Ex. No.: 10a)
Date: 11/4/25

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:

```
n=int(input("Enter no. of block"))
bs=[]
for i in range(n):
        n1=int(input("Enter the block size"))
        bs.append(n1)

bs1=bs.copy()
bs1.sort()
np=int(input("Enter no. of process"))
ps=[]
for i in range(np):
        n2=int(input("Enter process size"))
        ps.append(n2)

d={}
for i in range(np):
        c=0
        for j in range(len(bs1)):
                if(ps[i]<=bs1[j]):
                        c+=1
```


PDF Scanner ACE Scanner

Enter no. of block 5
Enter the block size 500
Enter the block size 200
Enter the block size 300
Enter the block size 600
Enter the block size 100

Enter no. of process 5
Enter process size 300
Enter process size 250
Enter process size 477
Enter process size 191
Enter process size 91

PROCESS No.	PROCESS Size	Block No.
0	300	2
1	250	0
2	477	3
3	191	1
4	91	4

PDF Scanner ACE Scanner

```
        d[i]=[ps[i], bs.index(bsl[j]), bsl[j]]
        bsl.remove(bsl[j])
        break
    if(c==0):
        d[i]=[ps[i], "can't fit"]
print("\t\t Process No. \t\t\t Process size \t\t\t Block No.")
for i in d:
    print("\t\t ",i," \t\t\t\t\t\t\t\t",d[i][0],"\t\t\t\t\t\t ",d[i][1])
```


PDF Scanner ACE Scanner

Sample Output:

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

Result:

Thus the program to implement solution Best Fit memory allocation technique using Python has been executed successfully.

PDF Scanner ACE Scanner

Ex. No.: 10b)
Date: 11/4/25

FIRST FIT

Aim:

To write a C program for implementation memory allocation methods for fixed partition using first fit.

Algorithm:

1. Define the max as 25.
2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max]. 3: Get the number of blocks,files,size of the blocks using for loop.
4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]
5: Check highest

Program Code:

```
n=int(input("Enter no. of block"))
bs=[]
for i in range(n):
        n1=int(input("Enter the block size"))
        bs.append(n1)
bs1=bs.copy()
np=int(input("Enter no. of process"))
ps=[]
for i in range(np):
        n2=int(input("Enter process size"))
        ps.append(n2)
d={}
for i in range(np):
        c=0
        for j in range(len(bs1)):
                if(ps[i]<bs1[j]):
                        Ct=1
```


PDF Scanner ACE Scanner

Enter no. of blocks 5
Enter the block size 500
Enter the block size 200
Enter the block size 300
Enter the block size 600
Enter the block size 100
Enter no. of process 5
Enter process size 300
Enter process size 250
Enter process size 477
Enter process size 191
Enter process size 91

Process No.	Process size	Block No.
0	300	0
1	250	2
2	477	3
3	191	1
4	91	4

PDF Scanner ACE Scanner

```
        d[i] = [ps[i], bs.index(bs1[j]), bs1[j]]
        bs1.remove(bs1[j])
        break
    if(c==0)
        d[i] = [ps[i], "can't fit"]
print("\t\tProcess No.\t\t\t Process Size \t\t\t Block No.")
for i in d:
    print("\t\t ", i, "\t\t\t\t\t\t\t ", d[i][0], "\t\t\t\t\t\t ", d[i][1])
```


ACE Scanner

Sample Output:

```
Enter the number of blocks:4
Enter the number of files:3

Enter the size of the blocks:-
Block 1:5
Block 2:8
Block 3:4
Block 4:10
Enter the size of the files:-
File 1:1
File 2:4
File 3:7

File_no:        File_size :     Block_no:       Block_size:     Fragment
1               1               1               5               4
2               4               2               8               4
3               7               4               10              3
```

Result:

Thus the program to implement solution First Fit memory allocation technique using Python, has been executed succesfully.

ACE Scanner