Lab Exercise 3

On Memory Management and Allocation Strategies

Use the information that follows to complete this exercise.

JOB LIST

Job Stream #	Time	Job Size
1	5	5760
2	4	4190
3	8	3290
4	2	2030
5	2	2550
6	6	6990
7	8	8940
8	10	740
9	7	3930
10	6	6890
11	5	6580
12	8	3820
13	9	9140
14	10	420
15	10	220
16	7	7540
17	3	3210
18	1	1380
19	9	9850
20	3	3610
21	7	7540
22	2	2710
23	8	8390
24	5	5950
25	10	760

MEMORY LIST

Memory Block	Size
1	9500
2	7000
3	4500
4	8500
5	3000
6	9000
7	1000
8	5500
9	1500
10	500

At one large batch-processing computer installation, the management wants to decide what storage placement strategy will yield the best possible performance. The installation runs a large real storage computer under **fixed partition multiprogramming**. Each user program runs in a single group of **contiguous storage locations**. Users state their storage requirements and time units for CPU usage on their Job Control Card (it used to, and still does work this way, although cards are not used nowadays). The OS allocates to each user the appropriate partition and starts up the user's job. The job remains in memory until completion. A total of 50,000 memory locations are available, divided into fixed blocks as indicated in the table above.

a) Write an event-driven simulation to help you decide which storage placement strategy should be used at this installation. Your program would use the job stream and memory partitioning as indicated. Run the program until all jobs have been executed with the memory as is (in order by address). This will give you the **first-fit** type performance results.

b) Do the same as (a), but this time implement the **worst-fit** placement scheme.

c) Sort the memory partitions by size and run the program a second time; this will give you the **best-fit** performance results. For all parts (a), (b) and (c) you are investigating the performance of the system using a typical job stream by measuring:
   1. Throughput (how many jobs are processed per given time unit)
   2. Storage utilization (percentage of partitions never used, percentage of partitions heavily used, etc.)
   3. Waiting queue length
   4. Waiting time in queue
   5. Internal fragmentation

d) Look at the results from the first-fit, worst-fit and best-fit. Explain what the results indicate about the performance of the system for this job mix and memory organization. Is one method of partitioning better than the other? Why or why not? Could you recommend one method over the other based on your sample run? Would this hold in all cases? Write some conclusions and recommendations.