

Operating System – CS23431

Ex 10a)	Best Fit
Name: B M Madhumitha	
Reg No: 230701168	

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 number of blocks:"))
bsize= {}
fill=[]
print("Enter Block Size:")

for i in range (0,n):
    bsize[i+1] = int(input(f'B {i+1} :'))
    fill.append(0)

p = int(input("Enter number of processes:"))
psize=[]
pblock=[]
for i in range (0,p):
    psize.append(int(input(f'P {i} :')))
    pblock.append(0)
bsize = dict(sorted(bsize.items(),key = lambda item:item[1]))

for i in range(0,p):
    for j,(key,val) in enumerate(bsize.items()):
        if psize[i] <= val and fill[j] == 0:
            fill[j] = 1
            pblock[i] = key
            break

print(f'{"Process No.':^12} {"Process size':^12} {"Block Size':^12} {"Block_id':^12}")
print("_____")
for i in range(0,p):

    if(pblock[i]!=0):
        print(f'{i+1:^12} {psize[i]:^12},{bsize[pblock[i]]:^12} {pblock[i]:^12}')
    else:
```

```
print(f'{i+1:^12} {psize[i]:^12} {NILL':^12} {'Not Allocated':^13}")
```

Output:

```
C:\Users\kambm\OneDrive\Desktop\Madhumitha\sem IV\OS Assignment\Final version>
py bestfit_FINAL.py
Enter number of blocks:4
Enter Block Size:
B1:870
B2:436
B3:256
B4:236
Enter number of processes:3
P0:45
P1:778
P2:34
Process No.   Process size  Block Size   Block_id
-----
      1         45      ,    236         4
      2        778      ,    870         1
      3         34      ,    256         3
C:\Users\kambm\OneDrive\Desktop\Madhumitha\sem IV\OS Assignment\Final version>
```

Result: Thus, the program was executed successfully.