КОНТРОЛНО ЗА ОФОРМЯНЕ НА УСПЕХ
ПО ПРОГРАМИРАНЕ 10 КЛАС

Времето за работа е 80 минути. Всяка задача е за оценка.

За 3

Дефинирайте структура browser_t, която съдържа в себе си име, версия и пълно име(име + версия). Дефинирайте функцията void release(struct browser_t*), която увеличава версията на браузъра с 1, след това презаписва пълното му име на база новата версия.

Пример: Ако браузърът се казва Internet и има версия 6, то при релийс версията става 7 и пълното име става Internet 7

Да се дефинира функцията void upgrade_oldest(struct brower_t *arr, int arr_size) която намира най-стария браузър и релийсва нова версия от него.

За 4

Имате масив от числа. Числата ще бъдат между 0 и 65535. Представяйки числата в двуичен вид бихме получили масив, който изглежда по подобен начин
[
0000000000000000, // 0
0000000000000100, // 4
0000000000001100, // 20
0101110000001110 // 23566
]
Имаплементирайте функцията int *bomb(int *arr, int arr_size, int center_row, int center_col, int bomb_size) която бомбардира определен ред и колона в масива с числата представени в двуичен вид. Бомбардировката се изразява в това да се сложат всички битове, които попадат в уцелената част на 0. Пример:
При така подаден масив, ако бомбардираме в ред 3, колона 13 и размера на бомбата е 3 червената единица е центъра на удара, а всички удебелени битове са уцелената част и всички трябва да се направят на 0.
[
0000000000000000, // 0
0000000000**001**00, // 4
0000000000**01****1**00, // 20
0101110000**011**10 // 23566
]

След удара функцията трябва да върне променения масив. От примера той би изглеждал така
[
0000000000000000, // 0
0000000000**000**00, // 0
0000000000**0**0**0**00, // 0
0101110000**000**10 // 23554
]

Забележка: размерът на бомбата винаги ще бъде нечетно число;

За 5

Дефинирайте структурата site_t, която съдържа заглавие и url на сайта. Дефинирайте и структура browser_t, която съдържа история (масив от сайтове) и размер на историята, както и начин да знае, кой е текущия сайт, който гледаме. Имплементирайте

- struct browser_t open(), която връща браузър с инициализирана история
- void close(struct browser_t*), която затваря браузъра и зачиства всички данни
- void visit_site(struct browser_t*, struct site_t), която добавя текущия сайт в историята и го слага като текущ за браузъра.
- void back(struct browser_t*), която се връща на предишния сайт, ако няма предишен нищо не се променя
- void forward(struct browser_t*), която отива на следващия сайт от историята, ако има такъв

Пример:

- Отварям браузър
- Посещава facebook.com
- Посещавам twitter.com
- Back - връща ме на facebook.com
- Forward - връща ме на twitter.com

За 6

Използвайки структурите за сайт и браузър от задачата за 5 имплементирайте същите функции с няколко промени:

- Ако сме се върнали няколко пъти с функцията back и след това извикаме visit_site историята да бъде зачистена и да се добави само последния сайт
    - Пример:
    - Посещаваме facebook.com
    - Посещаваме twitter.com
    - Посещаваме imgur.com

    - Даваме back (отиваме на twitter.com и имаме една стъпка напред - imgur)
    - Посещаваме tumblr.com (вече нямаме стъпки напред и няма как да стигнем до imgur; имаме 2 стъпки назад - twitter и facebook)
- Вместо масив от сайтове за историята използвайте подходяща структура от данни като си дефинирате само нужните методи за нея
- Променете пазенето на текущия отворен сайт спрямо структурата ако е нужно