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] = floorl(prod / diff);
                if( array[pos] > array[pos-1] && array[pos] * diff == prod )
                {
                  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);
                }
                return;
            }
            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 fname[256];
        sprintf(fname, "rec%ld.out", TOTAL);
        f = fopen(fname, "w");
        fprintf(f, "No of elements = %ld.\n", TOTAL);

        try(1, 0, 0);
        fflush(f);
        fclose(f);
        printf("Total %ld solutions found.\n", count);
        return 0;
}
```

Based on the above program the following table is formed.

Length	Number of Distinct Reciprocal Partitions
1	1
2	0
3	1
4	6
6	2320
7	245765

Reference:

[1] " Amarnath Murthy" , ' Smarandache Reciprocal Partition of Unity sets and sequences' , SNJ, Vol. 11, No. 1-2-3 , 2000.