ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ECE_Y215: ΔΙΑΔΙΚΑΣΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Δραστηριότητα με Εξωτερικές Βιβλιοθήκες της C: Γεννήτριες Ψευδοτυχαίων Αριθμών (PRNG) με χρήση της Βιβλιοθήκης GSL (GNU Scientific Library)

Εισαγωγή

Σκοπός της παρούσας δραστηριότητας είναι η εξοικείωση με την βιβλιοθήκη GSL (GNU Scientific Library) [1] και ειδικότερα με τις γεννήτριες ψευδοτυχαίων αριθμών (PRNG - Pseudo Random Number Generators) [2, 7, 10]. Ιστορικά, ο πρώτος αλγόριθμος δημιουργίας ψευδοτυχαίων αριθμών προτάθηκε από τον John von Neumann το 1949 [8] και εφαρμόστηκε για πρώτη φορά στον ENIAC [9].

Οι γεννήτριες ψευδοτυχαίων αριθμών χρησιμοποιούνται στον επιστημονικό υπολογισμό σε τομείς όπως η κρυπτογραφία, η προσομοίωση και μοντελοποίηση φυσικών φαινομένων, για παράδειγμα στην προσομοίωση Monte Carlo για τον υπολογισμό του αριθμού π [3, 4].

Δυο από τα βασικά χαρακτηριστικά των PRNG είναι η ομοιόμορφη κατανομή των παραγομένων αριθμών καθώς και η μη-προβλεψιμότητά τους [10]. Εσφαλμένες επιλογές PRNG μπορεί να δημιουργήσουν σημαντικά προβλήματα στην εύρυθμη λειτουργία των συστημάτων.

Ζητούμενα

Δημιουργήστε ένα πρόγραμμα στη C το οποίο να αξιολογεί τις ακόλουθες γεννήτριες ψευδοτυχαίων αριθμών:

- *rand* της βασικής βιβλιοθήκης <stdlib.h>
- *Mersenne Twister* της GSL <gsl/gsl_rng.h>

Παραδοτέα

Η δραστηριότητα περιλαμβάνει τρεις εκδόσεις. Στο Παράρτημα δίνεται ένας βασικός σκελετός κώδικα για κάθε έκδοση, ο οποίος περιλαμβάνει τις δηλώσεις των συναρτήσεων τις οποίες πρέπει να υλοποιήσετε. Ωστόσο, είστε ελεύθεροι να δημιουργήσετε ανεξάρτητα από τον ενδεικτικό κώδικα όποιο κώδικα εσείς επιθυμείτε.

Έκδοση 1: Δημιουργία, αρχικοποίηση των γεννητριών και έλεγχος λειτουργίας

Ο σκοπός της πρώτης έκδοσης είναι να αποκτήσετε εξοικείωση με τις δύο γεννήτριες και να παραδώσετε ένα πρόγραμμα το οποίο να ελέγχει την λειτουργία των γεννητριών στα πλαίσια 10 δοκιμαστικών κληρώσεων στο διάστημα [1 … 45].

Στην Εικόνα 1 του Παραρτήματος υπάρχει ενδεικτικός πηγαίος κώδικας και αποτέλεσμα εκτέλεσης της πρώτης έκδοσης. Στον πηγαίο κώδικα φαίνονται οι δηλώσεις των συναρτήσεων χωρίς τη υλοποίησή τους. Για παράδειγμα, οι συναρτήσεις *seed_gsl_mt_rng()* και *seed_rand_rng()* αρχικοποιούν τις δυο γεννήτριες, ενώ η συνάρτηση *draws()* δέχεται ως ορίσματα τον αριθμό των κληρώσεων και το εύρος των προς κλήρωση αριθμών και εκτυπώνει το αποτέλεσμα της κάθε κλήρωσης.

Έκδοση 2: Έλεγχος της ομοιόμορφης κατανομής των τυχαίων αριθμών για 10^9 κληρώσεις

Ο σκοπός της δεύτερης έκδοσης είναι να μελετήσετε την ομοιόμορφη κατανομή των παραγόμενων αριθμών. Για αυτό το λόγο, δημιουργήστε 10^9 κληρώσεις στο διάστημα [1 …

45]. Για κάθε κλήρωση και για κάθε PRNG αποθηκεύστε τη συχνότητα εμφάνισης του κάθε αριθμού σε διανύσματα (vector) μέσω της βιβλιοθήκης <gsl/gsl_vector.h>.

Στην Εικόνα 2 του Παραρτήματος υπάρχει ενδεικτικός πηγαίος κώδικας και αποτέλεσμα εκτέλεσης της δεύτερης έκδοσης. Ο ενδεικτικός κώδικας χρησιμοποιεί τα διανύσματα *v_rand_frequency* και *v_mt_frequency* για την αποθήκευση της ποσοστιαίας κατανομής των κληρώσεων. Επίσης χρησιμοποιεί τη συνάρτηση *print_frequency_tables()* για την ποσοστιαία εμφάνιση του κάθε αριθμού στις κληρώσεις.

Έκδοση 3: Έλεγχος της προβλεψιμότητας των αποτελεσμάτων

Ο σκοπός της τρίτης έκδοσης είναι να αποδείξετε ότι η *rand* είναι προβλέψιμη μετά από 2^{31}-1 κληρώσεις (2.147.483.647). Για να το κάνετε αυτό, αποθηκεύστε το ιστορικό των παραγομένων τυχαίων αριθμών για τις παραπάνω κληρώσεις σε δυο διανύσματα (π.χ., για την *rand PRNG* στο διάνυσμα *v_rand_results*, και *v_mt_results* αντίστοιχα για την *MT PRNG*).

Στη συνέχεια υλοποιήστε την συνάρτηση *forecast()* η οποία θα κάνει μερικές επιπλέον κληρώσεις. Προσπαθήστε να προβλέψετε το αποτέλεσμα της κάθε κλήρωσης χρησιμοποιώντας τα ιστορικά αποτελέσματα. Θα διαπιστώσετε ότι έχετε 100% επιτυχής πρόβλεψη στην *rand PRNG* ενώ δεν υπάρχει αντίστοιχη πρόβλεψη για την *MT PRNG* (Εικόνα 3 στο Παράρτημα).

Αν θέλαμε να πετύχουνε ένα πλήρη κύκλο με την *Mersenne Twister PRNG* θα έπρεπε να είμαστε σε θέση να προσομοιώσουμε 2^{19937}-1 κληρώσεις. Ο αριθμός αυτός δεν μπορεί να εκτυπωθεί με την βασική βιβλιοθήκη της C, αντιθέτως μπορεί να γίνει με χρήση της εξωτερικής βιβλιοθήκης GMP [6]. Ο αστρονομικός αυτός αριθμός φαίνεται στην παρακάτω εικόνα (Εικόνα 1)[1]. Θα μπορούσαμε να προβλέψουμε τις κληρώσεις MT όπως κάναμε με την rand σε έναν συμβατικό υπολογιστή όπως αυτούς που έχουμε στο σπίτι ή στο γραφείο μας;

```
2^19937 -1=
86036624185527798714277230489229107831218302956810947278479600043958203769904489362867808822772462247697604675904274482973301650699960960114265102135442982794960010728172217969339881322516969756360617266
80361315624243035822906791101831346644980001977508567795463513285417120314459785197542980720881124345707706959917509774358404325185005945762154925468894620794166378539456961300474849868905477392137832491
56753195217934092630926692144613626006534827706404796271808904087579134810144411749502931035449454177145941164952530269651890996448928571605977843601746684739308565841908490033383135434827064549735030893356
77392540860879212196171680804076083787536884617541687856126099973753383041904691421991756849215213470762931496263538628758202746049749617396604758690423030388981372592942074530896489049017742542306397846668
86337385207535330552198959603698223892645571026341972441596738599757730396233708568658045145725340544786418317834433612832620086144916714156486525314975582082361451862046654156389683333735607835012745599843
32011506538297503859091217221385894701388965847801850561945251450537721394851307919525565642720017906460745102130187901384428893508476079659908838224610720764186663173724532902133122626572272533497139290792
83010802918760950390493952572072611180319997179417003075995908291960324138075283523549834879593825027350275857741086888762644293649241453867227272790804554153232216041318994072793122175269755095105973042183
88329564967991168923186191894863211882345653679994942361466336644215511000973344013824092611947739172015893762422045566654777663952999770808632616115333255060900682276701584742718816729292724146987988928057
72619104649412225593090917511927042012442749028508054916851731437123710330428459941212297797873445968854024610952124461950181580559107412839414485605643998563147781212233858411220249881654567586616604664104
93779447426568830550310499263678552542632764343784637585202114747352585245003713618340540547565456559019737644795606440021334455401336338227045611488982034870446136148601740953206684167406505182829664040
90194900652955265174201868057295287506971461837439411695112627584814086600093473182110263265980788078459017206063802428073478999235089974497696744606744058331738812460389991978933661767652739211557652200834
28913787249799375494735550846806441885662994798103720630911971366886000453279358056056003536921584043690213641060961106377077294994816351366218284902973243276226078316888931259676373861230520949971599307965
92446791764387119014683967205076306425511133854433942412767649153742469610831605548214354457844624203536517399988773526888490555752517945402883539738786952414132558695559339902534889240991606954431662441222
55115176496853328128435366545903533112438828181467028321195856691901781397420760013654313982210333640746344147939517175821024478597939742414526684616675966562875571076153831468425264867302070167621440212792
23883144662294686618899378784452532744723293162138439448688195296234188359990099988216580991676038887054428127336908109726995849659431293811847179493990157892956389583401543628831137548957607635575481109428
62838526648993021271058142902403585641318776585354928915869637001032767560845440734913040739043156001985234956166664273564245502530963579520003175459847615956324515844713000161888234607830515678239906782765
40694704449087230732357130349263391714602089240308624500587753328914388825153324838836164135443569613693371291736810825798376004404734167750689776152411226275638991189531914469576824201789107405076555817
30856905016513454773709951405950413296909194949677422879373104685938931231548760538599927857363625421001296345513597845936345962894426201902112669580844767396172530592851455225414189923524291371791702392
14099934015555390213056795360614289192469804110392746219799931100255676223429132488987960482411507739434241957054947575490775012508479737961245568111896503300425431163233951424111827927563389007929670060147888224086375232955579166346952596780874555168628291559766535185236033299737008255302770011089401899940328921858957951613151240721917203815248043107283722420818576650640804995592568864895429433053190914397270557592157278357690660872198887451652787361776701836315629169085257940022041702233717493809787132931241439059527533178420107805047707030455467627327334151784453671718478930428864123808526747380558573684
86440248131576376282963205526063694701590027414898706598590772521638643079028688462137156536036292203676251078653457087355662185824067790013922253395793932390325741557047657699185786887763567426142756125
78018360110761705869895704487049526757893282712992960009715316703237002793320644058435317170769220091774229794422576024189529146549624856449221672981681924353130813965061464569818767299778491769263357021668
24531403266593628570369708059755027820029324566032664295138348348582770247419047455103120458908707244633268386452906223415613172234525840877745544326827335105637447683591614846797673905811972526529059953
67312693416252611141067356232769372590776394933456878547211895177828923390864611001462693053211752201837733413352413968218858414823660102294322122329115324539775378290047891030284600755345652076332281427333
36567743493962328230191313498160772436163111556764893482342095660438393604563394806384402716363938415533626421076940945495742639409289433117619315907897327367940238723509286385533220308961047961064272062677
17426776946731415629717872335702735593729467241636649504030342134047946343574102290844667686403959369345536737071103653320055891753858796179302791632347768912602627738797703796400891089965817582806736279
19761140644556469675050334233438185620392177526858331151811023003049782512696993886125767923252459377168212111513887653364930709528002812925568159323867345917720820153284529847028436633715378375856090866840
92539388334669529674941863889562621864064613135511164755031440100512568030101890333405145340941554694963733602293482682920747697780908189275779551642269022052942270499495987532055715391150602039374752078
28155397141972694175893823095464240804441833879267870803261051309567971606366040984282726 3
```

Εικόνα 1. Η περίοδος της Mersenne Twister PRNG

Καταγράψτε τις παρατηρήσεις σας σε μια μικρή έκθεση σχετικά με την επιλογή της γεννήτριας τυχαίων αριθμών που θα κάνατε. Ανεβάστε τις εκδόσεις του κώδικα σας μαζί με την αναφορά σας στο e-class σε μορφή συμπιεσμένου αρχείου και τίτλο τον αριθμό μητρώου σας.

---

[1] *Σημείωση: Μην προσπαθήσετε να εκτυπώσετε αυτό τον αριθμό χρησιμοποιώντας τους βασικούς τύπους της C οι οποίοι δεν μας επιτρέπουν την αποθήκευση πολύ μεγάλων αριθμών [5]. Η εκτύπωση της περιόδου έγινε με την χρήση της βιβλιοθήκης GMP [6] με την οποία θα ασχοληθούμε στις επόμενες ασκήσεις.*

Αναφορές

[1] https://www.gnu.org/software/gsl
[2] https://en.wikipedia.org/wiki/Applications_of_randomness
[3] https://en.wikipedia.org/wiki/Monte_Carlo_method
[4] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/video-lectures/lecture-20
[5] https://en.wikipedia.org/wiki/C_data_types
[6] https://gmplib.org/manual
[7] https://en.wikipedia.org/wiki/Pseudorandom_number_generator
[8] https://en.wikipedia.org/wiki/Middle-square_method
[9] https://en.wikipedia.org/wiki/ENIAC
[10] https://arxiv.org/pdf/1811.04035.pdf

Παράρτημα

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include<unistd.h>
#include <gsl/gsl_rng.h>

void seed_gsl_mt_rng();
void seed_rand_rng();
void draws(unsigned long int draws, int range) ;
static gsl_rng *r;

int main()
{
   seed_gsl_mt_rng();
   seed_rand_rng();

   printf("PRNGs seeded succesfully. Initiate trials.\n");
   draws(10, 45);

    gsl_rng_free(r);
   return 0;
}
```

```
PRNGs seeded succesfully. Initiate trials.
        [Draw]          [Rand]          [Mersen Twister]
        [1]             25               14
        [2]             19               8
        [3]             40               30
        [4]             32               2
        [5]             35               24
        [6]             20               12
        [7]             42               27
        [8]             25               10
        [9]             33               30
        [10]            14               32
```

Εικόνα 1. Ενδεικτικός πηγαίος κώδικας και παράδειγμα εκτέλεσης της πρώτης έκδοσης

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include<unistd.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_vector.h>
void seed_gsl_mt_rng();
void seed_rand_rng();
void draws() ;
void print_frequency_tables();

static gsl_rng *r;
static gsl_vector * v_rand_frequency;
static gsl_vector * v_mt_frequency;
static unsigned long int nof_draws=1000000000;
static int range=45;

int main()
{
   seed_gsl_mt_rng();
   seed_rand_rng();

   v_rand_frequency = gsl_vector_alloc (range);
   v_mt_frequency = gsl_vector_alloc (range);

   draws();
   print_frequency_tables();

   gsl_vector_free (v_rand_frequency);
   gsl_vector_free (v_mt_frequency);
   gsl_rng_free(r);
   return 0;
}
```

```
---Frequency tables ---
 [Number]        [Perc. Freq.Rand Rng]   [Perc. Freq.MT Rng]
[1]              2.222                   2.223
[2]              2.222                   2.223
[3]              2.222                   2.223
[4]              2.222                   2.223
[5]              2.223                   2.223
[6]              2.223                   2.223
[7]              2.222                   2.223
[8]              2.222                   2.223
[9]              2.222                   2.223
[10]             2.223                   2.223
[11]             2.223                   2.223
[12]             2.222                   2.223
[13]             2.223                   2.223
[14]             2.222                   2.223
[15]             2.222                   2.223
[16]             2.222                   2.223
[17]             2.222                   2.223
[18]             2.222                   2.223
[19]             2.222                   2.223
[20]             2.222                   2.223
[21]             2.222                   2.223
[22]             2.223                   2.223
[23]             2.223                   2.223
[24]             2.223                   2.223
[25]             2.222                   2.223
[26]             2.221                   2.223
[27]             2.222                   2.223
[28]             2.222                   2.223
[29]             2.222                   2.223
[30]             2.223                   2.223
[31]             2.223                   2.223
[32]             2.222                   2.223
[33]             2.222                   2.223
[34]             2.222                   2.223
[35]             2.223                   2.223
[36]             2.222                   2.223
[37]             2.223                   2.223
[38]             2.222                   2.223
[39]             2.223                   2.223
[40]             2.222                   2.223
[41]             2.222                   2.223
[42]             2.222                   2.223
[43]             2.222                   2.223
[44]             2.222                   2.223
[45]             2.222                   2.223
```

Εικόνα 2. Ενδεικτικός πηγαίος κώδικας και παράδειγμα εκτέλεσης δεύτερης έκδοσης

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include<unistd.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_vector.h>
void seed_gsl_mt_rng();
void seed_rand_rng();
void draws() ;
void forecast();

static gsl_rng *r;
static gsl_vector * v_rand_results;
static gsl_vector * v_mt_results;
static unsigned long int nof_draws=RAND_MAX-1;
static int range=45;
int main()
{
   seed_gsl_mt_rng();
   seed_rand_rng();

   v_rand_results = gsl_vector_alloc (nof_draws);
   v_mt_results = gsl_vector_alloc (nof_draws);

   draws();
   forecast();

   gsl_vector_free (v_rand_results);
   gsl_vector_free (v_mt_results);
   gsl_rng_free(r);
   return 0;
}
```

```
**********PREDCITING RAND**********
 [Rand Predict vs. Draw ]         [GSL MT Predict vs. Draw]
       31  vs.  31                      8  vs.  29
       11  vs.  11                     32  vs.  41
       41  vs.  41                      9  vs.  24
        3  vs.   3                     26  vs.  22
       10  vs.  10                      6  vs.  41
        0  vs.   0                     38  vs.  13
       37  vs.  37                     25  vs.   7
       32  vs.  32                     12  vs.  35
        3  vs.   3                     24  vs.   9
       42  vs.  42                     33  vs.   2
       36  vs.  36                      1  vs.   7
       39  vs.  39                     13  vs.  17
       24  vs.  24                     12  vs.   9
       34  vs.  34                     16  vs.  11
        6  vs.   6                      4  vs.  16
       38  vs.  38                     19  vs.  28
       34  vs.  34                     39  vs.  11
       22  vs.  22                      1  vs.  15
        8  vs.   8                     23  vs.  31
       35  vs.  35                      3  vs.  24
        7  vs.   7                     36  vs.  42
        3  vs.   3                     36  vs.  10
       18  vs.  18                     24  vs.  12
       15  vs.  15                     41  vs.   4
       43  vs.  43                      9  vs.  25
       35  vs.  35                     33  vs.  13
       13  vs.  13                     11  vs.  23
       23  vs.  23                      8  vs.  38
       33  vs.  33                      2  vs.  17
       42  vs.  42                      3  vs.  29
```

Εικόνα 3. Ενδεικτικός πηγαίος κώδικας και παράδειγμα εκτέλεσης της τρίτης έκδοσης