4. Write a C program to implement the first-fit algorithm for memory management.

Test Case:

Memory partitions: 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order) Show the outcome for the test case with first-fit algorithms to place the processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)

#include <stdio.h>

int main() {

// Define the memory partitions

int partitions[] = {300, 600, 350, 200, 750, 125};

int num\_partitions = sizeof(partitions) / sizeof(partitions[0]);

// Define the processes

int processes[] = {115, 500, 358, 200, 375};

int num\_processes = sizeof(processes) / sizeof(processes[0]);

// Initialize an array to keep track of which partitions are used

int used[num\_partitions];

for (int i = 0; i < num\_partitions; i++) {

used[i] = 0;

}

// Loop through the processes

for (int i = 0; i < num\_processes; i++) {

int found = 0;

// Loop through the partitions to find the first one that fits

for (int j = 0; j < num\_partitions; j++) {

if (!used[j] && partitions[j] >= processes[i]) {

printf("Process %d allocated to partition %d (size %d)\n", i+1, j+1, partitions[j]);

used[j] = 1;

found = 1;

break;

}

}

if (!found) {

printf("Process %d could not be allocated.\n", i+1);

}

}

return 0;

}