Ex. No.: 10a)
Date: 09/04/25

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:

```
def bestfit ( bsize, m, psize, n):
    alloc = [-1] * n
    for i in range(n):
        bestidx = -1
        for j in range(m):
            if bsize[j] >= psize[i]:
                if bestidx == -1:
                    bestIdx = j
                elif bsize[bestIdx] > bsize[j]:
                    bestidx = j
        if bestIdx != -1:
            alloc[i] = bestidx
            bsize[bestidx] -= psize[i]
for i in range(n):
    print(i+1, "    ", psize[i], end = "    ")
    if (alloc[i] != -1):
        print(alloc[i]+1)
    else:
        print("Not Allocated")
```

```
if __name__ == '__main__':
    bsize = [1000, 500, 200, 300, 600]
    psize = [212, 417, 312, 426]
    m = len(bsize)
    n = len(psize)
    bestfit(bsize, m, psize, n)
```

Output :-

Process No	Process Size	Block no
1	212	4
2	417	2
3	112	5
4	426	Not Allocated

Sample Output:

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

Result:

Hence the Best Fit for the given Processes is Implemented and Verified