

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:

```
#include <stdio.h>
int main() {
    int n, m;
    printf("Enter the no of block");
    scanf("%d", &n);
    printf("Enter no of processes");
    scanf("%d", &m);
    int block[n];
    int process[m];
    int allocation[m];
    for (int i=0; i<m; i++) {
        allocation[i] = -1;
    }
    for (int i=0; i<n; i++) {
        scanf("%d", &block[i]);
    }
    for (int i=0; i<m; i++) {
        printf("Enter process %d size: ", i+1);
        scanf("%d", &process[i]);
    }
}
```



```

int best_index;
for (int i = 0; i < m; i++)
{
    best_index = -1;
    for (int j = 0; j < n; j++)
    {
        if (blocks[j] >= process[i])
        {
            if (blocks[j] < block[best_index])
            {
                best_index = j;
                allocation[i] = best_index;
                blocks[best_index] -= process[i];
            }
        }
    }
}

```

```

Printf("\n process no. process size block NO");
for (int i = 0; i < m; i++)
{

```

```

    if (allocation[i] != -1)
    {
        printf("\n i.d %d %d %d %d %d", i+1, process[i],
            allocation[i], blocks[allocation[i]], blocks[allocation[i]] - process[i]);
    }
    else
    {
        printf("\n i.d %d %d %d %d %d Not allocated",
            i+1, process[i], allocation[i], blocks[allocation[i]], blocks[allocation[i]] - process[i]);
    }
}

```



Output:

Enter no of block: 4

Block size:

B1 - 100

B2 - 500

B3 - 150

B4 - 300

Enter no of processes: 3

Process size:

P<sub>1</sub> - 99

P<sub>2</sub> - 211

P<sub>3</sub> - 300

Process NO	Process size	Block NO
P <sub>1</sub>	99	B <sub>1</sub>
P <sub>2</sub>	211	B <sub>4</sub>
P <sub>3</sub>	300	B <sub>2</sub>

Sample Output:

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

Result: Hence the best fit memory allocation technique was successfully executed

*Signature*