

- Hostel room allotment system key features : -
 1. Room distribution accⁿ to academic year & stay time iteration
 2. Contains four Block (4 years stay)
 3. No. of floor can be accⁿ to permission
 4. No. of blocks given
 5. No. of floors also given
 6. Criteria of room allotment to students

→ Roll NO : eg → 21 CS10040

↓ ↓
 Current Department
 academic know
 year

Code must have current year information

→ Alternate Allot between 3rd year and 4th year

if ↓
 lost
 course is year
 4 year

→ 1st & 2nd years have its own block to make unity between each others and they should come to know each other.

* Inputs : -

1. Current Year
2. list of all Students with roll number & final year
3. No. of blocks
4. No. of floor
5. Occupancy of room (single, double sharing, triple)
6. No. of room in a floor

* Output : -

1. No. of students with Room number allotted

Raw Code :-

Function :-

1. Function to know the final year student

Input :-

- Current year student

- Students with Roll no. and name of stu.

Output :-

- List of student with Final year or not.

2. Function allotting alternate year like
21 → final
22 → final

3. Function allotting room serially

4. If allotted the return 1 → as matrix value

CS 4	CS 3	CS 4	CS 3	CS 4	CS 3
CS 4	CS 3	CS 4	CS 2	CS 1	CS 2
CS 1	CS 2	CS 1	CS 2	CS 1	CS 2

```
cpp > C: \hostelroomallotment.cpp > ...
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <map>
5 #include <sstream>
6 #include <algorithm>
7
8 using namespace std;
9
10 struct Student {
11     string rollNumber;
12     string name;
13     string year;
14 };
15
16 string getYear(int startYear) {
17     int currentYear = 2024;
18     int diff = currentYear - startYear;
19     switch (diff) {
20         case 0: return "First Year";
21         case 1: return "Second Year";
22         case 2: return "Pre-Final Year";
23         case 3: return "Final Year";
24         default: return "Unknown Year";
25     }
26 }
27
28 string allotRoom(vector<vector<bool>>& roomAvailability, int yearPriority, int roomsPerFloor) {
29     int totalFloors = roomAvailability.size();
30     int startFloor = totalFloors - yearPriority;
31     for (int floor = startFloor; floor >= 0; --floor) {
32         for (int room = 0; room < roomsPerFloor; ++room) {
33             if (!roomAvailability[floor][room]) {
34                 roomAvailability[floor][room] = true;
35                 return to_string(floor + 1) + to_string(room + 1).insert(0, 2 - to_string(room + 1).length(), '0');
36             }
37         }
38     }
39     return "No Room Available";
40 }
41
42 int main() {
43     ifstream inputFile("sample1.txt");
44     ofstream outputFile("output1.txt");
45
46     if (!inputFile.is_open() || !outputFile.is_open()) {
47         return 1;
48     }
49
50     vector<Student> students;
51     string line;
52
53     while (getline(inputFile, line)) {
54         if (!line.empty()) {
55             stringstream ss(line);
56             Student student;
57             ss >> student.rollNumber;
58             ss.ignore();
59             getline(ss, student.name);
60             int startYear = stoi(student.rollNumber.substr(0, 2)) + 2000;
61             student.year = getYear(startYear);
62             students.push_back(student);
63         }
64     }
65     inputFile.close();
66
67     map<string, int> yearPriority = {
68         {"Final Year", 1},
```

```
    {"Second Year", 2},
    {"First Year", 3},
    {"Pre-Final Year", 4}
}
```

```

42     int main() {
43         inputFile.close();
44
45         map<string, int> yearPriority = {
46             {"Final Year", 1},
47             {"Pre-Final Year", 2},
48             {"Second Year", 3},
49             {"First Year", 4}
50         };
51
52
53         sort(students.begin(), students.end(), [&](const Student& a, const Student& b) {
54             return yearPriority[a.year] < yearPriority[b.year];
55         });
56
57
58         int totalFloors, roomsPerFloor;
59         cin >> totalFloors >> roomsPerFloor;
60         vector<vector<bool>> roomAvailability(totalFloors, vector<bool>(roomsPerFloor, false));
61
62         for (const auto& student : students) {
63             string room = allotRoom(roomAvailability, yearPriority[student.year], roomsPerFloor);
64             outputFile << student.rollNumber << " " << student.name << "(" << student.year << ") - Room " << room << endl;
65         }
66
67         outputFile.close();
68         return 0;
69     }
70

```

Input file(.txt)

```

1  23CE10001 Aakarsh Raaj
2  22CE10002 Abhishek Burnwal
3  22CE10003 Adrij Bhattacharya
4  21CE10004 Alok Anand
5  22CE10005 Aman Meena
6  24CE10006 Ameya Neeraj Tambat

```

Output(.txt)

```

1  21CE10004 Alok Anand (Final Year) - Room 301
2  22CE10002 Abhishek Burnwal (Pre-Final Year) - Room 201
3  22CE10003 Adrij Bhattacharya (Pre-Final Year) - Room 202
4  22CE10005 Aman Meena (Pre-Final Year) - Room 203
5  23CE10001 Aakarsh Raaj (Second Year) - Room 101
6  24CE10006 Ameya Neeraj Tambat (First Year) - Room No Room Available
7

```

