Ex. No.: 10a)
Date: 11-4-25

BEST FIT

Aim:

To implement Best Fit memory allocation technique using Python.

Algorithm:

1. Input memory blocks and processes with sizes
2. Initialize all memory blocks as free.
3. Start by picking each process and find the minimum block size that can be assigned to current process
4. If found then assign it to the current process.
5. If not found then leave that process and keep checking the further processes.

Program Code:

```
#include<stdio.h>
void bestFit(int blockSize[], int blockCount, int processSize[]
             int processCount) {
    int allocation[processCount];
    for(int i=0; i< processCount ; i++){
        allocation[i] = -1;
        for(int j= 0; j< blockCount; j++) {
            if (blockSize[j] >= processSize[i]){
                allocation[i] = j+1;
                blockSize[j] -= processSize[i];
                break;
            }
        }
    }
    printf("\n Process no. process size block no.\n");
    for (int i= 0; i< processCount ; i++) {
```

```
def best_fit(blocks, processes):
    n = len(blocks)
    m = len(processes)
    allocation = [-1]*m
    for i in range(m):
        best_idx = -1
        for j in range(n):
            if blocks[j] >= process[i]:
                if best_idx == -1 or blocks[j] < blocks
                                        [best_idx]:
                    best_idx = j
        if best_idx != -1
            allocation[i] = best_idx
            blocks[best_idx] -= process[i]
    print("Process No. process size    Block No.")
    for i in range(m):
        if allocation[i] != -1:
            print(i+1, "\t", processes[i], "\t",
                                allocation[i]+1)
        else:
            print(i+1, "\t", processes[i], "\t",
                "Not allocated")
b = int(input("Enter the number of memory
                                blocks:"))
p = int(input("Enter the number of processes:"))
blocks = [int(input(f"Enter size of blocks {i+1}: "))
                                for i in range(b)]
processes = [int(input(f"Enter size of process {i+1}:"))
                                for i in range(p)]
best_fit(blocks, processes)
```

Sample Output:

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

Enter the number of memory blocks: 5
Enter the number of processes: 4
Enter the size of block 1: 200
Enter the size of block 2: 100
Enter size of block 3: 500
Enter size of block 4: 150
Enter size of block 5: 300
Enter size of process 1: 80
Enter size of process 2: 190
Enter size of process 3: 400
Enter size of process 4: 325

process No.	Process size	Block No.
1	80	2
2	190	1
3	400	3
4	325	Not allocated

Result:

Hence the python program to implement Best fit memory allocation has been successfully executed.