WORST FIT

PROGRAM:

#include <stdio.h>

#define MAX\_PARTITIONS 10

#define MAX\_PROCESSES 10

int partitions[MAX\_PARTITIONS];

int processes[MAX\_PROCESSES];

void worstFit(int n, int m) {

int allocation[m];

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

allocation[i] = -1;

}

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

int worstIndex = -1;

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

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

if (worstIndex == -1 || partitions[j] > partitions[worstIndex]) {

worstIndex = j;

}

}

}

if (worstIndex != -1) {

allocation[i] = worstIndex;

partitions[worstIndex] -= processes[i];

}

}

printf("\nProcess No.\tProcess Size\tPartition No.\n");

for (int i = 0; i < m; 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() {

int n = 4;

int m = 5;

partitions[0] = 40;

partitions[1] = 10;

partitions[2] = 30;

partitions[3] = 60;

processes[0] = 100;

processes[1] = 50;

processes[2] = 30;

processes[3] = 120;

processes[4] = 35;

worstFit(n, m);

return 0;

}

OUTPUT:

