```
package a9;
import java.util.*;

public class MemoryAllocation {

    // Method to allocate memory to blocks as per First Fit algorithm
    static void firstFit(int blockSize[], int m, int processSize[], int n) {
        int allocation[] = new int[n];
        Arrays.fill(allocation, -1);

        // pick each process and find suitable blocks
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                if (blockSize[j] >= processSize[i]) {
                    allocation[i] = j;
                    blockSize[j] -= processSize[i];
                    break;
                }
            }
        }

        System.out.println("Bhavika Verma TBCO22119");
        System.out.println("\nProcess No.\tProcess Size\tBlock no.");
        for (int i = 0; i < n; i++) {
            System.out.print(" " + (i + 1) + "\t\t" + processSize[i] + "\t\t");
            if (allocation[i] != -1)
                System.out.print(allocation[i] + 1);
            else
                System.out.print("Not Allocated");
            System.out.println();
        }
    }

    // Method to allocate memory to blocks as per Worst Fit algorithm
    static void worstFit(int blockSize[], int m, int processSize[], int n) {
        int allocation[] = new int[n];
        Arrays.fill(allocation, -1);

        // pick each process and find suitable blocks
        for (int i = 0; i < n; i++) {
            int wstIdx = -1;
            for (int j = 0; j < m; j++) {
                if (blockSize[j] >= processSize[i]) {
                    if (wstIdx == -1) {
                        wstIdx = j;
                    } else if (blockSize[wstIdx] < blockSize[j]) {
                        wstIdx = j;
                    }
                }
            }
            if (wstIdx != -1) {
                allocation[i] = wstIdx;
                blockSize[wstIdx] -= processSize[i];
            }
        }

        System.out.println("Sakshi Malusare TACO22150");
        System.out.println("\nProcess No.\tProcess Size\tBlock no.");
        for (int i = 0; i < n; i++) {
            System.out.print("   " + (i + 1) + "\t\t" + processSize[i] + "\t\t");
            if (allocation[i] != -1)
                System.out.print(allocation[i] + 1);
            else
                System.out.print("Not Allocated");
            System.out.println();
        }
    }

    // Main driver program with menu for user to select the algorithm
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Example data for block sizes and process sizes
        int blockSize[] = {100, 500, 200, 300, 600};
        int processSize[] = {212, 417, 112, 426};
        int m = blockSize.length;
        int n = processSize.length;

        // Displaying the user choice menu
        System.out.println("Select the memory allocation method:");
        System.out.println("1. First Fit");
        System.out.println("2. Worst Fit");
        System.out.println("Enter your choice (1/2):");

        int choice = scanner.nextInt();

        // Execute the selected memory allocation method
        switch (choice) {
            case 1:
                firstFit(blockSize, m, processSize, n);
                break;
            case 2:
                worstFit(blockSize, m, processSize, n);
                break;
            default:
                System.out.println("Invalid choice. Please select a valid option.");
                break;
        }
        scanner.close();
    }
}
```