Algorithm for initial configuration of chains:

Inputs:

lattice dimensions of (lx x ly x lz)=(7 x 7 x 20) consisting of i= 36 polymer chains, each chain of j=20 polymer beads.

Process (main()):

- Create a file chains.csv with header x,y,z.
- Choose any of the four defined subroutines (diagonal(), xarrange(), yarrange(), randomize())
- for each chain (i = 0 to 35),
    - for each bead (j = 0 to 19)
        - prints the bead coordinates
- Write bead coordinates to chains.csv and close the file
- return 0

Output:

- file chains.csv with 36 chains and 20 beads.

Subroutines:

diagonal():

- Initializes chain index (i = 0) and starting bead position (lsx, lsy,) = (0,0)
- for each chain index (i = 0 to nc-1),
    - for each bead index (j = 0 to bc-1)
        - the bead coordinates are set as (lsx, lsy, j)
- Modular operator (%) is used to compute the bead positions (lsx,lsy) and also sets the positions within range.
- chain arrangement goes like main diagonal, diagonal-1, diagonal+1, diagonal-2, diagonal+2, diagonal-3, diagonal+3.

xarrange():

- Initializes chain index(i=0) and starting bead coordinates (lsx, lsy) = (0,0).
- While (i<nc)
    - For each bead index (j = 0 to bc-1)

        - the bead coordinates are set as (lsx, lsy, j)
- For each chain index (i), lsy is held constant, lsx = (lsx +1)%lx and once lsx reaches the range end, then lsx is resets back and lsy = (lsy + 1)%ly.

yarrange():

- Initializes chain index(i=0) and starting bead coordinates (lsx, lsy) = (0,0).
- Loop While (i<nc)
    - For each bead index (j = 0 to bc-1)
        - the bead coordinates are set as (lsx, lsy, j)
- For each chain, lsx is held constant while incrementing the lsy (lsy = lsy + 1) and once lsy reaches the range end, then lsy is resets back and lsx = lsx + 1.

randomize():

- Initialize chain index (i=0), arrays valuex[nc], valuey[nc], srand(0)
- Loop while (i<nc)
    - valuex[i] = rand()%lx, valuey[i] = rand()%ly.
    - For each k =0 to k < i,
    - If (valuex[i]== valuex[k] && valuey[i]==valuey[k]): Repeated site
        - i=i-1
    - i=i+1
- While (i = 0 to i <nc)
    - For each bead index (j=0 to bc-1),
        - set its coordinates (valuex[i], valuey[i], j)