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

Test Case:

Memory partitions: 40 KB, 10 KB, 30 KB, 60 KB, (in order) Show the outcome for the test case with first-fit algorithms to place the processes of size 100 KB.50 KB.30 KB ,120 KB,35 KB (in order)

Program:-

#include <stdio.h>

#define MAX\_PARTITIONS 5

#define MAX\_PROCESSES 5

void firstFit(int partitions[], int numPartitions, int processes[], int numProcesses) {

int allocation[MAX\_PROCESSES] = {0};

for (int i = 0; i < numProcesses; i++) {

for (int j = 0; j < numPartitions; j++) {

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

allocation[i] = j;

partitions[j] -= processes[i];

break;

}

}

}

printf("Process\tSize\tPartition\n");

for (int i = 0; i < numProcesses; i++) {

printf("%d\t%d KB\t", i + 1, processes[i]);

if (allocation[i] != 0)

printf("%d\n", allocation[i] + 1);

else

printf("Not Allocated\n");

}

}

int main() {

int partitions[MAX\_PARTITIONS] = {40, 10, 30, 60};

int processes[MAX\_PROCESSES] = {100, 50, 30, 120, 35};

int numPartitions = sizeof(partitions) / sizeof(partitions[0]);

int numProcesses = sizeof(processes) / sizeof(processes[0]);

printf("Memory Partitions (in KB): ");

for (int i = 0; i < numPartitions; i++) {

printf("%d ", partitions[i]);

}

printf("\n\n");

printf("Processes (in KB): ");

for (int i = 0; i < numProcesses; i++) {

printf("%d ", processes[i]);

}

printf("\n\n");

firstFit(partitions, numPartitions, processes, numProcesses);

return 0;

}

Output:-

