OPERATING SYSTEM - CS23431

EXP 10(B)

FIRST FIT

NAME: S.KUMARAN **ROLL NO: 230701159**

PROGRAM:

```
#include <stdio.h>

int main() {
    int n1;
    printf("Enter number of memory blocks: ");
    scanf("%d", &n1);

    int mem[n1];
    printf("Enter values of memory blocks: ");
    for (int i = 0; i < n1; i++) {
        scanf("%d", &mem[i]);
    }

    int n2;
    printf("Enter number of process blocks: ");
    scanf("%d", &n2);

    int p[n2];
    printf("Enter values of process blocks: ");
    for (int i = 0; i < n2; i++) {
        scanf("%d", &p[i]);
```

```
}

int frag[n2], alloc[n2], emp[n1], allocsize[n2];

// Mark all memory blocks as empty
for (int i = 0; i < n1; i++) {
    emp[i] = 1;
}

// Initialize allocation to -1 (not allocated)
for (int i = 0; i < n2; i++) {
    alloc[i] = -1;
}

// First Fit Allocation
for (int i = 0; i < n2; i++) {
    for (int j = 0; j < n1; j++) {
        if (emp[j] && mem[j] >= p[i]) {
            alloc[i] = j;
            allocsize[i] = mem[j];
            frag[i] = mem[j] - p[i];
            emp[j] = 0;
            break;
        }
    }
}
```

```
    // Output
    printf("\nProcess\tSize\tBlock\tBlockSize\tFragment\n");
    for (int i = 0; i < n2; i++) {
        if (alloc[i] != -1) {
            printf("P%d\t%d\t%d\t%d\t\t%d\n", i + 1, p[i], alloc[i] + 1, allocsize[i], frag[i]);
        } else {
            printf("P%d\t%d\tNot Allocated\n", i + 1, p[i]);
        }
    }

    return 0;
}
```

OUTPUT:

```
Enter memory block size: 4
Enter value of memory blocks: 5
8
1
10
Enter process block size: 3
Enter values of process blocks: 1
4
7
FileNo  Filesize        BlockNo Blocksize       Fragment
0       1       0       5       4
1       4       1       8       4
2       7       3       10      3
```