**ATM Simulator**

**Submitted by**

|  |  |
| --- | --- |
| **Name** | **Arghyadip Chowdhury** |
| **Contact Number** | **+917044433043** |
| **Roll Number** | **23** |
| **Section** | **A** |
| **Subject** | **Programming for Problem Solving using C** |
| **Subject Code** | **ESC103(Pr.)** |
| **Department** | **Basic Science and Humanities** |
| **Year** | **B.Tech 1st Year** |

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITITES**

**INSTITUTE OF ENGINEERING & MANAGEMENT, KOLKATA**

**KEYS**

A program that manages student data, including their names , roll no., marks ,and grades in c language  
**Variables:**

* **MAX\_STUDENTS**: a constant integer that specifies the maximum number of students that can be managed by the program.
* **struct Student**: a user-defined data type that represents a student and contains the following fields:
  + **name**: a character array that stores the name of the student.
  + **rollNo**: an integer that stores the roll number of the student.
  + **marks**: an integer that stores the marks obtained by the student.
  + **grade**: a character that stores the grade obtained by the student based on their marks.
* **numStudents**: an integer that stores the number of students entered by the user.
* **i**: an integer used as a loop counter for iterating over the students array.
* **students**: an array of Student structs that stores the data for each student.
* **calculateGrade**: a function that takes an integer as input (marks) and returns a character (grade).
* **marks**: an integer that stores the marks obtained by each student.
* **name**: a character array that stores the name of each student.
* **rollNo**: an integer that stores the roll number of each student.
* **grade**: a character that stores the grade obtained by each student.

1. **Functions: char calculateGrade(int marks)**: This function takes an integer argument **marks** as input and returns a character that represents the grade obtained by the student based on the marks. The function uses if-else statements to determine the grade and returns the appropriate character.
2. **int main()**: This function is the entry point of the program. It declares the variables used in the program, including the **struct Student** data type. It prompts the user to enter the number of students and then gets the data for each student using a loop that iterates over the students array. Inside the loop, it calls the **calculateGrade** function to determine the grade of each student. After getting all the data, the function then displays the student data in a tabular format using another loop.

Both of these functions are defined within the same file as the main function. The **calculateGrade** function is called from within the main function.

**Files and Datasets:** The program in this code example does not make use of files or databases.

Files and databases are used to store data persistently, which means that the data can be accessed even after the program is closed or the computer is turned off. In contrast, the data in this program is stored in memory, which means that it is lost when the program terminates.

If we want to use files to store the data, you could modify the program to write the student data to a file, and read the data from the file when the program starts up. This would allow we to store the data persistently and avoid losing it when the program terminates.

If we want to use a database to store the data, we could modify the program to use a database library such as SQLite or MySQL. The program could connect to the database and store the student data in a table. This would allow us to perform more complex queries on the data and manage it more efficiently.

**REPORT**

1. Introduction: The Student Data Management System is a program written in C language that allows users to manage student data such as their names, roll no., marks, and grades. The program uses a struct to store the data and provides functions for calculating grades and displaying the data in a tabular format. The goal of this project is to create a user-friendly and efficient system for managing student data.
2. Program Design: The program is designed using a struct called "Student" that contains the following fields:

* Name: a string that stores the name of the student.
* Roll No.: an integer that stores the roll number of the student.
* Marks: an integer that stores the marks obtained by the student.
* Grade: a character that stores the grade obtained by the student based on their marks.

1. The program uses a loop to get the data for each student from the user and stores it in an array of Student structs. After getting the data, the program calculates the grade for each student using a separate function called "calculateGrade" that takes the marks as input and returns the grade as output. Finally, the program displays the student data in a tabular format.
2. Testing: The program was tested with various inputs to ensure that it works correctly. The following are some examples of the inputs and outputs of the program:

Example 1: Enter the number of students: 3 Enter the name of student 1: John Enter the roll no. of student 1: 101 Enter the marks of student 1: 85 Enter the name of student 2: Mary Enter the roll no. of student 2: 102 Enter the marks of student 2: 92 Enter the name of student 3: Peter Enter the roll no. of student 3: 103 Enter the marks of student 3

**CODE** *#include <stdio.h>*

*// Define the maximum number of students*

*#define MAX\_STUDENTS 100*

*// Define a struct for storing student data*

*struct Student {*

*char name[50];*

*int rollNo;*

*int marks;*

*char grade;*

*};*

*// Define a function to calculate the grade based on marks*

*char calculateGrade(int marks) {*

*if (marks >= 90) {*

*return 'A';*

*} else if (marks >= 80) {*

*return 'B';*

*} else if (marks >= 70) {*

*return 'C';*

*} else if (marks >= 60) {*

*return 'D';*

*} else {*

*return 'F';*

*}*

*}*

*int main() {*

*// Declare an array to store student data*

*struct Student students[MAX\_STUDENTS];*

*// Get the number of students from the user*

*int numStudents;*

*printf("Enter the number of students: ");*

*scanf("%d", &numStudents);*

*// Loop through each student and get their data*

*for (int i = 0; i < numStudents; i++) {*

*printf("Enter the name of student %d: ", i+1);*

*scanf("%s", students[i].name);*

*printf("Enter the roll no. of student %d: ", i+1);*

*scanf("%d", &students[i].rollNo);*

*printf("Enter the marks of student %d: ", i+1);*

*scanf("%d", &students[i].marks);*

*// Calculate the grade based on marks*

*students[i].grade = calculateGrade(students[i].marks);*

*}*

*// Print the student data*

*printf("Name\tRoll No.\tMarks\tGrade\n");*

*for (int i = 0; i < numStudents; i++) {*

*printf("%s\t%d\t\t%d\t%c\n", students[i].name, students[i].rollNo, students[i].marks, students[i].grade);*

*}*

*return 0;*

*}*