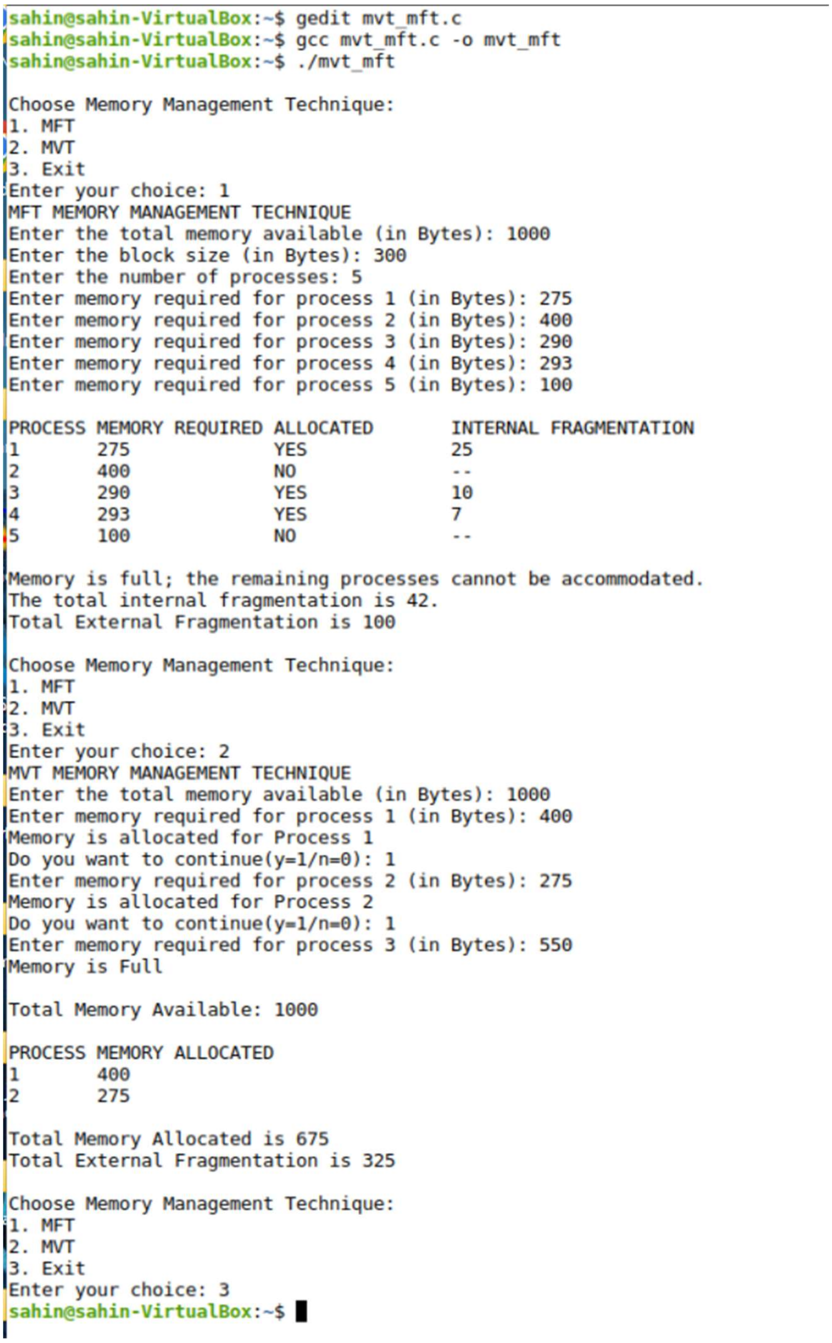
Name: Sahin Nayak **Enrollment Number : 12023006015086**

ASSIGNMENT-5

1. Write a C program to simulate the MVT and MFT memory management techniques.

```
#include <stdio.h>
void mft() {
    int total_memory, block_size, num_blocks, num_processes, i;
    int internal_fragmentation = 0, external_fragmentation = 0;
    int allocated_blocks = 0;
    printf("MFT MEMORY MANAGEMENT TECHNIQUE\n");
    printf("Enter the total memory available (in Bytes): ");
    scanf("%d", &total_memory);
    printf("Enter the block size (in Bytes): ");
    scanf("%d", &block_size);
    num_blocks = total_memory / block_size;
    printf("Enter the number of processes: ");
    scanf("%d", &num_processes);
    int memory_required[num_processes];
    int allocated[num_processes];
    for (i = 0; i < num_processes; i++) {
        printf("Enter memory required for process %d (in Bytes): ", i + 1);
        scanf("%d", &memory_required[i]);
        if (memory_required[i] <= block_size && allocated_blocks < num_blocks) {
            allocated[i] = 1;
            internal_fragmentation += (block_size - memory_required[i]);
            allocated_blocks++;
        } else { allocated[i] = 0; }}
    printf("\nPROCESS\tMEMORY REQUIRED\tALLOCATED\tINTERNAL FRAGMENTATION\n");
    for (i = 0; i < num_processes; i++) {
        printf("%d\t%d\t\t", i + 1, memory_required[i]);
        if (allocated[i]) {
            printf("YES\t\t%d\n", block_size - memory_required[i]);
        } else {printf("NO\t\t--\n");}}
    external_fragmentation = total_memory - (allocated_blocks * block_size);
    printf("\nMemory is full; the remaining processes cannot be accommodated.\n");
    printf("The total internal fragmentation is %d.\n", internal_fragmentation);
    printf("Total External Fragmentation is %d\n", external_fragmentation);
}
void mvt() {
    int i, total_memory, memory_allocated = 0, memory_required;
    int process_num = 0, choice;
    printf("MVT MEMORY MANAGEMENT TECHNIQUE\n");
    printf("Enter the total memory available (in Bytes): ");
    scanf("%d", &total_memory);
    int allocated_memory[100];
    while (1) {
        printf("Enter memory required for process %d (in Bytes): ", ++process_num);
        scanf("%d", &memory_required);
        if (memory_allocated + memory_required <= total_memory) {
            allocated_memory[process_num - 1] = memory_required;
            memory_allocated += memory_required;
            printf("Memory is allocated for Process %d\n", process_num);
        } else {
            printf("Memory is Full\n");
            process_num--;
            break;  }
        printf("Do you want to continue(y=1/n=0): ");
        scanf("%d", &choice);
        if (choice == 0) break; }
    printf("\nTotal Memory Available: %d\n", total_memory);
    printf("\nPROCESS\tMEMORY ALLOCATED\n");
    for (i = 0; i < process_num; i++) {
        printf("%d\t%d\n", i + 1, allocated_memory[i]);  }
    printf("\nTotal Memory Allocated is %d\n", memory_allocated);
    printf("Total External Fragmentation is %d\n", total_memory - memory_allocated); }
int main() {
    int choice;
    while (1) {
        printf("\nChoose Memory Management Technique:\n");
        printf("1. MFT\n2. MVT\n3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                mft();
                break;
            case 2:
                mvt();
                break;
            case 3:
                return 0;
            default:
                printf("Invalid choice! Please try again.\n"); } } return 0;}
```



```
sahin@sahin-VirtualBox:~$ gedit mvt_mft.c
sahin@sahin-VirtualBox:~$ gcc mvt_mft.c -o mvt_mft
sahin@sahin-VirtualBox:~$ ./mvt_mft

Choose Memory Management Technique:
1. MFT
2. MVT
3. Exit
Enter your choice: 1
MFT MEMORY MANAGEMENT TECHNIQUE
Enter the total memory available (in Bytes): 1000
Enter the block size (in Bytes): 300
Enter the number of processes: 5
Enter memory required for process 1 (in Bytes): 275
Enter memory required for process 2 (in Bytes): 400
Enter memory required for process 3 (in Bytes): 290
Enter memory required for process 4 (in Bytes): 293
Enter memory required for process 5 (in Bytes): 100

PROCESS MEMORY REQUIRED ALLOCATED       INTERNAL FRAGMENTATION
1       275             YES             25
2       400             NO              --
3       290             YES             10
4       293             YES             7
5       100             NO              --

Memory is full; the remaining processes cannot be accommodated.
The total internal fragmentation is 42.
Total External Fragmentation is 100

Choose Memory Management Technique:
1. MFT
2. MVT
3. Exit
Enter your choice: 2
MVT MEMORY MANAGEMENT TECHNIQUE
Enter the total memory available (in Bytes): 1000
Enter memory required for process 1 (in Bytes): 400
Memory is allocated for Process 1
Do you want to continue(y=1/n=0): 1
Enter memory required for process 2 (in Bytes): 275
Memory is allocated for Process 2
Do you want to continue(y=1/n=0): 1
Enter memory required for process 3 (in Bytes): 550
Memory is Full

Total Memory Available: 1000

PROCESS MEMORY ALLOCATED
1       400
2       275

Total Memory Allocated is 675
Total External Fragmentation is 325

Choose Memory Management Technique:
1. MFT
2. MVT
3. Exit
Enter your choice: 3
sahin@sahin-VirtualBox:~$ █
```