November 24, 2020 at 13:24

1. Intro. This little program makes empirical tests by which I can check the accuracy of the SETSET program. (See that program for explanations.)

```
#define m 12     /* size of each hand */
#define n 100000000     /* number of random trials */
#include <stdio.h>
#include "gb_flip.h"     /* use the Stanford GraphBase random number routines */
  char deck[81];
  char occ[81];
  char z[3][3] = {{0,2,1},{2,1,0},{1,0,2}};     /* x + y + z ≡ 0 (mod 3) */
  char third[81][128];
  main()
  {
    register int j, k, t;
    int reps, count;
    ⟨Initialize 2⟩;
    while (1) {
      count = 0;
      for (reps = 0; reps < n; reps++) {
        ⟨Shuffle the deck 4⟩;
        ⟨Increase count if there are no SETs 3⟩;
      }
      printf("%d/%d\n", count, n);
    }
  }
```

2. `#define pack(a,b,c,d) (((a) * 3 + (b)) * 3 + (c)) * 3 + (d)`

```
⟨Initialize 2⟩ ≡
  gb_init_rand(0);
  for (k = 0; k < 81; k++) deck[k] = k;
  for (k = 81 − m; k < 81; k++) occ[k] = 1;
  {
    int a, b, c, d, e, f, g, h;
    for (a = 0; a < 3; a++)
      for (b = 0; b < 3; b++)
        for (c = 0; c < 3; c++)
          for (d = 0; d < 3; d++)
            for (e = 0; e < 3; e++)
              for (f = 0; f < 3; f++)
                for (g = 0; g < 3; g++)
                  for (h = 0; h < 3; h++)
                    third[pack(a,b,c,d)][pack(e,f,g,h)] = pack(z[a][e], z[b][f], z[c][g], z[d][h]);
  }
```

This code is used in section 1.

3. ⟨Increase *count* if there are no SETs 3⟩ ≡

```
  for ($j = 81 - m$; $j < 80$; $j$++)
    for ($k = j + 1$; $k < 81$; $k$++)
      if ($occ[third[deck[j]][deck[k]]]$) goto $set\_found$;
  $count$++;
$set\_found$:
```

This code is used in section 1.

4. ⟨Shuffle the deck 4⟩ ≡

```
  for ($j = 81$; $j > 81 - m$; $j$--) {
    $t = deck[j - 1]$;
    $occ[t] = 0$;
    $k = gb\_unif\_rand(j)$;
    $deck[j - 1] = deck[k]$;
    $occ[deck[k]] = 1$;
    $deck[k] = t$;
  }
```

This code is used in section 1.

5. Index.

SETSET-RANDOM