20. WORST FIT

#include <stdio.h>

#include <stdlib.h>

#define MAX\_PARTITIONS 6

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

int used[MAX\_PARTITIONS] = {0};

void worst\_fit(int process\_size) {

int i, j, index = -1;

for (i = 0; i < MAX\_PARTITIONS; i++) {

if (!used[i] && partitions[i] >= process\_size) {

if (index == -1 || partitions[i] > partitions[index]) {

index = i;

}

}

}

if (index != -1) {

used[index] = 1;

printf("Allocated %d KB to process of size %d KB\n", partitions[index], process\_size);

} else {

printf("Could not allocate memory to process of size %d KB\n", process\_size);

}

}

int main() {

int processes[5] = {115, 500, 358, 200, 375};

int i;

printf("Initial memory partitions:\n");

for (i = 0; i < MAX\_PARTITIONS; i++) {

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

}

printf("\n");

for (i = 0; i < 5; i++) {

worst\_fit(processes[i]);

}

printf("Final memory partitions:\n");

for (i = 0; i < MAX\_PARTITIONS; i++) {

printf("%d KB ", partitions[i] - used[i] \* processes[i]);

}

printf("\n");

return 0;

}

OUTPUT

