

OS LAB MANUAL (CS23431)

Roll No:230701246

EX.NO:10(A)

BEST FIT

Aim: To implement Best Fit memory allocation technique using Python.

Program:

```
#include <stdio.h>

int main() {
    int m, n, i, j;
    printf("Enter number of memory blocks: ");
    scanf("%d", &m);
    int blockSize[m];
    printf("Enter sizes of %d memory blocks:\n", m);
    for (i = 0; i < m; i++) {
        scanf("%d", &blockSize[i]);
    }
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int processSize[n];
    printf("Enter sizes of %d processes:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &processSize[i]);
    }
    int allocation[n];
    for (i = 0; i < n; i++)
        allocation[i] = -1;
    for (i = 0; i < n; i++) {
        int bestIdx = -1;
        for (j = 0; j < m; j++) {
```

```

        if (blockSize[j] >= processSize[i]) {
            if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx])
                bestIdx = j;
        }
    }
    if (bestIdx != -1) {
        allocation[i] = bestIdx;
        blockSize[bestIdx] -= processSize[i];
    }
}
printf("\nProcess No.\tProcess Size\tBlock no.\n");
for (i = 0; i < n; i++) {
    printf(" %d\t\t%d\t\t", i + 1, processSize[i]);
    if (allocation[i] != -1)
        printf("%d\n", allocation[i] + 1);
    else
        printf("Not Allocated\n");
}
return 0;
}

```

Input:

```

praveen@LAPTOP-Q0D806DB:~$ vi best.c
praveen@LAPTOP-Q0D806DB:~$ gcc best.c
praveen@LAPTOP-Q0D806DB:~$ ./a.out
Enter number of memory blocks: 4
Enter sizes of 4 memory blocks:
20
12
30
45
Enter number of processes: 3
Enter sizes of 3 processes:
4
6
8

```

Output:

```
praveen@LAPTOP-Q0D806DB:~$ vi best.c
praveen@LAPTOP-Q0D806DB:~$ gcc best.c
praveen@LAPTOP-Q0D806DB:~$ ./a.out
Enter number of memory blocks: 4
Enter sizes of 4 memory blocks:
20
12
30
45
Enter number of processes: 3
Enter sizes of 3 processes:
4
6
8
```

Process No.	Process Size	Block no.
1	4	2
2	6	2
3	8	1