Algoritmi probabilisti (randomised algorithms):

* algoritmii genetici
* algoritmi Las Vegas:
  + algoritmi cu un factor probabilistic care mereu furnizeaza rezultatul exact iar complexitatea depinde de factorul probabilistic.
  + Ex: quickSort cu pivot ales aleator: complexitate medie O(n logn) - de multe ori superioara altor algoritmi O(n logn).
* algoritmi Monte Carlo:
  + algoritmi cu un factor de aproximare care furnizeaza o solutie apropiata de cea dorita. Altfel spus, converge catre solutia dorita:
  + Ex: Algoritm pentru aproximarea lui pi.

Pasii algoritmului:

fie un patrat cu coltul stanga-jos in (0,0) si latura 1.

fie un cerc inscris in acel patrat:

* centrul cercului: (½,½)
* raza cercului: 1/2

Generam aleator, uniorm distribuit, *n* puncte in patrat. Fie *p* din acele *n* puncte cele care se afla si in cerc. Pentru un *n* suficient de mare p/n va tinde la raportul dintre suprafata cercului si cea a patratului, adica la pi/4.

programul vostru:  
 sa definiti o structura de tip punct

sa scrieti o functie care genereza aleator puncte in patratul descris mai sus

sa scrieti o functie care calculeaza distanta dintre 2 puncte

folositi aceasta functie pentru a verifica daca un punct este in cerc sau nu.

generati pe rand un numar mare de puncte (ex 1.000.000)

numarati cate puncte au fost generate in cerc

raportul dintre numarul de puncte din cerc si numarul total de puncte, inmultit cu 4, va tinde catre *pi*.