

Exp – 10a):

AIM: to implement best fit memory allocation technique using python

PROGRAM:

```
def best_fit(block_size, process_size):
    allocation = [-1] * len(process_size)

    for i in range(len(process_size)):
        best_idx = -1
        for j in range(len(block_size)):
            if block_size[j] >= process_size[i]:
                if best_idx == -1 or block_size[j] < block_size[best_idx]:
                    best_idx = j

        if best_idx != -1:
            allocation[i] = best_idx
            block_size[best_idx] -= process_size[i]

    print("\nProcess No.\tProcess Size\tBlock No.")
    for i in range(len(process_size)):
        print(f"{i+1}\t\t\t{process_size[i]}\t\t\t", end='')
        if allocation[i] != -1:
            print(f"{allocation[i] + 1}")
        else:
            print("Not Allocated")

# Sample Data
block_size = [100, 500, 200, 300, 600]
process_size = [212, 417, 112, 426]

best_fit(block_size, process_size)
```

OUTPUT:

```
\jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/Python programs$ python3 bestFit.py
Process No.      Process Size      Block No.
1                212              4
2                417              2
3                112              3
4                426              5
```

Exp – 10b)

AIM: to write a c program for implementation of memory allocation methods for fixed partition using first fit

PROGRAM:

```
#include <stdio.h>
#define max 25

int main() {
    int frag[max], b[max], f[max], i, j, nb, nf, temp;
    static int bf[max], ff[max];

    printf("Enter the number of blocks: ");
    scanf("%d", &nb);
    printf("Enter the number of files: ");
    scanf("%d", &nf);

    printf("Enter the size of the blocks:\n");
    for(i = 0; i < nb; i++) {
        printf("Block %d: ", i + 1);
        scanf("%d", &b[i]);
        bf[i] = 0;
    }

    printf("Enter the size of the files:\n");
    for(i = 0; i < nf; i++) {
        printf("File %d: ", i + 1);
        scanf("%d", &f[i]);
    }

    for(i = 0; i < nf; i++) {
        for(j = 0; j < nb; j++) {
            if(bf[j] == 0 && b[j] >= f[i]) {
                ff[i] = j;
                frag[i] = b[j] - f[i];
                bf[j] = 1;
                break;
            }
        }
    }

    printf("\nFile_no:\tFile_size:\tBlock_no:\tBlock_size:\tFragment\n");
    for(i = 0; i < nf; i++) {
        printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n",
            i + 1, f[i], ff[i] + 1, b[ff[i]], frag[i]);
    }

    return 0;
}
```

OUTPUT:

```
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ gcc FirstFit.c -o firstfit
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./firstfit
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
```