Ex. No.: 10a)
Date: 9/4/25

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 blocksize[10], processsize[10], blockallocated[10], allocation[10];
    int i,j, nb,np;
    printf("Enter number of memory blocks: ");
    scanf("%d",&nb);
    printf("Enter size of each memory block:\n");
    for (i=0; i<nb; i++){
        printf("Block %d", i+1);
        scanf(" %d", &blocksize[i]);
        blockAllocated[i]=0;
    }
    printf("Enter number of process: ");
    scanf(" %d", &np);
    printf("Enter size of each process:\n");
    for (i=0; i<np; i++){
        printf(" process %d", i+1);
```

```
            scanf("%d", &processsize[i]);
            allocation[i]=-1;
      }
      for(i=0; i<np; i++){
            int  bestIdx=-1;
            for(j=0; j<nb; j++){
                  if(!blockAllocated[j] && blocksize[j] >= processsize[i]){
                        if(bestIdx == -1 || blocksize[j] < blocksize[bestIdx]){
                              bestIdx=j;
                        }
                  }
            }
            if(bestIdx != -1){
                  allocation[i]=bestIdx;
                  blockAllocated[bestIdx]=1;
            }
      }
      printf("\n Process No \t Processsize \t Block No \n");
      for(i=0; i<np; i++){
            printf("%d\t\t %d\t\t %d \t\t", i+1, processsize[i]);
      }
}
```

Sample Output:

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

The remaining fragments of block:

```
90
15
13
5
20
```

Process	Process_size	Block_No	Fragment
P_1	20	3	13
P_2	30	2	15
P_3	50	5	20
P_4	40	4	5
P_5	10	1	90

Result:

Using C program the best fit memory allocation algorithm is implemented.

[illegible]