Project information sheet for IC 150 P Computation for engineers Lab

Odd semester: August to November, 2016

Objectives

- To work as a team on a bigger, and more challenging software project than in the regular lab sessions.
- To discuss, and work together as a team.
- To teach to others, and learn from others in the team.
- To collectively design the overall architecture of the software, and to implement it.
- To get a grasp of the overall project, so that any member of the team can handle any algorithm challenge, and any debugging challenge on the project.

Time table for project activities

Week	Activities, Deadlines
3 PM, Nov 3rd	Project topics released. Groups announced.
3 PM, Nov 7th	**DEADLINE:** for making choice on choice link
3 PM, Nov 14th	**DEADLINE:** for project design document
Nov 15th to 18th	Group project work, and, evaluation of design document
Nov 22nd to 25th	Group project work, and, evaluation of code
Nov 29th to Dec 2nd	Group project final evaluation

Grading of the project activities:

Each student will receive up to 20 marks from evaluations of the projects. The break up of the 20 marks is: **5 marks** (common to all members of the group, and for the design document) + **7 marks** (common to all members of the group, and for how the project as a whole was executed) + **8 marks** (specific to each individual, and for individual contributions to the project).

Further break-up of the 5 marks for the design document

Component	Marks
Complete specification of component functions with proper listing of each function and the programmer responsible for it	2 marks
A flow-chart, or appropriate pseudocode	2 marks
For overall clarity and excellence	1 mark
Total	5 marks

If you chose to write pseudo-code, plan it i such a way that you do not need more than 20 lines to write it.

For tips on how to write some pseudo code, consult:
http://www.wikihow.com/Write-Pseudocode

If you chose to present a flow chart, plan it i such a way that you do not need more than 20 boxes to present it.

For tips on how to design a flow chart, consult:
https://www.lucidchart.com/pages/how-to-make-a-flowchart

Further break-up of the 7 marks for the execution of the project

Component	Marks
Program meeting specifications, and more	2 marks
Any randomly picked member of the team is able to handle debugging challenge	2 marks
Any randomly picked member of the team is able to handle algorithm challenge	2 marks
For well-written `readme.txt` file	1 mark
Total	7 marks

Further break-up of the 8 marks for individual contributions to the project

Component	Marks
Algorithm challenge	2 marks
Justification for function specification	1 mark
Debugging challenge 1	2 marks
Debugging challenge 2	2 marks
Readable code	1 mark
Total	8 marks

1 Project topics

1.1 To develop a billing program for IIT Mandi canteen

(for clarifications, contact Kaustav Sarkar) There should be a separate file of items containing serial number, item name and price Inputs should be read from keyboard The output should be in text file format You are free to add as many features as you wish like keeping a track of most popular items etc.

1.2 To develop a dictionary of IIT Mandi birds and animals

(for clarifications, contact Kaustav Sarkar) There should be a separate file of items containing serial number, item name and brief description Inputs should be read from keyboard The output should be produced on screen You are free to add as many features as you wish to make the program interactive and interesting

1.3 To develop a program for the calculation of cricket match statistics

(for clarifications, contact Kaustav Sarkar) There can be a separate file of inputs containing vital information Real time inputs can be provided through keyboard. The output should be produced on screen/file. Consider plotting the run-rate curve. You are free to add as many features as you wish to make the program interesting like predicting possible totals before and during the match

1.4 To develop a program for the maintaining attendance records for a course

(for clarifications, contact Kaustav Sarkar) There should be a separate file of items containing roll number, student name and attendance The program should calculate percentage attendance for every student and check it against lower threshold set by instructor The output should be produced on screen and in a file You are free to add as many features as you wish to make the program useful. Can the program be made to send intimation to students with poor attendance?

1.5 To develop a program to locate the nearest city

(for clarifications, contact Kaustav Sarkar)There should be a separate file of major cities including their names and location in terms of latitude and longitude The program should take a location input from keyboard, find the nearest city(ies) and produce the output on screen You are free to add as many features as you wish to make the program useful, like climatic conditions, local language, important contacts, state of law and order or any other relevant warning etc.

1.6 Numerical differentiation

(for clarifications, contact Siddhartha Sarma) In this exercise, we will use the following formula to calculate derivatives of some of the well known functions.

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{y(x + \Delta x) - y(x)}{\Delta x}. \tag{1}$$

For example, consider $\ln(x)$ and $\sin(x)$. We know that their first order derivatives are $\frac{1}{x}$ and $\cos(x)$, respectively. Calculate derivatives of $\ln(x)$ and $\sin(x)$ using the Equation (1) and compare with the one we got directly, i.e., $\frac{1}{x}$ and $\cos(x)$. You should choose a very small value for Δx, e.g., 0.001.

1.7 Numerical integration

(for clarifications, contact Siddhartha Sarma) In similar manner, numerical integration can be done by using trapezoidal rule, which can be stated as:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n} \left(\frac{f(a)}{2} + \sum_{k=1}^{n-1} f\left(a + k\frac{b-a}{n}\right) + \frac{f(b)}{2} \right) \tag{2}$$

Using Equation (2), integrate $\frac{1}{x}$ and $\cos(x)$. Compare them with the values obtained from $\ln(x)$ and $\sin(x)$. You may consider $a = 1$, $b = 1.1$ and $n = 100$.

More about numerical integration can be found in the following wikipage: https://en.wikipedia.org/wiki/Numerical_integration

1.8 Random number generation and histogram

(for clarifications, contact Siddhartha Sarma) Generate 1000 random numbers having values between -5 and 5. Create a histogram of those numbers. You may use the function `rand()` for this purpose. Read about histogram in wikipage: https://en.wikipedia.org/wiki/Histogram. Reference for random number generation in C: http://stackoverflow.com/questions/822323/how-to-generate-a-random-number-in-c

1.9 Simulation

(for clarifications, contact Siddhartha Sarma) Simulate coin toss experiment of a biased coin. For a biased coin, probability of 'HEAD' or 'TAIL' is not equal to $\frac{1}{2}$.

If you have finished this task successfully, simulate the *matching pennies* game for different probability values of both the coins. More about the matching pennies game can be found here: https://en.wikipedia.org/wiki/Matching_pennies.

1.10 Finding rank of a matrix

(for clarifications, contact Siddhartha Sarma)

1. checking whether a vector is scaled version of another vector or not.
2. check whether a vector is a linear combination of a set of vectors or not.

1.11 Solve a set of linear equations using Gaussian elimination method .

(for clarifications, contact Siddhartha Sarma)

http://mathworld.wolfram.com/GaussianElimination.html

1.12 Solving a set of linear equations by inverting a matrix.

(for clarifications, contact Siddhartha Sarma)

1. need to find the determinant of a matrix.
2. need to find cofactor matrix (https://en.wikipedia.org/wiki/Adjugate_matrix)
3. perform matrix and vector multiplication.

1.13 Frequency response of a second order system

(for clarifications, contact Siddhartha Sarma)

1. First calculate the amplitude and phase of the transfer function for given range of ω (rads/sec).
2. Plot the response using gnuplot.

1.14 Impedance calculator:

(for clarifications, contact Siddhartha Sarma) Given the values of circuit elements, operating frequency and the arrangement of elements, one need to calculate impedance in both Cartesian and Polar form.

Example: For the following circuit, one possible way to write the arrangement is: $10||(200mF + 2H + (6||500mF))$. Here '||' means components are in parallel and '+' means they are in series.

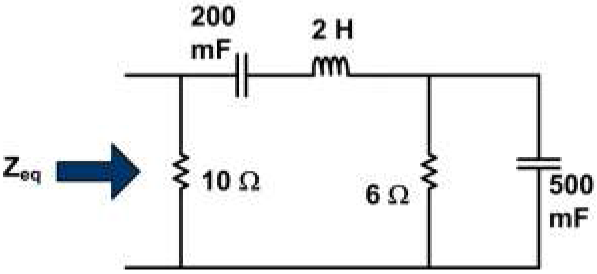


Figure 1: An RLC circuit (Image courtesy: IC 160 Tutorial 7 by Prof. Ramesh Oruganti)

1.15 Find paths between two points in a graph:

(for clarifications, contact Siddhartha Sarma) Given an adjacency matrix, find all the paths between two nodes provided by the user.

1.16 Road planning on a conical mountain

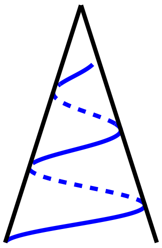


Figure 2: Road along a Conical mountain

(for clarifications, contact Maben Rabi) A mountain is in the shape of a right circular cone. A road is to be planned on it so that i goes round the mountain and doe not have any hair-pin bends. The road is supposed to built such that its length is close to being the shortest possible, while the grade (instantaneous slope of road with the horizontal) never exceeds a given upper limit (such a for example: 12 percent). Write a C program that takes as inputs from the user the parameters of the right circular cone, the start and end points for the road (in azimuthal or Cartesian coordinates as the user pleases), and the maximum grade permitted. The program must output the details of a reasonable choice of road path, meeting given specifications. (finding the best choice of road path is hard. You need to carry ot something called Dynamic optimization, using some basic ideas of Maxima and Minima for functions whose arguments are not points but curves. This topic is called the *Calculus of variations.*)

Extra task: Suppose you can find a model for fuel efficiency as a function of the gear used (which in turn depends on the instantaneous grade), for the typical vehicle on this road, then apart from calculating the shortest road, you must also calculate the road that consumes the least fuel.

1.17 Road planning on a pyramidal mountain

(for clarifications, contact Maben Rabi) Carry out the previous project when the mountain is a pyramid whose base is a convex polygon to be specified by the user. The user also specifies the position of the mountain top. Here, you can actually try to find the optimal road path, and not hust a good sub-optimal one.

1.18 Estimating mean and standard deviation

The user edits a preexisting skeleton of a C function, and writes a mathematical function that takes in a real number and outputs a real number.

Write a program that does some reasonable check that the user's function is a probability density function. If the input function is not negative anywhere, then if possible, carry out some normalization to make it a probability density function.

Take a number n from the user. Learn how to generate a random number of arbitrary density, from another random number that is uniformly distributed between 0 and 1. See:
https://www.av8n.com/physics/arbitrary-probability.htm

For this purpose, you can use the library function rand() to generate a pseudo-random sequence of uniform random numbers. See:
http://stackoverflow.com/questions/6218399/how-to-generate-a-random-number-between-0-and

1.19 Calculating optimal height of a Pratt-Caleb truss

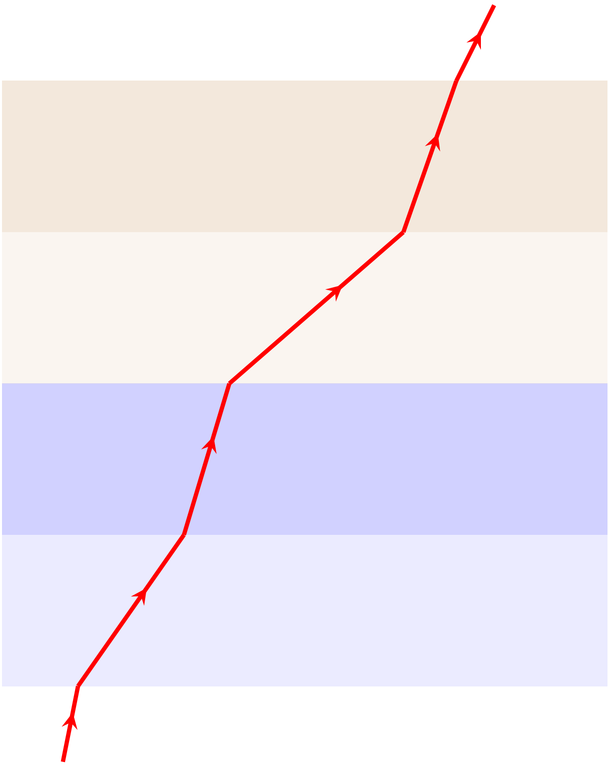
(for clarifications, contact Maben Rabi, Kaustav Sarkar) The Pratt-Caleb truss is a structure for bridge construction. See:
https://www.cs.princeton.edu/courses/archive/fall09/cos323/assign/truss/truss.html
You must write a program to which the user inputs the span of the bridge, and the number of sections of the truss. The program must output: the height of the truss that minimizes cost, where
cost = maximum modulus of force in any member × total length of all members
Your program must also output the maximum force felt in any member, and some indication of whether is is a compressive force, or tension.

1.20 Shortest distance between cities

(for clarifications, contact Maben Rabi) The user provides details of trunk roads (with no city on its path other than the start city and the destination city) available between some cities. The user also specifies the lengths of these roads. For two cities specified by the user, print on screen the shortest route and its length.

1.21 Modelling light passing through different materials

(for clarifications, contact Maben Rabi) Parallel slabs of different optical materials are stacked on top of each other. The user supplies the number of slabs, and the thickness, and refraction index of each slab. The use also supplies the coordinates of two points A, B. Both the points are in a plane that is normal to all slabs.



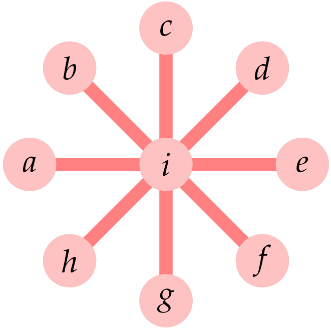
Write a program that will specify (and possibly draw using Gnuplot or ASCII-art) the path of the light ray.

1.22 Designing bus schedules with 'hub-and-spokes' map

(for clarifications, contact Maben Rabi) Here, you are given a map of towns connected by bus services, and you are also given the time it takes for a bus to operate one any road segment connecting two consecutive towns.

User inputs the number of towns, and travel times along road segments. User also inputs minimum buffer time needed for a passenger to be able to swtich from one bus to another. Your task is to find a schedule of bus services. You may use any good criterion to find this schedule. For example, you may want to minimize the maximum travel time possible between any two cities, and at the same time make sure that your schedule guarantees the minimum buffer time for transfers. You may assume that the route map has the following shape.

Suppose that you are given the interconnection map shown below:



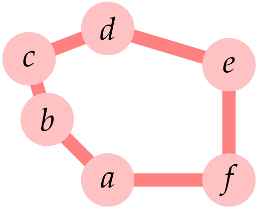
Nodes (towns) from a to h are each individually connected to node i, which is the hub. Each such connection (edge) is bidirectional. Thus for each edge in the map, a bus service runs once a day from the non-hub node to the hub i, and another bus service runs once a day from the hub i back to the non-hub node on that edge. Assume that each trip takes less than 11 hours.

You are given the schedules of all the bus services, and you are asked to check if a passenger starting at any node can catch a connecting bus at the hub, with a minimum waiting time of 20 minutes. This minimum waiting time is enforced to handle slight delays enroute. How will you examine the given set of schedules and find out whether the minimum waiting time requirement is met or not ?

Suppose you are asked to design a set of schedules. Can you design one that minimizes the maximum possible waiting time over all possible travel plans ?

1.23 Designing bus schedules with ring map

(for clarifications, contact Maben Rabi) Same task as in the previous project, but with the map taking the shape of a ring. Suppose that you are given the interconnection map shown below:



1.24 DC circuit analysis using a network of resistors and voltage sources

(for clarifications, contact Maben Rabi) The user provides details of a circuit made up of only DC voltage sources and resistors. List all the DC voltages and currents.

1.25 DC circuit analysis using a network of resistors, capacitors and voltage sources

(for clarifications, contact Maben Rabi) Carry out the previous project with Capacitors also present.

1.26 Finding roots of a function

(for clarifications, contact Maben Rabi) The user edits a skeleton C function and provides a mathematical function of a real variable, and taking real values. The user specifies an interval. Print on screen a list of all roots that you manage to find on this interval.

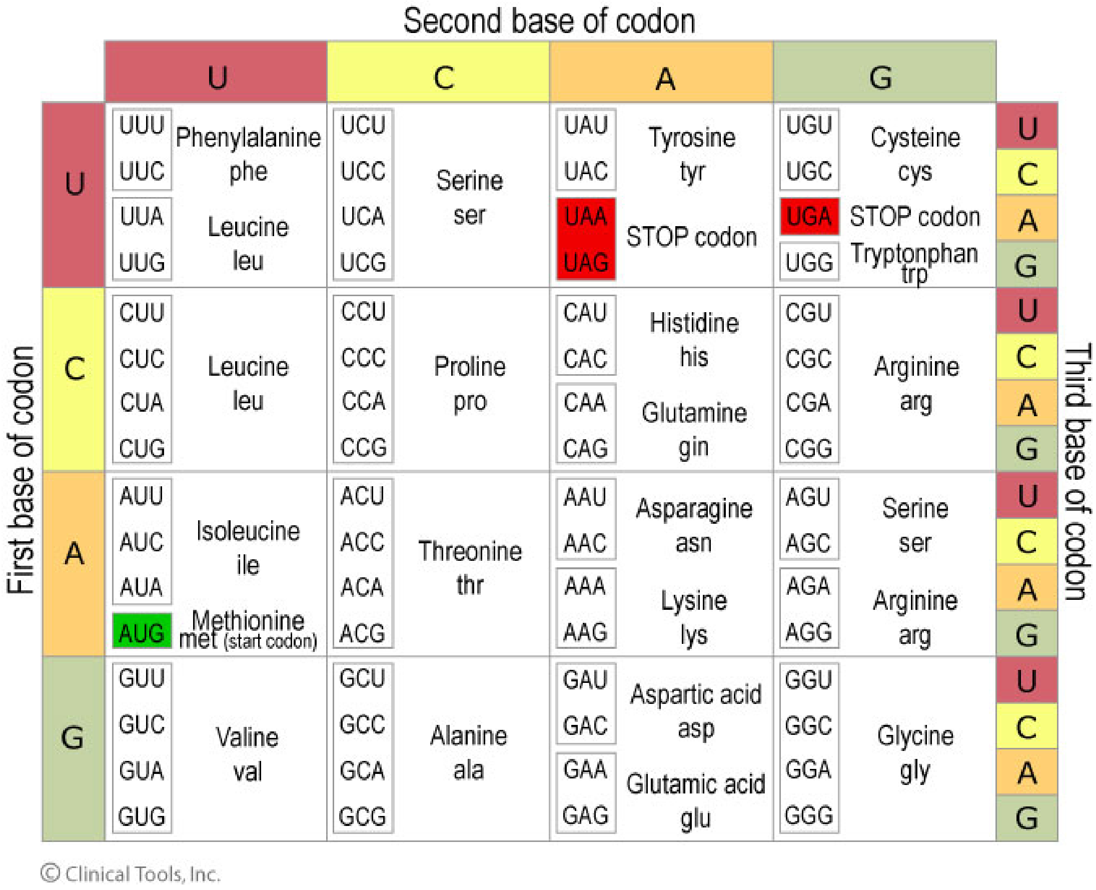
Can you do a better job if the user informs you that the function is a polynomial, and also informs the degree of the polynomial ?

1.27 Transcription from DNA sequence to protein sequence

(for clarifications, contact Tulika Srivastava)

Write a program that can perform translation of a gene to protein: break the DNA string in sets of 3 consecutive letters (non-overlapping), called as codons, read a hash table where each codon refers to an amino acid letter (for example AUG -> Methionine (M) and so on, see attached codon table), report the translated protein sequence and its length (length of protein = length of DNA / 3).

You will be given a data files with the DNA sequences. Sometimes, there may be errors in the DNA letters.



Second base of codon

First base of codon

Third base of codon

	U		C		A		G		
U	UUU	Phenylalanine phe	UCU	Serine ser	UAU	Tyrosine tyr	UGU	Cysteine cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leucine leu	UCA		UAA	STOP codon	UGA	STOP codon	A
	UUG		UCG		UAG		UGG	Tryptonphan trp	G
C	CUU	Leucine leu	CCU	Proline pro	CAU	Histidine his	CGU	Arginine arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Glutamine gin	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Isoleucine ile	ACU	Threonine thr	AAU	Asparagine asn	AGU	Serine ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lysine lys	AGA	Arginine arg	A
	AUG	Methionine met (start codon)	ACG		AAG		AGG		G
G	GUU	Valine val	GCU	Alanine ala	GAU	Aspartic acid asp	GGU	Glycine gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glutamic acid glu	GGA		A
	GUG		GCG		GAG		GGG		G

© Clinical Tools, Inc.

Figure 3: Codon table

1.28 A minimization engine

(for clarifications, contact Maben Rabi) The user edits a skeleton C function and provides a mathematical function of a real variable, and taking real values. The user specifies an interval. Using some intelligent combination of:

- The golden section method with function evaluations,
- The bisection method with derivative evaluations, and,
- Newton's method,
- Any other method that you like,

build an engine for function minimization. For details on these methods, see: *Optimization: insights and applications*, by Jan Brinkhuis, and Vladimir Tikhomirov.

Can you do a better job if the user informs you that the function is a polynomial, and also informs the degree of the polynomial ?

1.29 Finding the convex hull

(for clarifications, contact Maben Rabi) The user specifies a number of points on the plane. Write a program to compute describe (and possibly draw) the convex hull of these points.

1.30 Is one convex hull inside another ?

(for clarifications, contact Maben Rabi) The user specifies two sets of points on the plane. Write a program to report if one convex hull is completely inside the other, or if the two convex hulls never intersect.

1.31 Finding the largest inscribed circle

(for clarifications, contact Maben Rabi) The user specifies a number of points on the plane. Write a program to compute describe (and possibly draw) the largest circle that can be inscribed in the convex hull of the given points.

1.32 Axis parallel rectangles using ASCII-art

(this project takes off from one of the tasks in lab assignmnet sheet 9. For clarifications, contact Maben Rabi) An axis parallel rectangle is a rectangle whose four corners take the special form of coordinates: $(x, y), (x + a, y), (x + a, y + b), (x, y + b)$. User inputs a list of axis-parallel rectangles. Your program must draw in outline, the union of all of these rectangular sets, and filled, the intersection of all of these rectangular sets.

1.33 Vertical histograms using ASCII-art

(for clarifications, contact Siddhartha Sarma) (this project takes off from one of the tasks in lab assignmnet sheet 9. For clarifications, contact Maben Rabi) User inputs some data and bins. Your program must draw a histogram using ASCII-art. But the bars of the histograms must be vertical.

1.34 Geo-location

(for clarifications, contact Siddhartha Sarma)

1. Convert GPS coordinates (represented in degree, minute and second form) to floating numbers.

2. Euclidean distance: given two locations (GPS coordinates), calculate shortest distance between them.

3. For a given a set of coordinates group them based on the distance between them. If the distance is less than d, then two coordinates should belong to the same group. Note that it possible that some coordinates belong to multiple groups.

1.35 Huffman coding for a given a piece of text

(for clarifications, contact Siddhartha Sarma) Input: A text file containing containing around 1000 alphabets.

Tasks:

1. Find the frequency of the alphabets.
2. Find the Huffman code for each of those alphabets.
3. Encoder: Generate a new text file containing binary sequences by reading the input text file.
4. Decoder: When frequency of alphabets and the text file containing the binary sequence are provided, decode it to generate the original text file.

1.36 Run-length coding for images

(for clarifications, contact Siddhartha Sarma) Input: From a black and white image, we will first generate a 2D matrix containing greyscale values.

Tasks:

1. You will read this matrix from a text file.
2. Compression: Using Run-length coding you will generate a compressed file.
3. Decompression: Read the compressed file and decompress it.

1.37 Two species competition model

(for clarifications, contact Siddhartha Sarma) Consider a 2D matrix containing only following three entries (you can initialize it randomly).

- 0: space is available (to be occupied by either species 1 or 2)
- 1: space is taken by species 1
- 2: space is taken by species 2

Initially only 10% cells of this matrix will be occupied, out of which 7% and 3% (of total number of cells) will be occupied by species 1 and 2, respectively.

Now you run a loop for N times, where N can be as large as 1000.

During each loop you update the matrix using following rules. Let us assume current population of species 1 and 2 is N_1 and N_2, respectively.

- Birth process: Fill up the available cells by $b_1 N_1$ and $b_2 N_2$ off-springs of the respective species.
- Death process: Empty $d_1 N_1$ cells for species 1 and $d_2 N_2$ cells for the species 2.
- Competition: Identify the cells in which a member of species 1 is located in one the nearby cells of species 2. Empty $\frac{2}{3}$ of such cells. Similarly, identify the cells in which a member of species 2 is located in one the nearby cells of species 1. Empty $\frac{1}{3}$ of them.

Task: Print the population of each species after every iteration.

1.38 Simulation of 2D random walk with periodic boundary

(for clarifications, contact Siddhartha Sarma) For 2D random walk with periodic boundary, we first define the boundaries of 2D plane. Lets consider symmetric boundaries. We are interested in the rectangular region is characterized by the coordinates $(-x_{max}, -y_{max})$, $(x_{max}, -y_{max})$, (x_{max}, y_{max}) and $(-x_{max}, y_{max})$. Now a person can start walking from any arbitrary point within this rectangular region. At each time step the person moves to next coordinates according to the following formula:

$$x(t+1) = x(t) + dx$$
$$y(t+1) = y(t) + dy$$

Here, $x(t), y(t)$ represent the coordinates at tth time step. dx and dy values are generated randomly. If at any point of time the person moves out of the boundary, then will use the following rule to bring him/her within the boundary:

$$\begin{aligned}\tilde{x}(t+1) &= x(t+1) - 2 * x_{max} \quad if \quad x(t+1) > x_{max}\\ \tilde{x}(t+1) &= x(t+1) + 2 * x_{max} \quad if \quad x(t+1) < -x_{max}\end{aligned}$$

Similar rule will be applied for y coordinates also.

Output: Using 2D array (e.g., coordinates [2][1000]) store all the coordinates obtained due to periodic random walk. Trace the random walk using lineplot in Gnuplot. Reference: https://en.wikipedia.org/wiki/Random_walk

1.39 Hermitian matrices

(for clarifications, contact Siddhartha Sarma)

1. Find complex conjugate of a matrix (need to represent the complex numbers in "$a + ib$" form).
2. Find whether a complex matrix is a Hermitian matrix or not.
3. Express a square complex matrix as a sum of a Hermitian and a skew-Hermitian matrix.

1.40 Lightweight markup language

(for clarifications, contact Siddhartha Sarma)

Lightweight markup languages are used to create simple html webpages. (Source https://en.wikipedia.org/wiki/Lightweight_markup_language).

Write a code that will read a text file and convert it into a html page. Example: Jemdoc (http://jemdoc.jaboc.net/example.html), DLLUP (https://daniel.lawrence.lu/programming/dllup/)

Tips on plotting using Gnuplot from inside a C program

See: https://www.cs.hmc.edu/~vrable/gnuplot/using-gnuplot.html
http://people.duke.edu/~hpgavin/gnuplot.html
http://gnuplot.sourceforge.net/docs_4.0/gpcard.pdf
http://linux.byexamples.com/archives/487/plot-your-graphs-with-command-line-gnuplot/
and the code on the next page.

```c
/*
  Code written by Ben B on 3rd August 2011, and made available at:
  http://stackoverflow.com/questions/3521209/making-c-code-plot-a-graph-auto
*/

#include <stdlib.h>
#include <stdio.h>
#define NUM_POINTS 5
#define NUM_GNUPLOT_OPTIONS 5

int main()
{
    char * commandsForGnuplot[] = {"set style data linespoints","set title 'Simple plotting example'", "set xlabel 'x coordinate'","set ylabel 'y coordinate'","plot 'data.temp'" };
    double xvals[NUM_POINTS] = {1.0, 2.0, 3.0, 4.0, 5.0};
    double yvals[NUM_POINTS] = {5.0 ,3.0, 1.0, 3.0, 5.0};
    FILE * temp = fopen("data.temp", "w");

    /*Opens an interface that one can use to send commands as if they were typing into the
     *     gnuplot command line.  "The -persistent" keeps the plot open even after your
     *     C program terminates.
     */
    FILE * gnuplotPipe = popen ("gnuplot -persistent", "w");

    int i;
    for (i=0; i < NUM_POINTS; i++)
    {
    fprintf(temp, "%lf %lf \n", xvals[i], yvals[i]); //Write the data to a temporary file
    }

    for (i=0; i < NUM_GNUPLOT_OPTIONS; i++)
    {
    fprintf(gnuplotPipe, "%s \n", commandsForGnuplot[i]); //Send commands to gnuplot one by one.
    }
    fflush(gnuplotPipe);
    return 0;
}
```

gnuplotFromC.c