Ex. No.: 10a)
Date: 9/4/25

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:

```
def bestfit (bsize ,m, psize ,n):
    alloc = [-1] * n
    for i in range (n) :
        best idx = -1
        for j in range(m) ;
            if b size [j] >= psize [i] :
                if best idx = = -1 :
                    bestidx = j
                elif b size [ bestidx] > bsize[j] :
                    best idx = j
    if best Idx ! = -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_":
        b size = [100, 500, 200, 300, 600]
        p size = [212, 419, 312, 426]
        m = Len (bsize)
        n = Len (psize)
        bestfit (bsize, m, psize, n)
```

Sample Output:

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

OUTPUT:

```
Process No        Process Size        Block no

   1                  212                4
   2                  419                2
   3                  312                5
   4                  426             Not Allocated
```

Result:

Hence the Best fit for the given processes is Implemented and verified