If you are using a jupyter notebook (recommended), then keep all your programs in a single notebook. A good programming style is to define a function for one task with clearly defined input (arguments) and output. For plots you may use matplotlib (if you are using python) or gnuplot (if you are using c or fortran) or LsqFit module if you are using Julia.

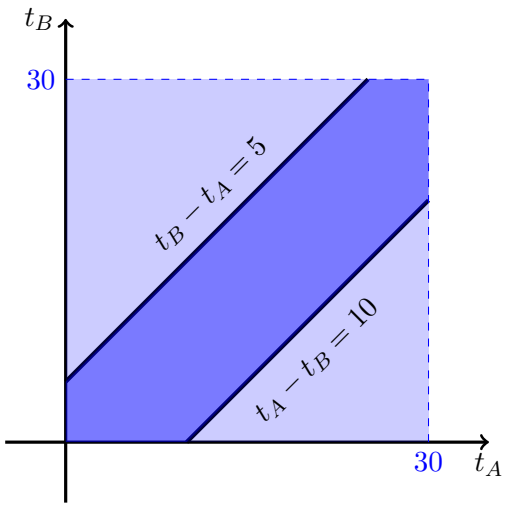
If you are planning to submit separate programs, then please follow the guideline below:

- Keep all files of a worksheet in a single folder.
- Follow a systematic naming convention. You may name the program files as `Q1.py` or `Q1a.py`, `Q1b.py` for question 1 (if you have created multiple files for a single question). The data file should be named as `Q1-data-a.dat` and so on.
- Finally compress the entire folder as a single `.zip` or `.tgz` (using `tar cvfz archive.tgz folder-name/`, and submit the file in WeLearn.

---

1. Write a program that generates a random integer between n_1 and n_2 (as supplied by the user). The function should use the standard uniform random number generator (typically between 0 and 1).

2. Two friends A and B agree to meet in the canteen during their 30 minute tiffin break. A will wait for B for 5 minutes if he shows up first at the canteen. While B, the more patient of the two, will wait for 10 minutes if she were the first one at the canteen. Calculate the probability that the two will meet.



A graphical depiction of the "meeting problem". The light blue rectangle denotes the region in the $t_A - t_B$ plane which covers all possible arrival times – while the dark blue region is where a meeting takes place.

The following lines of code simulate a single such event by generating two random values t_A and t_B between 0 and 30 and will set the variable meet to True if they meet, while if they do not meet, the variable has the value False.

```
from random import random
meet = False
tA = 30*random()
tB = 30*random()
if tA<tB:
    if tB-tA<5:
        meet = True
else:
        if tA-tB<10:
            meet = True
```

By using similar code, try to estimate the probability that the two friends meet. Note that to estimate the probability, we will have to generate many instances of this event and find the fraction of cases where the meeting actually takes place.

3. *Buffon's needle*: consider the floor of a room having parallel strips of wood of width d. The boundary of the strips are marked by thin lines. Now consider a needle of length ℓ thrown on the floor. Using the steps explained in the class, find the probability that the needle crosses a line. Consider both the cases, $\ell \geqslant d$ and $\ell < d$.

4. Consider a random walk process of total number of steps N. Verify that the probability of reaching a position n after N steps is given by

$$P_n = \frac{2}{\sqrt{2\pi N}} \exp\left(-n^2/2N\right)$$

   You may proceed in the following way:

   1. Define a function that simulates taking a walk by using $x_n = x_{n-1} + S_n$, where x_n is the position after n^{th} step and S_n is the n^{th} step (either of $+1$ or -1). The function should take the total steps as argument and return only the final position.
   2. For the walk of N steps, there will be $N + 1$ possible final positions. Create an array of size $N + 1$ to store the occurences of a particular position as discussed in the class.
   3. Perform 100000 walks each having 50 steps. Plot the resulting data along with the exact formula given above. Write your observations as documentation.

5. In this problem, you need to check whether $\langle x^2 \rangle = N$. You may proceed the following way:

   1. Use an array to store the positions of all walkers (say, 100000). All walkers start from the origin (0).
   2. After each step, update the position and calculate $\langle x^2 \rangle$ and store.
   3. Plot $\langle x^2 \rangle$ as a function of steps.