20. Write a C program to implement the worst-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 worst-fit algorithms to place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)

Program:

#include<stdio.h> #include<stdlib.h> int memory\_size = 6; int process\_size = 5; int memory[6] = {300, 600, 350, 200, 750, 125}; int process[5] = {115, 500, 358, 200, 375}; int allocated[5] = {0}; void worst\_fit\_algorithm() { int i, j, max\_index; for(i = 0; i < process\_size; i++) { max\_index = -1; for(j = 0; j < memory\_size; j++) { if(memory[j] >= process[i]) { if(max\_index == -1) { max\_index = j;

}

else if(memory[j] > memory[max\_index]) {

max\_index = j;

}

}

}

if(max\_index != -1) {

allocated[i] = 1;

memory[max\_index] -= process[i];

}

} } void print\_memory() {

int i;

printf("Memory allocation:\n"); for(i = 0; i < memory\_size; i++) { printf("%d KB ", memory[i]);

} printf("\n"); } void print\_process() {

int i;

printf("Process allocation:\n"); for(i = 0; i < process\_size; i++) { if(allocated[i]) {

printf("%d KB - Allocated\n", process[i]);

}

else {

printf("%d KB - Not allocated\n", process[i]);

} } printf("\n"); } int main() { worst\_fit\_algorithm(); print\_memory(); print\_process(); }

Output:

