**W8 -** PRACTICE

*Strings + Structures*

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

STRING

* Input **strings** and **list of strings**
* Manipulate **string with functions**
* Implement functions like strcmp, strcat, atoi, strlen, tolower, toupper
* Applying the **Top to Bottom approach** to string-based problems

STRUCTURES

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

**STRING - CREATE**

**EX 1 - *Stylish Initials Generator*)**

Create a function that takes a full name (e.g., "John Ronald Reuel Tolkien") and outputs the initials in the uppercase with dots (e.g., "J.R.R.T.").

| FUNCTION NAME | printInitials | |
| --- | --- | --- |
| FUNCTION DESCRIPTION | Print string initials in uppercase with dots (e.g., "J.R.R.T."). | |
| PARAMETERS | const char\* | The string |
| RETURN | Void |  |

**Example of usage**

| INPUT | TO PRINT |
| --- | --- |
| "Aaa Bbb Ccc" | A.B.C. |
| "Ronan Ogor" | R.O. |
| "a b c" | A.B.C. |
| "John Ronald Reuel Tolkien" | J.R.R.T. |

**#include <stdio.h>**

**#include <string.h>**

**#include <ctype.h>**

**void printInitail(const char\* string) {**

**printf("%c.", toupper(string[0]));**

**for(int i = 1; string[i] != '\0'; i++){**

**if(string[i] == ' ' && string[i + 1] != '\0' && string[i + 1] != ' ') printf("%c.", toupper(string[i+1]));**

**}**

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

**}**

**int main(){**

**char string[50];**

**printf("Enter your name : ");**

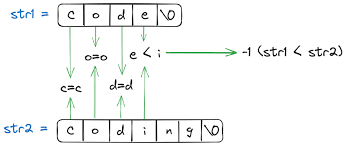
**fgets(string, sizeof(string), stdin);**

**printInitail(string);**

**return 0;**

**}**

**EX 2 - *Implement Your Own strcmp() HARD ! BONUS***



*Implement a function that compares 2 strings, characters by characters*

Write a C program that implements a custom version of the strcmp() function. Your implementation should **follow the original behavior and specification** of the standard library function:

int strcmp(const char \*s1, const char \*s2);

**It should:**

* Return **0** if both strings are equal.
* Return a **negative value** if s1 is lexicographically less than s2.
* Return a **positive value** if s1 is lexicographically greater than s2.

**Requirements**

1. **Break down** your solution into at least two functions
2. **Handle Edge Cases**
   * One or both strings are empty ("").
   * Strings have different lengths.
   * Strings are identical up to a point, but one continues further
3. **Avoid Using <string.h>**
   * Do **not** use built-in functions like strcmp(), strlen(), etc.
   * All logic must be implemented manually

**Examples**

my\_strcmp("apple", "apple"); // returns 0

my\_strcmp("apple", "apricot"); // returns a negative value

my\_strcmp("banana", "apple"); // returns a positive value

my\_strcmp("", ""); // returns 0

my\_strcmp("a", ""); // returns a positive value

my\_strcmp("", "a"); // returns a negative value

#include <stdio.h>

int stringLen(const char\* string) {

int i = 0;

for(i = 0; string[i] != '\0'; i++);

return i;

}

int my\_stringcmp(const char\* string1, const char\* string2) {

if(stringLen(string1) < stringLen(string2)) return -1;

else if(stringLen(string1) > stringLen(string2)) return 1;

else {

for(int i = 0; string1[i] != '\0'; i++) {

if(string1[i] > string2[i]) return 1;

else if(string1[i] < string2[i]) return -1;

}

}

return 0;

}

int main() {

char string1[30];

char string2[30];

printf("Enter string 1 : ");

fgets(string1, sizeof(string1), stdin);

printf("Enter string 2 : ");

fgets(string2, sizeof(string2), stdin);

printf("%d\n", my\_stringcmp(string1, string2));

return 0;

}

**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 = 3h15*

* 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);**

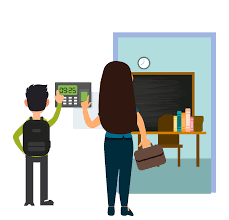
**printf("%dh %dmin\n", sumTime.hours, sumTime.min);**

**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**

**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:

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

//date check in/out

typedef struct {

// input example DD/MM/YY

char checkIn[50];

char checkOut[50];

} Date;

typedef struct {

char guestName[50];

int people; // how many people are staying in the room

int phoneNumber;

int days; // booked for how many day

} GuestInfo;

//guest info and checkin/out date

typedef struct{

GuestInfo info;

Date date;

} Booking;

typedef struct {

int roomNum;

char type[10]; // single or double

char status[20]; // available or booked

Booking booking;

} Room;

Room room[60]; // array of room struct

int choice = 0; // makes choice on the hotel menu

int in = 0; // room number for check in

int out = 0; //rOom number for checking out

int check = 0; // room number for checking info

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

**#include <stdio.h>**

**#include <string.h>**

**//date check in/out**

**typedef struct {**

**// input example DD/MM/YY**

**char checkIn[50];**

**char checkOut[50];**

**} Date;**

**typedef struct {**

**char guestName[50];**

**int people; // how many people are staying in the room**

**int phoneNumber;**

**int days; // booked for how many day**

**} GuestInfo;**

**//guest info and checkin/out date**

**typedef struct{**

**GuestInfo info;**

**Date date;**

**} Booking;**

**typedef struct {**

**int roomNum;**

**char type[10]; // single or double**

**char status[20]; // available or booked**

**Booking booking;**

**} Room;**

**void checkIn(Room\* room){ //info you need to fill for check in**

**printf("Enter guest name : ");**

**scanf(" %s", room->booking.info.guestName);**

**printf("Enter number of people :");**

**scanf(" %d", &room->booking.info.people);**

**printf("Enter your phone number : ");**

**scanf(" %d", &room->booking.info.phoneNumber);**

**printf("Enter how many days you want to book : ");**

**scanf(" %d", &room->booking.info.days);**

**strcpy(room->status, "booked");**

**printf("Enter the check in dates : ");**

**scanf(" %s", room->booking.date.checkIn);**

**printf("Enter the check out dates : ");**

**scanf(" %s", room->booking.date.checkOut);**

**printf("Room booked successfully!\n");**

**}**

**void checkOut(Room\* room){ // clear the guest info after checking out**

**strcpy(room->booking.info.guestName, "\0");**

**room->booking.info.people = 0;**

**room->booking.info.phoneNumber = 0;**

**room->booking.info.days = 0;**

**strcpy(room->booking.date.checkIn, "\0");**

**strcpy(room->booking.date.checkOut, "\0");**

**strcpy(room->status, "available");**

**}**

**void printRoomInfo(Room room){**

**printf("-----Room Status-------\n");**

**printf("Room number : %d\n", room.roomNum);**

**printf("Room type : %s\n", room.type);**

**printf("Status : %s\n", room.status);**

**printf("guest name : %s\n", room.booking.info.guestName);**

**printf("guest number : %d\n", room.booking.info.phoneNumber);**

**printf("poeple amount : %d\n", room.booking.info.people);**

**printf("days stayed : %d\n", room.booking.info.days);**

**printf("Check in date : %s\n", room.booking.date.checkIn);**

**printf("Check out date : %s\n", room.booking.date.checkOut);**

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

**}**

**int main() {**

**Room room[60]; // array of room struct**

**for(int i = 0; i < 60; i++) {**

**room[i].roomNum = i + 1; // set room room number o**

**strcpy(room[i].status, "available"); // set all the room to available**

**if(i < 30 ) strcpy(room[i].type, "Single"); // set room number from 1-30 to single**

**else strcpy(room[i].type, "Double"); // set room number from 31-60 to double**

**}**

**int choice = 0;**

**while(choice != 4){**

**// Room menu**

**printf("---- HOTEL MENU ----\n");**

**printf("1. Check In\n");**

**printf("2. Check Out\n");**

**printf("3. Check Room Info\n");**

**printf("4. Exit\n");**

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

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

**if(choice < 0 || choice > 5) {**

**printf("Invaild input !\n");**

**continue;**

**}**

**if(choice == 1) {**

**int in = 0;**

**printf("What room do you what to book?\nSingle Bed (1-30)\nDouble Bed (31-60)\n");**

**printf("Enter room number (XX) : ");**

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

**if(in < 0 || in > 60) {**

**printf("Invaild input !\n");**

**continue;**

**}**

**if(strcmp(room[in - 1].status, "booked") != 0) {**

**printf("Room is available!\n");**

**checkIn(&room[in - 1]);**

**} else printf("Room is already booked!\n");**

**}else if(choice == 2) {**

**int out = 0;**

**printf("Enter the check out room : ");**

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

**if(out < 0 || out > 60) {**

**printf("Invaild Input!\n");**

**continue;**

**}**

**checkOut(&room[out - 1]);**

**}else if(choice == 3) {**

**int check = 0;**

**printf("Enter the room number : ");**

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

**if(check < 0 || check > 60) {**

**printf("Invaild input !\n");**

**continue;**

**}**

**printRoomInfo(room[check - 1]);**

**}**

**}**

**return 0;**

**}**