

**EXP NO:** **10a) BEST FIT**

**DATE:16/3/25**

**PROGRAM:**

#include <stdio.h>

void bestFit(int blockSize[], int m, int processSize[], int n) {

    int allocation[n];

    for (int i = 0; i < n; i++) {

        allocation[i] = -1;

    }

    for (int i = 0; i < n; i++) {

        int bestIdx = -1;

        for (int j = 0; j < m; j++) {

            if (blockSize[j] >= processSize[i]) {

                if (bestIdx == -1 || blockSize[bestIdx] > blockSize[j]) {

                    bestIdx = j;

                }

            }

        }

        if (bestIdx != -1) {

            allocation[i] = bestIdx;

            blockSize[bestIdx] -= processSize[i];

        }

    }

    printf("Process No.\tProcess Size\tBlock No.\n");

    for (int i = 0; i < n; i++) {

        if (allocation[i] != -1) {

            printf("%d\t\t%d\t\t%d\n", i + 1, processSize[i], allocation[i] + 1);

        } else {

            printf("%d\t\t%d\t\tNot Allocated\n", i + 1, processSize[i]);

        }

    }

}

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]);

    bestFit(blockSize, m, processSize, n);

    return 0;

}

**OUTPUT:**

Process No. Process Size Block No.

1 212 4

2 417 2

3 112 3

4 426 5