Домашно 2 - софтуер

# Задача 1

Дефинирайте структура hero\_t, която описва един герой с максимална и текуща кръв (hp), щета и защита (числа с плаваща запетая).

Дефинирайте функцията float hero\_power(struct hero\_t) която пресмята силата на един герой като сума от щетата, защитата и максималната му кръв (без текущата).

Дефинирайте функцията int can\_attend\_instance(int min\_power\_required, struct hero\_t \*heroes, int heroes\_count), която проверява дали всеки един герой има поне 50% от максималната си кръв и има поне min\_power\_required сила за да участва в дадената битка.

# Задача 2

Дефинирайте структура chair\_t, която описва един стол с цена (в лева), максимално тегло на седящия (в килограми) и дали е с кожена тапицерия.

Дефинирайте функцията void apply\_discount(struct char\_t \*chairs, int chairs\_count), която намалява цената на кожените столове с 18% от първоначалната им и с 20% на некожените столове.

Дефинирайте функцията struct chair\_t pick\_chair(int person\_weight\_kg, struct char\_t \*chairs, int chairs\_count), която връща най-евтиния кожен стол с максимално тегло на седящия поне person\_weight\_kg или най-евтиния некожен стол отново с максимално тегло поне person\_weight\_kg ако нито един кожен, отговарящ на изиксванията. Приемете, че има поне 1 стол с максимално тегло на седящия поне person\_weight\_kg килограма.

# Задача 3

Дефинирайте структура person\_t, която дефинира човек с 2 булеви променливи - дали знае за <https://rationalwiki.org/wiki/Roko%27s_basilisk> и дали е помагал за създаването на свръхинтелигентен изкуствен интелект.

Дефинирайте функцията int ai\_should\_punish(struct person\_t), която казва дали свръхинтелигентния изкуствен интелект трябва да накаже този човек (само ако знае, но не е помагал).

Дефинирайте функцията int should\_punish\_all(struct person\_t \*people, int people\_count) която казва дали свърхинтелигентния изкуствен интелект ще накаже цялата група от хора. Това трябва да се случи ако мнозинството от хората трябва да бъдат наказали (половината + 1 или повече).

# Задача 4

Дефинирайте функцията int rightmost\_one(int n), която връща побитова маска с 1-ца само там, където се намира най-дясната единица в двоичния запис на числото n.

Пример:

rightmost\_one(5) -> 1 (0001)

rightmost\_one(10) -> 2 (0010)

rightmost\_one(20) -> 4 (0100)

rightmost\_one(0) -> 0 (0000)

# Задача 5

Дефинирайте структура bike\_tire\_t, която описва гума на колело с диаметър (в инчове), широчина (от 0.9 до 4.8, в инчове) и минимално и максимално налягане (в барове).

Дефинирайте структурата tire\_stand\_t, която има указател към динамично заделен масив от bike\_tire\_t гуми и големината на масива.

Дефинирайте функцията struct tire\_stand\_t read\_tires(), която прочита от стандартния вход гумите в една стойка, като всяко описание на гума е на нов ред. Четенето на гуми спира когато се срещне празен ред (сами измислете формат на записване на параметрите на една гума и го изпишете на стандартния изход в началото на функцията). Не се знае предварително бройката на гумите, които ще се прочетат.

Дефинирайте функцията int is\_for\_mountainbike(struct bike\_tire\_t) която казва дали дадена гума е подходяща за каране в планина. Това е така когато гумата:

1. има диаметър 26, 27.5 или 29 инча
2. широчината е поне 1.7 инча
3. минималното налягане е под 2 бара

Дефинирайте функцията void print\_mountainbike\_tires(struct tire\_stand\_t), която принтира на стандартния изход гумите, подходящи за каране в планина.

# Задача 6

Направете собствена имплеменатция на вектор - масив, който сам се уголемява когато свърши . Вашият вектор трябва да работи с елементи от типа bike\_tire\_t от задача 5.

Имплементирайте операциите:

1. push\_back - добавя елемент в края на вектора
2. size - връща колко елемента има във вектора
3. get - по даден индекс връща съответната стойност на този индекс

Можете да добавите и още операции, ако са необходими.

Дефинирайте вектора в в отделен файл (vector.c и vector.h)

Променете tire\_stand\_t от задача 5 да използва вектор, за да държи в себе си гуми (може да изкопирате кодът си от задача 5 и да го промените).

Напишете собствен Makefile, който да билдва вектора (vector.o) и частта с гумите (tires.o) и ги линква (за да се получи изпълним файл).

Имплементирайте struct tire\_vector\_t filter\_non\_mountainbikes(struct tire\_stand\_t), която връща вектор с всички гуми, които не стават за каране в планина.