MALNAD COLLEGE OF ENGINEERING

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)
Hassan-573202, Karnataka, India

OPERATING SYSTEM(21CS502)

Activity - 02

Submitted by :

NAMRATHA JAY	4MC21CS101
NANDAN C V	4MC21CS102
NANDINI C N	4MC21CS103
NIHARI H R	4MC21CS104

Under the guidance of

Mrs.Chandana H M

Asst.Professor

Department of Computer Science and Engineering
Malnad College of Engineering
Hassan-573202, Karnataka, India

Write a C program for implementation memory allocation methods for fixed partition using best fit.

```
#include <stdio.h>
#include <limits.h>

#define MAX_PARTITIONS 10

struct Partition {
    int id;
    int size;
    int allocated;
};

void bestFit(struct Partition partitions[], int numPartitions, int processSize) {
    int bestFitIndex = -1;
    int minSize = INT_MAX;


    for (int i = 0; i < numPartitions; i++) {
        if (!partitions[i].allocated && partitions[i].size >= processSize) {
            if (partitions[i].size < minSize) {
                minSize = partitions[i].size;
                bestFitIndex = i;
            }
        }
    }

    if (bestFitIndex != -1) {
```

```
        printf("Process of size %d allocated to Partition %d\n", processSize,
partitions[bestFitIndex].id);
        partitions[bestFitIndex].allocated = 1;
    }
else {
        printf("Unable to allocate process of size %d\n", processSize);
    }
}
int main() {

    struct Partition partitions[MAX_PARTITIONS] = {
        {1, 100, 0},
        {2, 200, 0},
        {3, 50, 0},
        {4, 300, 0},
        {5, 150, 0}
    };

    int numPartitions = 5;

    int processSizes[] = {30, 100, 200, 80, 120};

    int numProcesses = sizeof(processSizes) / sizeof(processSizes[0]);
    printf("Initial Partitions:\n");
    for (int i = 0; i < numPartitions; i++) {
        printf("Partition %d: %d\n", i+1, partitions[i].size);
    }
    printf("\nProcess to be allocated:\n");
```

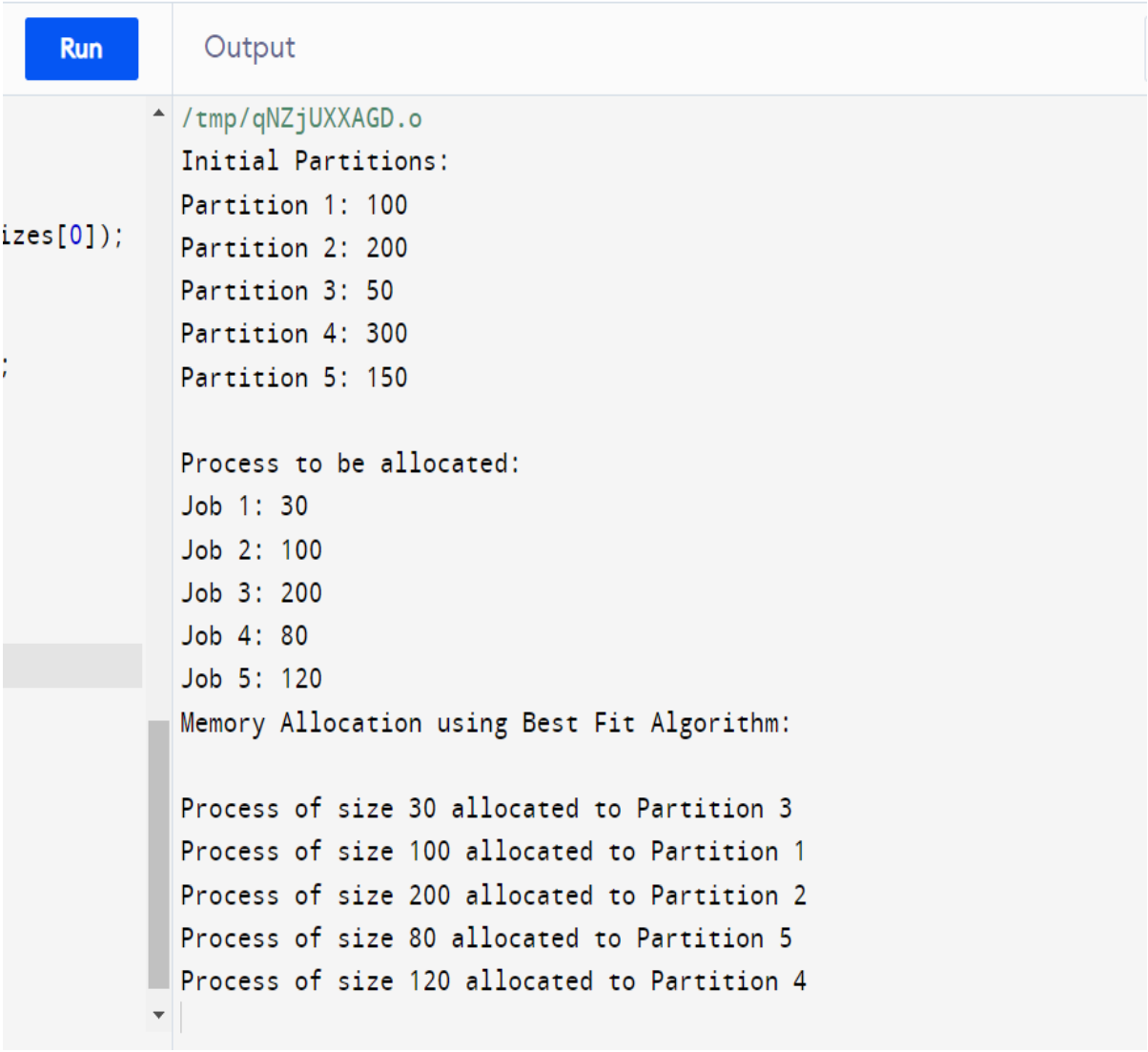
```
    for (int i = 0; i < numProcesses; i++) {
        printf("Job %d: %d\n", i+1, processSizes[i]);
    }


    printf("Memory Allocation using Best Fit Algorithm:\n");

    for (int i = 0; i < numProcesses; i++) {
        bestFit(partitions, numPartitions, processSizes[i]);
    }

    return 0;
}
```

Output:


Run
Output
izes[0]);
;
/tmp/qNZjUXXAGD.o
Initial Partitions:
Partition 1: 100
Partition 2: 200
Partition 3: 50
Partition 4: 300
Partition 5: 150
Process to be allocated:
Job 1: 30
Job 2: 100
Job 3: 200
Job 4: 80
Job 5: 120
Memory Allocation using Best Fit Algorithm:
Process of size 30 allocated to Partition 3
Process of size 100 allocated to Partition 1
Process of size 200 allocated to Partition 2
Process of size 80 allocated to Partition 5
Process of size 120 allocated to Partition 4