It depends on how you call it. The purpose of srand() is to seed the pseudo-random number generator used by rand(). So when you call srand(i), it will initialise rand() to a fixed sequence which depends on i. So when you re-seed with the same seed, you start getting the same sequence.

The most common use case is to seed the generator just once, and with a suitable "random" value (such as the idiomatic time(NULL)). This ~~guarantees~~ makes it likely that you'll get different sequences of pseudo-random numbers in different program executions.

However, occasionally you might want to make the pseudo-random sequence "replayable." Imagine you're testing several sorting algorithms on random data. To get fair comparisons, you should test each algorithm on the exact same data - so you'll re-seed the generator with the same seed before each run.

In other words: if you want the numbers simply pseudo-random, seed once, and with a value as random as possible. If you want some control & replayability, seed as necessary.

**int const limit = 101; -**nes uzduotyje sako kad generuojami skaiciai turi buti iki 100, tad didziausias skaicius turi buti 99, o viena vieta palikta nuliui.

**mas [i\*K+j] –** uzraso dvimati masyva vienmaciu format

**rand()%limit –** generuoja atsitiktinius skaicius iki limito (-db yra 100)

**srand (time(NULL));-** naudoju sita, nes buvau susidurusi su problema, kad sugeneruoja masyvuose tas pacias reiksmes. Tad parasius sita issprende problema. Tai sitas **srand(time(NULL)) –** inicializuoja srand.

**NULL - null** pointer constant.

(<https://stackoverflow.com/questions/21726414/calling-srand-twice-in-the-same-program> )

**int \*mas, \*mas1; - sukuriu rodykle, nes nezinau kokio dydzio bus masyvas**

**///mas = (int\*)calloc(K\*K, sizeof(int));**

* **calloc –** priskiria atminyje tam tikra vieta ir tas reiksmes prilygina nuliui. calloc sintakse - calloc(size\_t nitems, size\_t size)

nitems – skaicius elementu (**siuo atveju bus K\*K) elemetu didis**

size – dydis element (**siuo atveju bus int) element kiekis**

**(**[**https://www.tutorialspoint.com/c\_standard\_library/c\_function\_calloc.htm**](https://www.tutorialspoint.com/c_standard_library/c_function_calloc.htm) **)**

|  |  |  |  |
| --- | --- | --- | --- |
| **into dydis -** int | 2 or 4 bytes |  | -32,768 to 32,767 or -2,147,483,648 to 2,147,483,647 |

**ar reiktu prideti?(pries return 0)**

**free (mas);**

**free (mas1);**