Programming Project

Team members

Malak Sharaf	23-101169 (Team Leader)
Sondos Aly	23-101189
Nour El Bakary	23-101084
Farah Ezzat	23-101224
Areej Taher	23-101030

Project Overview

The program receives input from the user about different courses and their dates and student IDs with their course codes then it sorts all the data into arrays which is used when the user asks about any function.

The functions include the number of students, halls or the student which attend in specific hall or date and many other function which will be described in detail.

CoursesExams=[101,3/5/2016,A;110,5/5/2016,B;103,5/5/2016,A;120,6/5/2016,D;132,7/5/2016,B]
Students=[191001,201,101,110;191002,201,110,103;191003,110,120,132;191006,103,132;191000,102,105]

2D courses

course	day	mon	year
101	3	5	2016
110	5	5	2016

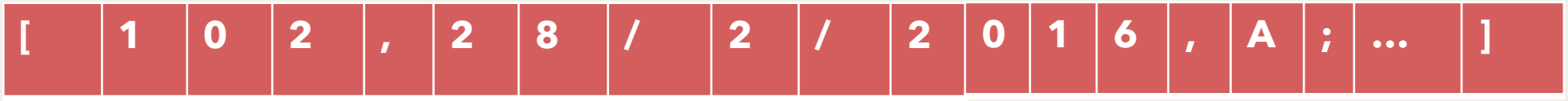
1D halls

halls
A
B

2D Students

Students	C_1	C_2	C_3	C_4	C_5
191001	201	101	110	0	0
191002	201	101	103	0	0

Input



- 3 for nested loops
- Count = count*10 + (array[n]-'0')
- if(array[n] >= 'A' && array[n] <= 'Z')
- Loop with always true which does not end until it scan "Quit" which return zero.

Function 1 (Number_Students)

Objective: Number of Students

Parameters: Students Array

- The function iterates until Array[i] == 0 is true
- Printf i
- Break;

Function 2 (Number_Halls)

Objective: Number of Halls

Parameters: Halls Array

- Copy the array
- Sort
- Count when there's a different character

*Function 3 (*Student_ID_Min*)*

The 3rd Function (Student_ID_Min) takes the student array (with the size [50] [6]) as a parameter. In the function I initialize an integer called min_id to the first element in the array.

Then using a for loop, I pass by every Student ID and if it is less than the assigned number to min_id, the integer is redefined by the new ID.

After the loop has passed all IDs, it then prints the minimum ID.

*Function 4 (*Students_Dropped_ID*)*

The 4th Function (Students_Dropped_ID) also takes the student array (with the size [51] [6]) as a parameter. And for this function we have 3 variables 2 integers (min_id and max_id) both assigned to the first element in the array and a Boolean (a).

The function first uses the same loop used in Function 3 with the difference of also assigning the greatest ID to max_id
Then I use a nested for loop where the outer loop adds 1 to the min_id as long as it's less than the max_id

The inner loop then passes by all the IDs in the array to check if one of them equals the new variable. If the variable is found, the Boolean (a) changes to True and the whole loop breaks

However, if the Id is not in the array and the (a) is still false the ID will be printed, and the a will be reassigned as false to ensure that the loop continues correctly

*Function 5 (*Exams_Period_InDays*)*

Firstly, 2 for loops including nested if to calculate max, min dates. Different cases are considered based on whether the start and end years are the same or different. **If years & months are the same** , days are counted as:

days_count = max_day- min_day + 1

If different months:

*days_count += daysinmonth (min_month, min_year) - min_day + 1

days_count += max_day*

To iterate months between max, min month we use

for (int n = 1; (min_month + n) < max_month; n++)

{days_count += daysinmonth((min_month + n), min_year);}

*Function 5 (*Exams_Period_InDays*)*

If statement used here when **min year < max year**, we will count days as:

days_count += daysinmonth(min_month, min_year) - min_day + 1

We will need to count months between diff years. So, in **1st year months will be iterated as:**

```
for (int n = 1 ; (min_month + n) <= 12 ; n++) {
        days_count += daysinmonth((min_month + n), min_year);}
```

For 2nd year months will be iterated as:

```
for (int x = 1 ; x < max_month ; x++) {
        days_count += daysinmonth(x, max_year);}
        days_count += max_day;
```

At the end **print** days_count

Function 6 (Course_Students)

- The function Course_Students requires the user to input a course code and the output will be the students who have taken this course.

- x is the course code that the user will input. int x = 0

-The program will **iterate** row by row and by nested for move column by column to check course code for each student to know if the student has taken this course by making **nested for loop**.

```
for ( int i = 0 ; Course_Students[i][0]!=0 ; i++){
    for ( int j = 0 ; j < 6 ; j++){
```

If the student **take this course then print this student ID**.if (Course_Students [i][j] == x){

```
//if these students take this course
    printf("%d\n", Course_Students[i][0]);
```

Function 7 (List_Course_Students_More)

This function iterates through each course in the Courses Exams array counts the number of students enrolled in each course by comparing course codes in the student array, and outputs the course codes with more than n students enrolled in.

```c
// Function 7
void List_Course_Students_More(int Student[51][6] , int CoursesExams[20][4]){
    //the outer loop iterate over each column in the course array (representing each course)
    int n =0;
    scanf("%d",&n);
    for(int x = 0 ; x < 20 ; x++){// Loop through each course in CoursesExams
         // Initialize count for the number of students in the current course
        int count = 0;
        // Loop through each student in Student array
        for(int i = 0 ; i < 51 ; i++){
                 // Loop through course codes
            for(int j = 1 ; j<6 ; j++){
                // Check if the current student is enrolled in the current course
                if(Student[i][j] == CoursesExams[x][0]){
                    count++;// Increment count if the student is enrolled in the course
                }
            }
        }
         // Check if the count of students in the current course is greater than n

        if(count > n && CoursesExams[x][0]!=0){
                 //printing course codes of courses having more than n students

            printf("%d\n" , CoursesExams[x][0]);
        }
    }
}
```

Function 8 (List_Student_Courses_Less)

Prints student IDs of the students attending more than n courses. It iterates over the row (representing the students) and checks the number of courses attended by each student. If the number of courses enrolled by the student is less than n it prints it out.

```
1  void function_8(int number_of_courses, int array[51][6]) {
2      // The outer loop Iterates through the row each row of the 2D array
           representing students
3      for(int i = 0; i < 51; i++) {
4          // Initialize a counter to keep track of the number of courses each
               student takes
5          int count = 0;
6          // Loop through each column starting from the second column (index 1)
               (representing courses attended by the student)
7          for(int j = 1; j < 6; j++) {
8              // Check if the value in the current cell is greater than 0 (meaning
                   the student is attending the course) if yes it increments it by 1
9              if (array[i][j] > 0) {
10                 // If it is, increment the count
11                 count++;
12             }}
13         // Check if the count of attended courses is greater than the specified
               number_of_courses
14         if (count < number_of_courses) {
15             // If it is, print the student ID (value in the first column, index 0
                   ) in a separate output line
16             printf("%d\n", array[i][0]);
17         }}}
```

Function 9 (List_Hall_Students)

Objective: List IDs of the students in a certain hall at a certain date

Sample input: (A, 28/4/2016)

Parameters: Courses Exams array , Students array, Halls array

- 2 scanf
- For loop that turns the string into integers using (Day = Day*10) and (Date[i] - ‘0’)
- Find the course code with that certain hall and date
- Nested for loop that iterates on students
- When course is found the student's ID is printed and the inner loop breaks

Function 10 (List_Hall_Students_InAnyday)

List_Hall_Students_InAnyday

Where we take hall name 'A' then we print all IDs without repeating

```
for(int i=0; i<20;i++){
      if(hallname[i] == hall){
        temparray[num]=courseCode[i][0];            num++;}}
```

3 nested for loops to check each course with all courses in the temparray if it found it print the ID then break the loop.

Function 11

The last function is quit, it ends the program. It takes no parameters and returns a variable (0) when called. Return Quit(); breaks the main function and ends the whole program.

Team Roles/ Responsibilities

Functionality	Responsible Team Member(s)
Function 1, Function 2, Function 9	Nour El Bakary
Function 3, Function 4	Areej Taher
Function 5, Function 6	Sondos Aly
Function 7, Function 8	Farah sultan
Function 10	Malak Samy
Function 11	Nour El Bakary
INPUT	Malak Samy
Arranged Project Report	Sondos Aly
Project report details	All of us
Presentation Slides	Areej Taher, Sondos Ahmed, Nour El Bakary

Thank you

Questions?