Ex. No.: 10a)
Date: 11.04.25

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 blocks[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", &blocks[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 %d \t\t %d \t\t %d", i+1,
                                process[i], allocation[i] +1)
     }
     else {
       printf ("\n %d \t\t %d \t\t Not allocated",
                                i+1, process[i]);
     }
  }
}
```

Enter no of block : 3

Block size :

250

100

150

Enter no of process: 3

100

200

300

Process no	Process Size	Block no
1	100	2
2	200	1
3	300	not allocated

Sample Output:

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

Result:

Thus the c program for best fit memory allocation has been successfully executed.