**Experiment No: 03**

**Aim:** **To implement Best Fit and First Fit Memory Allocation policy.**

**Code:**

**First Fit**

def FirstFit(block\_Size, blocks, process\_Size, proccesses):

    allocate = [-1] \* proccesses

    occupied = [False] \* blocks

    for i in range(proccesses):

        for j in range(blocks):

            if not occupied[j] and (block\_Size[j] >= process\_Size[i]):

                allocate[i] = j

                occupied[j] = True

                break

    print("Block sizes are:")

    for i in range(len(block\_Size)):

        print(block\_Size[i],end='\t')

    print("\nProcess No. \t\tProcess Size \t\tBlock No.")

    for i in range(proccesses):

        print(str(i + 1) + "\t\t\t" + str(process\_Size[i]) + "\t\t\t", end=" ")

        if allocate[i] != -1:

            print(allocate[i] + 1)

        else:

            print("Not Allocated")

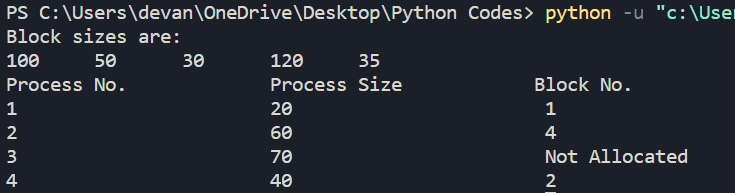
block\_Size = [100, 50, 30, 120, 35]

process\_Size = [20, 60, 70, 40]

m = len(block\_Size)

n = len(process\_Size)

FirstFit(block\_Size, m, process\_Size, n)



**Best Fit**

def bestFit(blockSize, m, processSize, n):

    allocation = [-1] \* n

    for i in range(n):

        bestIdx = -1

        for j in range(m):

            if blockSize[j] >= processSize[i]:

                if bestIdx == -1:

                    bestIdx = j

                elif blockSize[bestIdx] > blockSize[j]:

                    bestIdx = j

        if bestIdx != -1:

            allocation[i] = bestIdx

            blockSize[bestIdx] -= processSize[i]

    print("Process No. Process Size  Block no.")

    for i in range(n):

        print(i + 1, "       ", processSize[i],

                                end = "      ")

        if allocation[i] != -1:

            print(allocation[i] + 1)

        else:

            print("Not Allocated")

if \_\_name\_\_ == '\_\_main\_\_':

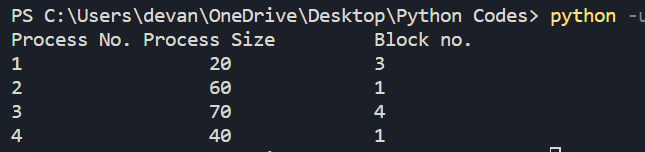
    blockSize = [100, 50, 30, 120, 35]

    processSize = [20,60,70,40]

    m = len(blockSize)

    n = len(processSize)

    bestFit(blockSize, m, processSize, n)

****