NAME: G. GANESH

REG NO: 192373008

EXERICSE-21

Develop a C program to implement the worst fit algorithm of memory management.

Aim:

To develop a C program to implement the Worst Fit memory allocation algorithm.

Algorithm:

1. **Initialize Memory Blocks and Processes:**
   - Create arrays for memory block sizes and process sizes.
2. **Find Worst Fit:**
   - For each process, find the memory block with the largest size that can accommodate it.
   - Allocate the process to this block and reduce the block size.
3. **Unallocated Processes:**
   - If no suitable block is found, mark the process as unallocated.
4. **Display Results:**
   - Show the allocation details for each process.

Procedure:

1. Input the sizes of memory blocks and processes.
2. For each process, check all memory blocks to find the largest suitable block.
3. Allocate the process to the block and adjust the block size.
4. Print the allocation details for each process and memory block.

Code:

```
#include <stdio.h>
void worstFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];
    for (int i = 0; i < n; i++) {
        allocation[i] = -1; // Initialize allocation array
    }
    for (int i = 0; i < n; i++) {
        int worstIndex = -1;
        for (int j = 0; j < m; j++) {
```

```
            if (blockSize[j] >= processSize[i]) {
                if (worstIndex == -1 || blockSize[j] > blockSize[worstIndex]) {
                    worstIndex = j;
                }
            }
        }
        if (worstIndex != -1) {
            allocation[i] = worstIndex;
            blockSize[worstIndex] -= processSize[i];
        }
    }
    printf("Process 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[] = {100, 500, 200, 300, 600};
    int processSize[] = {212, 417, 112, 426};
    int m = sizeof(blockSize) / sizeof(blockSize[0]);
    int n = sizeof(processSize) / sizeof(processSize[0]);
    worstFit(blockSize, m, processSize, n);
    return 0;
}
```

Result:

The program successfully implements the Worst Fit memory allocation algorithm, assigning processes to the memory blocks with the largest available space.

Output: