PROGRAM FOR FINDING OUT NUMBER OF SMARANDACHE DISTINCT RECIPROCAL PARTITION OF UNITY OF A GIVEN LENGTH

(Amarnath Murthy, S.E. (E&T), Well Logging Services, Oil And Natural Gas Corporation Ltd., Sabarmati, Ahmedbad, India-380005.)
e-mail: amarnath_murthy@yahoo.com

ABSTRACT: Smarandache Distinct Reciprocal partition of unity for a given length 'n' is defined as the number of ways in which unity can be expressed as the sum of the reciprocals of 'n' distinct numbers. In this note a program in 'C' is given.

// This is a program for finding number of distinct reciprocal partitions of unity of a given length written by K Suresh, Software expert, IKOS, NOIDA, INDIA.

#include<stdio.h>
#include<math.h>

unsigned long TOTAL;
FILE* f;
long double array[100];
unsigned long count = 0;

void try(long double prod, long double sum, unsigned long pos)
{
    if( pos == TOTAL - 1 )
    {
        // last element..
        long double diff = prod - sum;
        if( diff == 0 ) return;
        array[pos] = floor(prod / diff);
        {
            fprintf(f, "(%ld) %ld", ++count, (unsigned long)array[0]);
            int i;
            for(i = 1; i < TOTAL; i++) fprintf(f, "%ld", (unsigned long)array[i]);
            fprintf(f, "\n");
            fflush(f);
        }
    }
}
long double i;
if (pos == 0)
    i = 1;
else
    i = array[pos-1];

while(1) {
    i++;
    long double new_prod = prod * pow(i, TOTAL-pos);
    long double new_sum = (TOTAL-pos) * (new_prod / i);
    unsigned long j;
    for(j = 0; j < pos; j++) new_sum += new_prod / array[j];
    if (new_sum < new_prod)
        break;

    new_prod = prod * i;
    array[pos] = i;
    new_sum = prod + sum * i;
    if (new_sum >= new_prod) continue;

    try(new_prod, new_sum, pos+1);
}
return;

main()
{
    printf("Enter no of elements ?");
    scanf("%ld", &TOTAL);
    char frame[256];
    sprintf(frame, "rec%ld.out", TOTAL);
    f = fopen(frame, "w");
    fprintf(f, "No of elements = %ld\n", TOTAL);

    try(1, 0, 0);
    fflush(f);
    fclose(f);
    printf("Total %ld solutions found.\n", count);
Based on the above program the following table is formed.

<table>
<thead>
<tr>
<th>Length</th>
<th>Number of Distinct Reciprocal Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2320</td>
</tr>
<tr>
<td>7</td>
<td>245765</td>
</tr>
</tbody>
</table>

Reference: