NAME: ALISTAIR SALDANHA

SAPID: 60009200024

BATCH: K1

EXPERIMENT-6

In [25]:

```
def FirstFit(frag_size, proc_size):
 # Initially allocate the memory with -1
 occupied_block = [-1] * len(proc_size)
 # Loop through each process size
 for i in range(len(proc_size)):
  # Loop through each block size
  for j in range(len(frag_size)):
   # The first space when frag_size > proc_size is encountered, occupy the space and brea
k
   if frag_size[j] >= proc_size[i]:
    occupied_block[i] = j
    frag_size[j] -= proc_size[i]
    break
 # Print the table
 print("Process No\tProcess Size\tBlock no.")
 for i in range(len(proc_size)):
  print(i + 1,"\t\t", proc_size[i],end = "\t\t  ")
  # If the process is allocated print the block number
  if occupied_block[i] != -1:
   print(occupied_block[i] + 1)
  # else print Not Allocated
  else:
   print("Not Allocated")
```

In [26]:

```
def WorstFit(frag_size, proc_size):
 # Initially allocate the memory with -1
 occupied_block = [-1] * len(proc_size)
 # Loop through each process size
 for i in range(len(proc_size)):
  # Initialise worst Index to -1 and loop through block size
  wstIdx = -1
  for j in range(len(frag_size)):
   # if frag_size >= proc_size and
   if frag_size[j] >= proc_size[i]:
    # if worstIndex = -1 or frag_size[worstIndex] < frag_size[curr] then worstIndex=curr
    if ((wstIdx == -1) or (frag_size[wstIdx] < frag_size[j])):
     wstIdx = j
  if wstIdx != -1:
   occupied_block[i] = wstIdx
   frag_size[wstIdx] -= proc_size[i]
 # Print the table
 print("Process No\tProcess Size\tBlock no.")
 for i in range(len(proc_size)):
  print(i + 1,"\t\t", proc_size[i],end = "\t\t  ")
  # If the process is allocated print the block number
  if occupied_block[i] != -1:
   print(occupied_block[i] + 1)
  # else print Not Allocated
  else:
   print("Not Allocated")
```

In [27]:

```
def BestFit(frag_size, proc_size):
 occupied_block = [-1] * len(proc_size)
 for i in range(len(proc_size)):
  bestIdx = -1
  for j in range(len(frag_size)):
   if frag_size[j] >= proc_size[i]:
    if bestIdx == -1:
     bestIdx = j
    elif frag_size[bestIdx] > frag_size[j]:
     bestIdx = j
  if bestIdx != -1:
   occupied_block[i] = bestIdx
   frag_size[bestIdx] -= proc_size[i]
 # Print the table
 print("Process No\tProcess Size\tBlock no.")
 for i in range(len(proc_size)):
  print(i + 1,"\t\t", proc_size[i],end = "\t\t  ")
  # If the process is allocated print the block number
  if occupied_block[i] != -1:
   print(occupied_block[i] + 1)
  # else print Not Allocated
  else:
   print("Not Allocated")
```

In [28]:

```
for _ in range(1,4):
    frag_size = [100, 500, 200, 300, 600]
    proc_size = [212, 417, 112, 426]
    choice = int(input("Enter 1 for First Fit, 2 for Best Fit, 3 for Worst Fit: "))
    if(choice==1):
        print("----------FIRSTFIT----------")
        FirstFit(frag_size, proc_size)
    elif(choice==2):
        print("----------BESTFIT----------")
        BestFit(frag_size, proc_size)
    elif(choice==3):
        print("----------WORSTFIT----------")
        WorstFit(frag_size, proc_size)
```

```
----------FIRSTFIT----------
Process No Process Size Block no.
1    212    2
2    417    5
3    112    2
4    426    Not Allocated
----------BESTFIT----------
Process No Process Size Block no.
1    212    4
2    417    2
3    112    3
4    426    5
----------WORSTFIT----------
Process No Process Size Block no.
1    212    5
2    417    2
3    112    5
4    426    Not Allocated
```