NAME: ARUN MC

ROLL NO: 230701036

Exp:10a **BEST FIT**

Aim:

To implement Best Fit memory allocation technique using Python.

CODE:

```
def best_fit(memory_blocks, processes):
    allocation = [-1] * len(processes)  # Initialize allocation for each process

    for i in range(len(processes)):
        best_index = -1
        for j in range(len(memory_blocks)):
            if memory_blocks[j] >= processes[i]:
                if best_index == -1 or memory_blocks[j] < memory_blocks[best_index]:
                    best_index = j

        # If a suitable block is found
        if best_index != -1:
            allocation[i] = best_index + 1  # Allocate block (using 1-based index)
            memory_blocks[best_index] -= processes[i]  # Update available size of block

    # Output the allocation result
    print("Process No.\tProcess Size\tBlock No.")
    for i in range(len(processes)):
        print(f"{i + 1}\t\t{processes[i]}\t\t{allocation[i] if allocation[i] != -1 else 'Not Allocated'}")


# Input: Memory blocks and process sizes
memory_blocks = [100, 500, 200, 300, 600]
processes = [212, 417, 112, 426]

best_fit(memory_blocks, processes)
~
~
~
~
```

OUTPUT:

```
[cse36@localhost ~]$ vi 10a_bestfit.py
:[cse36@localhost ~]$ python3 10a_bestfit.py
Process No.     Process Size    Block No.
1               212             4
2               417             2
3               112             3
4               426             5
[cse36@localhost ~]$
```