Physics 2200 Midterm II Project

- In probability theory, the *birthday problem* or *birthday paradox* concerns the probability that, in a set of n randomly chosen people, at least two of them will have the same birthday. This probability obviously reaches 100% when the number of people reaches 366, i.e. when the number of people in the group is more than the number of different birthdays. (We assume that there are only 365 possible birthdays.) However, 99.9% probability is reached with just 70 people, and 50% probability with 23 people. These conclusions are based on the assumption there are 365 days in a year, and that each day of the year is equally probable for a birthday.

  Use Monte Carlo method to calculate the probability, p_b, that, in a set of **n** randomly chosen people, at least a pair of them have the same birthday. Plot the graph $p_b(n)$ for $0 \leq n \leq 50$. Use the graph to estimate (with $\sim 5\%$ relative error) the numbers of people, $n_{\frac{1}{4}}$ and $n_{\frac{3}{4}}$, such that $p(n_{\frac{1}{4}}) = \frac{1}{4}$ and $p(n_{\frac{3}{4}}) = \frac{3}{4}$

- *Reproducible research* is a concept relevant for a data analysis or computational project. It requires that the data and the computer code which went into doing that analysis be available to others so that they might examine what you've done and reproduce your findings.

  In the spirit of reproducible research approach, your code must be 'compilable' (using make) and 'runnable' and produce the numerical results and graphics without any grader's intervention.

  As an exception from the general rule, you are allowed to 'extract' $n_{\frac{1}{4}}$ and $n_{\frac{3}{4}}$ by the visual inspection of a graph (which must be automatically generated after/during your program run and available for the grader's inspection.)

- You are welcome to discuss the project's ideas with others in order to better understand them but the work you turn in must be your own. In particular, you must write your own code, run your own calculations, and communicate and explain the results in your own words.

- You are encouraged to talk to the instructor at all stages of your project.

Name: _____

Date: _____

Question:	1	2	3	4	5	6	Total
Points:	10	10	40	20	10	10	100
Score:							

Required project steps:

1. (10 points) Create a GitLab repository for your project, call it **m2**, and upload there all your C code, your **self-documented** Makefile, and .gitignore file.
2. (10 points) Consult the Guide to software licensing for scientist and chose a license for your project.
3. (40 points) Your C code should (a) produce no warnings or errors when compiled, (b) produce the results, (c) use reliable components, (d) be well commented, (e) be written elegantly, and (f) properly and consistently formatted.
   - In your code use random number generators from the GNU Scientific Library (GSL) as following:

```
gsl_rng *r = gsl_rng_alloc (gsl_rng_taus);
gsl_rng_set (r, seed);
```

   ...

```
i = gsl_rng_uniform_int (r, 365UL);
```

   ...

```
gsl_rng_free (r);
```

   - Design your program to use an array, e.g. int bday[dim], to keep track of birthdays and their 'collisions'.
   - Consider writing the C functions with the following declaration

```
void init(int bday[], unsigned long dim);
```

   that cleans the array bday after each simulation run;

```
int collision (int n, int bday[],
               unsigned long dim, gsl_rng *r);
```

   that generates a random birthday and checks if another birthday already exist on the same date;

```
double simulate (int n, long nrep, int bday[],
                 unsigned long dim, gsl_rng *r);
```

that does the simulations and returns the required probability.

- Use your functions to write a C program that loops over the value of the parameter n, from $n = 0$ to $n = 50$, and prints the probability, p_b of a 'collision' of birth dates.

4. (20 points) **Your repository should contain snapshots of your code as you develop it!**
5. (10 points) Plot a graph of $p_b(n)$. Use a scripting program, e.g. gnuplot.
6. (10 points) Create README.md file that describes (in some details details) the design, the implementation, and the results of your project.

Your results:

$n_{\frac{1}{4}}$: _____

$n_{\frac{3}{4}}$: _____