222;
        cout<<"\n\n\n\n\n\n\n\n";
        cout<<"\t\t\t\t    "<<x<<" THANK YOU FOR USING CollegeTheka \a"<<y;

        getch();
        system("cls");
        cout<<"\n\n\tPREPARED BY:-"<<endl;
        cout<<"\n\t"<<x<<" RUDRA BARAD      18DCS007";
        cout<<"\n\t"<<x<<" MANAN JOSHI      18DCS030";
        cout<<"\n\t"<<x<<" KRISH PABANI     18DCS054";
        cout<<"\n\t"<<x<<" DEVARSH JADEJA   18DCS028\n\n\n\n\n";
        getch();
        goto E;
        break;
    }
default:
    {
```



```
                cout<<"\nEnter the proper choice...\a";
                getch();
                goto A;
            }
        }
    }
    }
    E:
    return 0;
}
```

Chapter 7: **Results**

The "CollegeTheka" project aspires to create an extensive College Management System, and the expected outcomes of the development process encompass several key aspects.

1. Efficient Administrative Processes:
The primary goal is to streamline administrative tasks. The administrative module of the system intends to equip administrators with efficient tools for managing student and faculty records, course details, and financial transactions. The expected results involve a more organized and systematic approach to administrative operations, reducing manual efforts, and minimizing errors in record-keeping.

2. Enhanced User Experience:
The development of a user-friendly interface, be it through command-line or graphical means, aims to enhance the overall user experience for administrators, students, and faculty members. Intuitive design, easy navigation, and clear communication channels are anticipated to contribute to a positive and efficient interaction with the College Management System.

3. Improved Data Accessibility for Students:
For students, the system aims to provide enhanced access to personal and academic information. This includes features such as viewing grades, attendance records, and course schedules. The expected results involve a user-centric approach, empowering students with readily available and easily understandable information, contributing to a more engaged and informed student body.

4. Efficient Faculty Management Tools:
Faculty members are expected to benefit from tools that simplify grade management, attendance tracking, and course material handling. The anticipated results showcase an enhanced workflow for faculty, reducing administrative burdens and allowing more focus on the core aspects of teaching and research.

5. Secure System Architecture:
The implementation of robust security measures, including authentication mechanisms, data encryption, and access controls, aims to ensure the security of user data and the overall system. The expected results demonstrate a secure environment that safeguards sensitive information, instilling confidence in stakeholders regarding the integrity of the College Management System.

6. Effective Communication Channels:
The integration of communication features within the system is anticipated to improve collaboration and information exchange between students, faculty, and administrators. The expected results reflect a more connected and communicative college community, with

announcements, messaging, and notifications enhancing overall communication efficiency.

7. Scalable and Maintainable Codebase:
The use of C++ and modular design principles aims to create a scalable and maintainable codebase. The expected results demonstrate a well-structured and easily extensible system, facilitating future updates, enhancements, and the addition of new features.

8. Thorough Testing and Quality Assurance:
The comprehensive testing conducted during the development process is expected to result in a reliable and robust system. The expected results showcase data accuracy, security robustness, and system reliability, providing stakeholders with confidence in the performance and stability of the CollegeTheka application.

In conclusion, the expected outcomes of the "CollegeTheka" project encompass not only the successful implementation of functional requirements but also the positive impact on administrative efficiency, user experience, and overall system reliability. The project's outcomes aim to address the specific needs of administrators, students, and faculty members, contributing to an improved and well-managed college environment.

Chapter 8: **Conclusion**

In conclusion, the "CollegeTheka" project represents a pivotal initiative in the domain of college administration, aiming to address intricate challenges through the development of a holistic College Management System. The envisioned system, constructed on the pillars of the C++ programming language, a relational database management system, and user-friendly interfaces, is poised to bring about transformative outcomes across various facets of college operations.

The project's fundamental objectives, encompassing streamlined administrative processes, enriched user experiences, and improved communication channels, are designed to establish a dynamic and efficient environment for administrators, students, and faculty members. By automating administrative tasks and centralizing operations, the system is anticipated to significantly reduce manual workload, mitigate errors in record-keeping, and empower administrators with effective tools for decision-making.

For students, the system pledges enhanced access to personal and academic information, nurturing a more informed and engaged student community. Faculty members are expected to benefit from streamlined management tools, allowing them to prioritize teaching and research. Robust security measures are incorporated to ensure the confidentiality and integrity of user data, contributing to a secure system architecture.

The project's commitment to scalability, maintainability, and rigorous testing underscores a forward-looking approach. The results seek to yield a functional and reliable codebase that can readily adapt to future enhancements and modifications. Furthermore, the collaborative and communicative aspects of the system aim to strengthen community ties within the college, fostering a more connected and informed environment.

In essence, the "CollegeTheka" project aspires to usher in positive changes in the daily operations of a college, providing a comprehensive solution to the diverse needs of stakeholders. The anticipated outcomes align with the project's vision of an efficient, user-centric, and secure College Management System that contributes to the overall success and advancement of the academic institution. As the system evolves into an integral part of the college ecosystem, the envisioned results are poised to positively impact the way administrative tasks are conducted and information is managed within the college environment.

Chapter 9: **Bibliography**

1. The C++ Programming Language (4th Edition) By Bjarne Stroustrup
2. Practical C++ Programming (2nd Edition) By Steve Oualline
3. https://youtube.com/playlist?list=PLDzeHZWIZsTryvtXdMr6rPh4IDexB5NIA&si=EcZDhLSh-UZHRreD
4. https://youtube.com/playlist?list=PLxgZQoSe9cg0df_GxVjz3DD_Gck5tMXAd&si=RDs9HZ8p7gjdkvj2

7% Overall Similarity

Top sources found in the following databases:

- 5% Internet database
- Crossref database
- 4% Submitted Works database
- 0% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**Fareham College on 2023-10-15** Submitted works	2%
2	**zh.scribd.com** Internet	1%
3	**coursehero.com** Internet	<1%
4	**fasteducationlearning.blogspot.com** Internet	<1%
5	**Middle East College of Information Technology on 2020-07-09** Submitted works	<1%
6	**Liverpool John Moores University on 2021-05-13** Submitted works	<1%
7	**github.com** Internet	<1%
8	**kostkecilku.blogspot.com** Internet	<1%

9	University of Wales Institute, Cardiff on 2023-09-05 Submitted works	<1%
10	South Bank University on 2021-06-03 Submitted works	<1%
11	University of Wales central institutions on 2023-01-16 Submitted works	<1%