**gcc singlePlayer.c tigr.c -o snake -lm -lGL -lX11 -lpthreadW8 -** PRACTICE

*Structures & Data Modeling*

## *At the end of this practice, you should be able to…*

* Use **struct** to model real-world entities.
* Write and reuse **functions** with **structured** **data**.
* Combine **arrays** and **nested structures**.
* **Design data models** from **problem** descriptions.

## *How do we structure exercises?*

We organize this practice into 4 parts:

| ANALYSE | **Understand** existing codes, find the **bugs** or **complete** missing gaps |
| --- | --- |
| MANIPULATE | Ensure you can **apply the theory** with some basic challenges |
| CREATE | **Express your creativity** with more complex challenges |

## *Are you lost?*

You can read the following documentation to be ready for this practice

Structs

<https://www.w3schools.com/c/c_structs.php>

**MANIPULATE**

**EX 1 (*The Students*)**



* Define a structure called **Student**:
  + The student’s name (array of chars, size 50).
  + The student id (integer).
  + The student scores (array of floats, size 5)
* Declare 2 variables of type Student: viseth and sousdey
* Initialize both students’ data using structure initialization
* Calculate both students’ average marks

Bonus: define a function to calculate the student’s average marks

double getScoresAverage (Student student);

* Print information

**#include <stdio.h>**

**#include <string.h>**

**typedef struct {**

**char name[50];**

**int id;**

**double score[5];**

**}Student;**

**double avgScores(Student name){**

**int sum = 0 ;**

**for(int i = 0; i < 5; i++){**

**sum += name.score[i];**

**}**

**return sum/5 ;**

**}**

**int main (){**

**Student viseth;**

**strcpy(viseth.name,"Vorng Viseth");**

**viseth.id = 12345678;**

**for(int i = 0; i < 5; i++){**

**viseth.score[i] = 10;**

**}**

**printf("name : %s\nid : %d\naverage scores : %lf\n", viseth.name, viseth.id, avgScores(viseth));**

**printf(".....................\n");**

**Student sousdey;**

**strcpy(sousdey.name,"sousdey");**

**sousdey.id = 9898989;**

**for(int i = 0; i < 5; i++){**

**sousdey.score[i] = 15;**

**}**

**printf("name : %s\nid : %d\naverage scores : %lf\n", sousdey.name, sousdey.id, avgScores(sousdey));**

**return 0;**

**}**

**EX 2 (*The Time*)**



* Define a structure named **Time** with members hours, minutes, and seconds.
* Create a function sumTime, which returns the sum of 2 times

Time sumTime(Time time1, Time time2);

Example:

sumTime( {1,45,00}, {1,30,00} ); // {3, 15,00}

*Explanation: 1h45 + 1h30 = 3h15x*

* Create a function compareTime, which compares 2 times :

*The function returns (1 if time1 is greater, -1 is time2 is greater, 0 if equal)*

compareTime( {1,45,00}, {1,30,00}); // 1

*Explanation: 1h45 > 1h30*

* Create 2 variables of type Time
* Print the 2 times, the sum of the 2 times and which time is greater

**#include <stdio.h>**

**typedef struct{**

**int hours;**

**int min;**

**}Time;**

**Time sum(Time time1, Time time2){**

**int minSum = time1.min + time2.min;**

**int hourSum = time1.hours + time2.hours;**

**if(minSum >= 60) {**

**minSum -= 60;**

**hourSum++ ;**

**}**

**Time sum;**

**sum.hours = hourSum;**

**sum.min = minSum;**

**return sum;**

**}**

**int compareTime(Time time1, Time time2){**

**if (time1.hours == time2.hours && time1.min == time2.min) return 0;**

**else if (time1.hours > time2.hours) return 1;**

**else if (time1.hours == time2.hours && time1.min > time2.min) return 1;**

**else return -1;**

**}**

**int main(){**

**Time time1;**

**time1.hours = 1;**

**time1.min = 45;**

**Time time2;**

**time2.hours = 1;**

**time2.min = 45;**

**Time sumTime = sum(time1,time2);**

**printf("time 1 : %dh %dmin\n", time1.hours, time1.min);**

**printf("time 2 : %dh %dmin\n", time2.hours, time2.min);**

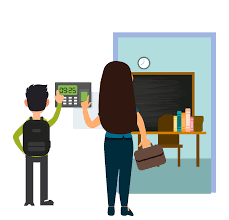
**printf("%dh %dmin\n", sumTime.hours, sumTime.min);**

**printf("%d\n", compareTime(time1, time2));**

**return 0;**

**}**

**EX 3 (*The Attendance*)**



* Define a structure called **Attendance** composed of 2 members:
  + A student name (array of chars, size 50).
  + A time (use the Time structure - previous exercise)
* Create a **list of attendances** (array of Attendance, size )
* Create a function **getLastest**, which return which student is the latest.

*The function returns the attendance with the greatest time*

Attendance getLatest(Attendance attendances[5]);

Tips: you can reuse the function compareTime from the previous exercise.

* Print the name of the student who came the latest in the class

*Example : The latest student is: NamKea*

#include <stdio.h>

typedef struct{

int hours;

int min;

}Time;

typedef struct {

char studentName[50];

Time time;

} Attendance ;

int compareTime(Time time1, Time time2){

if (time1.hours > time2.hours) return 1;

else if (time1.hours == time2.hours && time1.min > time2.min) return 1;

else return 0;

}

Attendance getLatest(Attendance attendance[5]){

Attendance latest = attendance[0];

for(int i = 1; i < 5; i++){

if (compareTime( attendance[i].time, latest.time)) latest = attendance[i];

}

return latest;

}

int main(){

Attendance attendance[5];

for(int i = 0; i < 5; i++){

printf("Enter student's %d name : ", i + 1);

scanf(" %s",attendance[i].studentName);

printf("Enter student's %d time (hour min): ", i + 1);

scanf(" %d %d", &attendance[i].time.hours, &attendance[i].time.min);

}

Attendance latest = getLatest(attendance);

printf("the latest student is :\n");

printf("%s\n%dh %dmin\n", latest.studentName, latest.time.hours, latest.time.min);

return 0;

}

**ANALYSE**

*For the below 4 exercises:*

* **Describe** your data model

*All needed data : integer, char, double, arrays, strings, structs, nested structs to solve this problem*

* **Illustrate** with an example, to explain your solution

*You can perform each exercise in groups (3 students) :*

* *Write down your solution on paper, paper board*
* *Come to present your approach to the whole class*

**PROBLEM 1** **(Manage a hotel)**

Manage a hotel with multiple rooms. Each room has a number, type (single/double), and status (available/booked). Each booking has guest info and check-in/check-out dates.

**Q1 –** **Describe** your data model:

**Q2 –** **Illustrate** with an example, to explain your solution:

**PROBLEM 2** **(Manage a sport league)**

A sports league has multiple teams. Each team has a name, coach, and list of players. Each player has a number, name, and position.

**Q1 –** **Describe** your data model:

**Q2 –** **Illustrate** with an example, to explain your solution:

**PROBLEM 3** **(Game 2D)**

Model a game map as a 2D grid. Each tile can be empty or contain an object like a player, enemy, wall, or item. Each object has different properties.

**Q1 –** **Describe** your data model:

**Q2 –** **Illustrate** with an example, to explain your solution:

**PROBLEM 4** **(Memory Game)**

****

Manage a memory game where players flip cards to find matching pairs.

* The board is a grid of face-down cards
* Each card has a symbol and a flipped state (face up or down)
* Players take turns flipping two cards

**Q1 –** **Describe** your data model:

**Q2 –** **Illustrate** with an example, to explain your solution:

**CREATE**

**CHOOSE A PROBLEM, IMPLEMENT IT…**

1. **Choose one** of the 4 data modeling problems provided

(*hotel, sport league, 2d game, memory game)*

1. **Implement the functionality** to play the game or to manage the system.

*Your program should include well defined functions to separate the code into smaller tasks*

1. Finally **implement the** **main function** that tests the use of your application.