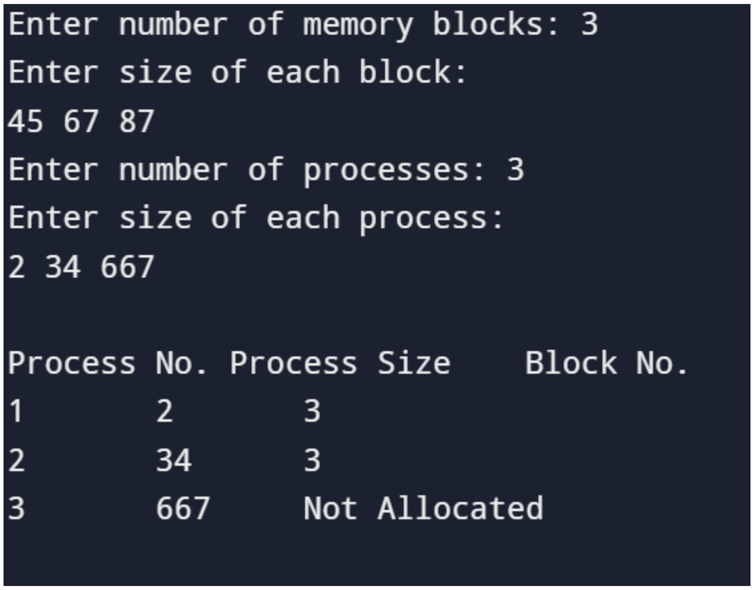
EXP 21: Develop a C program to implement worst fit algorithm of memory management.

```
#include <stdio.h>

#define MAX 25

void worstFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[MAX];

    // Initially no process is allocated
    for (int i = 0; i < n; i++)
        allocation[i] = -1;

    // Pick each process and find the worst fit block
    for (int i = 0; i < n; i++) {
        int worstIdx = -1;
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (worstIdx == -1 || blockSize[j] > blockSize[worstIdx])
                    worstIdx = j;
            }
        }

        // If a block was found
        if (worstIdx != -1) {
            allocation[i] = worstIdx;
            blockSize[worstIdx] -= processSize[i];
        }
    }

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

int main() {
    int blockSize[MAX], processSize[MAX];
    int m, n;

    printf("Enter number of memory blocks: ");
    scanf("%d", &m);
    printf("Enter size of each block:\n");
    for (int i = 0; i < m; i++)
        scanf("%d", &blockSize[i]);

    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("Enter size of each process:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &processSize[i]);
    worstFit(blockSize, m, processSize, n);


    return 0;
}
```

Sample Output



```
Enter number of memory blocks: 3
Enter size of each block:
45 67 87
Enter number of processes: 3
Enter size of each process:
2 34 667

Process No. Process Size    Block No.
1       2       3
2       34      3
3       667     Not Allocated
```