d)

For simulation the sizes of requests for memory come from a random integer between 100 and k*1000(equally likely to be any integer in the range). The program is run with 4 different values of k. The life time of each block is an integer equally likely to be any integer between 1 and 100. The total available memory is 2^28 .

Two sets of runs are performed . In the first set of runs the deallocation of blocks is stopped after 2000 iterations and when external fragmentation occurs, the ordered pair (specified in the assignment statement) printed out is being recorded. In the second set of runs the deallocation is performed throughout and the program runs to completion.

Block Size	External Fragmentation	Internal Fragmentation	No of splits
100 – 2000	(1345,78.15%)	17.3%	262074
100 – 3000	(2915,88.12%)	19.4%	176597
100- 4000	(2934,78.2%)	17.4%	132786
100-5000	(2173,84.3%)	22.6%	104196

Block Size	Internal Fragmentation	No of splits
100 – 2000	17.4%	10807045
100 – 3000	19.3%	10833478
100 – 4000	17.4%	10299661
100 – 5000	22.7%	10171934

e) The polynomial arithmetic program has been tested with the three memory management methods
1. SP Heap Allocation
2. One bin Allocation
3. malloc() and free()

The program is tested with polynomial division problems of the form x^n -1 / (x -1) for values of n ranging from 10 to 50000. The time elapsed during the memory allocation and deallocation has been computed by timing every call to allocation and deallocation routines.

The time spent in memory allocation and deallocation in **microseconds(μs)** for the polynomial division is given below for the three allocation procedures.

Degree	**OneBin**	**SP_Heap**	**Malloc and free**
10	33	216	9
100	453	971	65
1000	25326	39034	496
5000	450130	931606	2352
10000	1789249	4023160	4725
15000	4006795	10442721	6500
20000	7098710	21460212	7325
25000	11079772	36405679	8233
35000	21739813	78076808	11261
50000	44723240	172878456	14363