

# **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<stdlib.h>

// Structure to represent each memory block
struct MemoryBlock {
    int blockSize;
    int allocated;
};

// Function to allocate memory using Best Fit algorithm
void bestFit(int blockSize[], int m, int processSize[], int n) {
    // Allocate memory blocks to processes
    int allocation[n];

    // Initialize all allocations as -1
    for (int i = 0; i < n; i++)
        allocation[i] = -1;

    // Iterate through each process and allocate memory
    for (int i = 0; i < n; i++) {
        // Find the best fit memory block for current process
        int bestIdx = -1;
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (bestIdx == -1)
```

```

        bestIdx = j;
    else if (blockSize[j] < blockSize[bestIdx])
        bestIdx = j;
    }
}

// Allocate memory block to current process
if (bestIdx != -1) {
    allocation[i] = bestIdx;
    blockSize[bestIdx] -= processSize[i];
}
}

// Display allocation results
printf("\nProcess No.\tProcess Size\tBlock no.\n");
for (int i = 0; i < n; i++) {
    printf(" %d \t\t %d \t\t ", i + 1, processSize[i]);
    if (allocation[i] != -1)
        printf("%d", allocation[i] + 1);
    else
        printf("Not Allocated");
    printf("\n");
}
}

// Main function
int main() {
    int blockSize[] = { 100, 500, 200, 300, 600};

```

```

int processSize[] = {212, 417, 112, 426};

int m = sizeof(blockSize) / sizeof(blockSize[0]);

int n = sizeof(processSize) / sizeof(processSize[0]);

bestFit(blockSize, m, processSize, n);

return 0;
}

```

## Output:

The screenshot shows the Programiz C Online Compiler interface. The code in the editor is as follows:

```

main.c
43     if (allocation[i] != -1)
44         printf("%d", allocation[i] + 1);
45     else
46         printf("Not Allocated");
47     printf("\n");
48 }
49 }
50
51 // Main function
52 int main() {
53     int blockSize[] = {100, 500, 200, 300, 600};
54     int processSize[] = {212, 417, 112, 426};
55     int m = sizeof(blockSize) / sizeof(blockSize[0]);
56     int n = sizeof(processSize) / sizeof(processSize[0]);
57
58     bestFit(blockSize, m, processSize, n);
59
60     return 0;
61 }
62

```

The output window displays the following table:

Process No.	Process Size	Block no.
1	212	4
2	417	2
3	112	3
4	426	5

