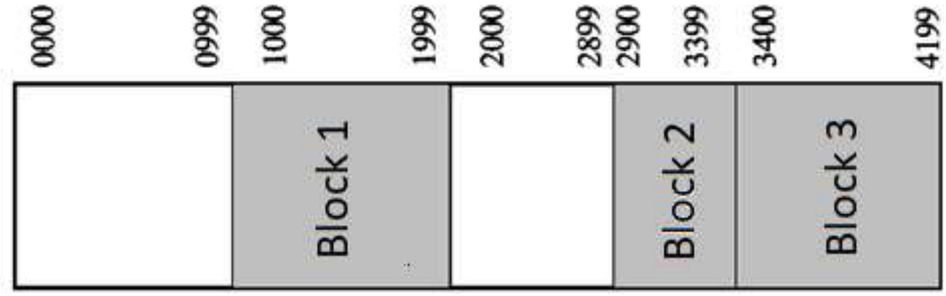
CONTIGUOUS ALLOCATION (USING BEST-FIT ALGORITHM)

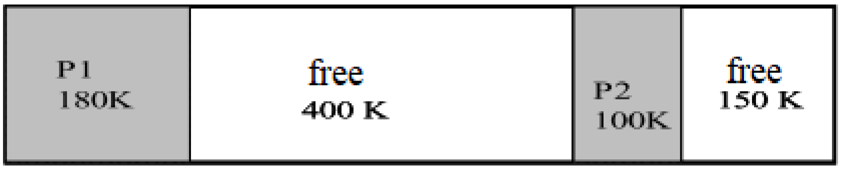
1. The following exercise will be based on a system with fixed partitioning and a memory with 4200 blocks. The memory has been divided into five partitions of 1000, 1000, 900, 500 and 800 blocks as you can see in the picture below. There are currently three processes.

Initial Address	Length
1000	1000
2900	500
3400	800



   a. From this moment on, we will need to load processes of 500 and 200 data blocks respectively. Draw the memory content after loading these two blocks.
   b. Is there internal fragmentation?
   If so, explain where it is and calculate the size.
   c. The processes of blocks 2 and 5 have finished. Now, we will try to load a process of 1200 blocks. Is there internal fragmentation? Explain why or why not

2. The following exercise will be based on a system with variable partitioning and a memory. Below, you can see the current state of the memory.



We have the following jobs waiting to be executed: P4 (120K), P5(200K) y P6(90K).

   a. Suppose none of the current processes finish. Draw the memory content after trying to load these three processes.
   How many free space remains?
   Is there internal fragmentation?
   Explain why or why not.

   b. Now, we want to execute P7 (100K). Draw the memory content with the new process.
   How many free space remains?
   Is there external fragmentation?
   Explain why or why not.

c. Suppose we compact all the free partitions. Draw the memory with all the processes that the system has loaded.

3. We have a 512 byte memory managed using paging with 16 byte pages (32 pages in total). At the start, we will have the initial state as shown below.

Process	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Size (bytes)	30	36	8	75	45	60	4	10	25	10
Pages	1,4	2,3,10	5	7,12-15	17,19,20	23-26	22	16	28,30	32

a) Draw the state of the memory as it was in its initial stage.

b) Calculate the total fragmentation in the initial stage.