**VAISHANVI E 231901059**

**Ex. No.: 10a)**

**Date:07-04-20205**

**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 best\_fit(block\_size, process\_size): n = len(block\_size) m = len(process\_size) allocation = [-1] \* m

for i in range(m): best\_idx = -1 for j in range(n): if block\_size[j] >= process\_size[i]: if best\_idx == -1 or block\_size[j] < block\_size[best\_idx]:

best\_idx = j if best\_idx != -1:

allocation[i] = best\_idx + 1 # 1-based indexing for block number block\_size[best\_idx] -= process\_size[i]

# Output

print("\nProcess No.\tProcess Size\tBlock No.") for i in range(m):

print(f"{i + 1}\t\t{process\_size[i]}\t\t", end='') if allocation[i] != -1: print(f"{allocation[i]}") else:

print("Not Allocated")

**PRIYANGA M 231901037**

# Input from user block\_size = [] process\_size = []

nb = int(input("Enter number of memory blocks: ")) for i in range(nb):

size = int(input(f"Enter size of block {i + 1}: ")) block\_size.append(size)

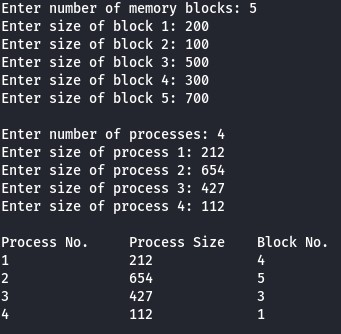
np = int(input("\nEnter number of processes: ")) for i in range(np):

size = int(input(f"Enter size of process {i + 1}: ")) process\_size.append(size)

# Call the function

best\_fit(block\_size, process\_size)

**OUTPUT:**



**RESULT:**

Hence, Best Fit memory allocation technique using Python has been implemented.