12. Write a C program to implement the best-fit algorithm and allocate the memory block to each process.

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 the best-fit algorithms to place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)

Program:

#include <stdio.h>

#define MAX\_PARTITIONS 6

#define MAX\_PROCESSES 5

int partitions[MAX\_PARTITIONS] = {300, 600, 350, 200, 750, 125}; int processes[MAX\_PROCESSES] = {115, 500, 358, 200, 375}; int allocation[MAX\_PROCESSES];

void best\_fit()

{

int i, j;

int best\_index; for (i = 0; i < MAX\_PROCESSES; i++) { best\_index = -1;

for (j = 0; j < MAX\_PARTITIONS; j++) { if (partitions[j] >= processes[i]) { if (best\_index == -1) { best\_index = j;

} else if (partitions[j] < partitions[best\_index]) { best\_index = j;

}

}

}

if (best\_index != -1) { allocation[i] = best\_index; partitions[best\_index] -= processes[i];

} else { allocation[i] = -1;

}

}

}

void print\_allocation()

{

int i;

printf("\nProcess No.\tProcess Size\tPartition No.\n"); for (i = 0; i < MAX\_PROCESSES; i++) { printf("%d\t\t%d\t\t", i+1, processes[i]); if (allocation[i] != -1) { printf("%d\n", allocation[i]+1);

} else {

printf("Not Allocated\n");

}

}

}

int main() { best\_fit(); print\_allocation(); return 0; }

Output:

