Ex. No.: 10b)

FIRST FIT

Aim:

To write a C program for implementation memory allocation methods for fixed partition using the first fit.

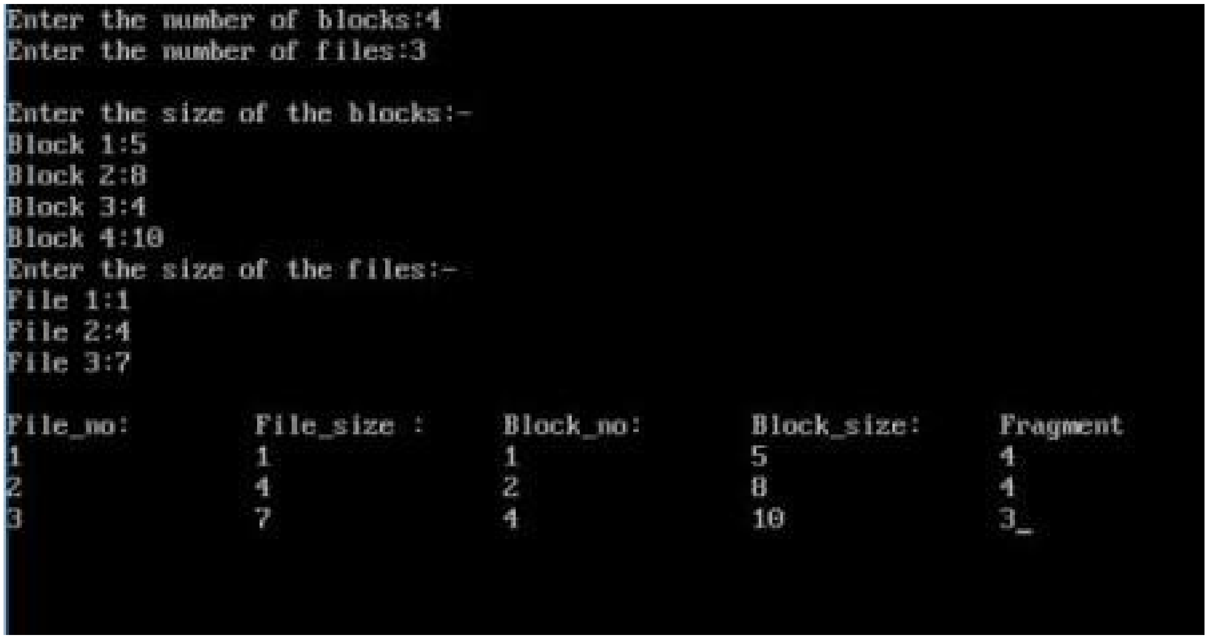
Algorithm:

1. Define the max as 25.
2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max].
3:
Get the number of blocks,files,size of the blocks using for loop.
4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]
5: Check highest

Program Code:

```
 #include <stdio.h>

#define MAX 10

int main() {
    int partitions[MAX], processes[MAX], allocation[MAX];
    int num_partitions, num_processes;

    printf("Enter number of memory partitions: ");
    scanf("%d", &num_partitions);
    printf("Enter sizes of partitions:\n");
    for (int i = 0; i < num_partitions; i++) {
        printf("Partition %d: ", i + 1);
        scanf("%d", &partitions[i]);
    }

    printf("Enter number of processes: ");
    scanf("%d", &num_processes);
    printf("Enter sizes of processes:\n");
    for (int i = 0; i < num_processes; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d", &processes[i]);
        allocation[i] = -1;
    }
    for (int i = 0; i < num_processes; i++) {
        for (int j = 0; j < num_partitions; j++) {
            if (partitions[j] >= processes[i]) {
                allocation[i] = j;
                partitions[j] -= processes[i];
                break;
            }
        }
    }

    printf("\nProcess No.\tProcess Size\tPartition No.\n");
    for (int i = 0; i < num_processes; i++) {
        printf("%d\t\t%d\t\t", i + 1, processes[i]);
        if (allocation[i] != -1)
            printf("%d\n", allocation[i] + 1);
        else
            printf("Not Allocated\n");
    }

    return 0;
}
```

Sample Output:



```
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_
```

Output:

```
Enter number of memory partitions: 3
Partition 1: 100
Partition 2: 500
Partition 3: 200

Enter number of processes: 4
Process 1: 212
Process 2: 417
Process 3: 112
Process 4: 426
```

Result:

First Fit memory allocation technique has been successfully implemented and the output has been verified.