

Name : Aakash A. Joshi

Roll no. : 0077

Subject : SPOS

Class : TE Computer

Batch: T4

Assignment no. 6.1(FirstFit)

Code:

```
import java.util.Scanner;
```

```
public class firstfit {  
    // 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];  
        for (int i = 0; i < allocation.length; i++)  
            allocation[i] = -1;  
  
        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("\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();  
        }  
    }  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter the number of processes: ");  
        int n = scanner.nextInt();  
        int processSize[] = new int[n];  
  
        System.out.println("Enter the process sizes:");  
        for (int i = 0; i < n; i++) {  
            processSize[i] = scanner.nextInt();  
        }  
    }  
}
```

```

    }

    System.out.print("Enter the number of blocks. ");
    int m = scanner.nextInt();
    int blockSize[] = new int[m];

    System.out.println("Enter the block sizes.");
    for (int i = 0; i < m; i++) {
        blockSize[i] = scanner.nextInt();
    }

    firstFit(blockSize, m, processSize, n);
    scanner.close();
}
}

```

Output:

Enter the number of processes: 4

Enter the process sizes:

212

417

112

426

Enter the number of blocks: 5

Enter the block sizes:

100

500

200

300

600

Process No.	Process Size	Block no.
1	212	2
2	417	5
3	112	2
4	426	Not Allocated

Name : Aakash A. Joshi

Roll no. : 0077

Subject : SPOS

Class : TE Computer

Batch: T4

Assignment no. 6.2(NextFit)

Code:

```
import java.util.*;
```

```
public class nextfit {
```

```
    static void NextFit(int blockSize[], int m, int processSize[], int n) {  
        int allocation[] = new int[n], j = 0, t = m - 1;  
        Arrays.fill(allocation, -1);
```

```
        for (int i = 0; i < n; i++) {  
            while (j < m) {  
                if (blockSize[j] >= processSize[i]) {  
                    allocation[i] = j;  
                    blockSize[j] -= processSize[i];  
                    t = (j - 1) % m;  
                    break;  
                }  
                if (t == j) {  
                    t = (j - 1) % m;  
                    break;  
                }  
                j = (j + 1) % m;  
            }  
        }  
    }
```

```
    System.out.print("\nProcess No.\tProcess Size\tBlock no.\n");  
    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("");  
    }  
}
```

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter the number of processes: ");  
    int n = scanner.nextInt();
```

```

int processSize[] = new int[n];

System.out.println("Enter the process sizes:");
for (int i = 0; i < n; i++) {
    processSize[i] = scanner.nextInt();
}

System.out.print("Enter the number of blocks: ");
int m = scanner.nextInt();
int blockSize[] = new int[m];

System.out.println("Enter the block sizes:");
for (int i = 0; i < m; i++) {
    blockSize[i] = scanner.nextInt();
}

NextFit(blockSize, m, processSize, n);
scanner.close();
}
}

```

Output:

Enter the number of processes: 3

Enter the process sizes:

10

20

5

Enter the number of blocks: 3

Enter the block sizes:

5

10

20

Process No.	Process Size	Block no.
1	10	2
2	20	3
3	5	1

Name : Aakash A. Joshi

Roll no. : 0077

Subject : SPOS

Class : TE Computer

Batch: T4

Assignment no. 6.3(BestFit)

Code:

```
import java.util.Scanner;
```

```
public class bestfit {
    static void bestFit(int blockSize[], int m, int processSize[], int n) {
        int allocation[] = new int[n];
        for (int i = 0; i < allocation.length; 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)
                        bestIdx = j;
                    else if (blockSize[bestIdx] > blockSize[j])
                        bestIdx = j;
                }
            }

            if (bestIdx != -1) {
                allocation[i] = bestIdx;
                blockSize[bestIdx] -= processSize[i];
            }
        }

        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();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of processes: ");
    }
}
```

```

int n = scanner.nextInt();
int processSize[] = new int[n];
System.out.println("Enter the process sizes.");
for (int i = 0; i < n; i++) {
    processSize[i] = scanner.nextInt();
}

System.out.print("Enter the number of blocks: ");
int m = scanner.nextInt();
int blockSize[] = new int[m];
System.out.println("Enter the block sizes.");
for (int i = 0; i < m; i++) {
    blockSize[i] = scanner.nextInt();
}

scanner.close();

bestFit(blockSize, m, processSize, n);
}
}

```

Output:

Enter the number of processes: 4

Enter the process sizes:

212

417

112

426

Enter the number of blocks: 5

Enter the block sizes:

100

500

200

300

600

Process No.	Process Size	Block no.
1	212	4
2	417	2
3	112	3
4	426	5

Name : Aakash A. Joshi

Roll no. : 0077

Subject : SPOS

Class : TE Computer

Batch: T4

Assignment no. 6.4(WorstFit)

Code:

```
import java.util.Scanner;
```

```
public class worstfit {
```

```
    // Method to allocate memory to blocks as per worst fit algorithm  
    static void worstFit(int blockSize[], int m, int processSize[], int n) {
```

```
        // Stores block id of the block allocated to a process  
        int allocation[] = new int[n];
```

```
        // Initially no block is assigned to any process  
        for (int i = 0; i < allocation.length; i++)  
            allocation[i] = -1;
```

```
        // pick each process and find suitable blocks according to its size and assign to it  
        for (int i = 0; i < n; i++) {
```

```
            // Find the best fit block for the current process
```

```
            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 we could find a block for the current process
```

```
            if (wstIdx != -1) {
```

```
                // Allocate block j to p[i] process
```

```
                allocation[i] = wstIdx;
```

```
                // Reduce available memory in this block.
```

```
                blockSize[wstIdx] -= processSize[i];
```

```
            }
```

```
        }
```

```
        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();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of processes: ");
        int n = scanner.nextInt();

        int processSize[] = new int[n];
        System.out.println("Enter the process sizes:");
        for (int i = 0; i < n; i++) {
            processSize[i] = scanner.nextInt();
        }

        System.out.print("Enter the number of blocks: ");
        int m = scanner.nextInt();

        int blockSize[] = new int[m];
        System.out.println("Enter the block sizes:");
        for (int i = 0; i < m; i++) {
            blockSize[i] = scanner.nextInt();
        }

        worstFit(blockSize, m, processSize, n);

        scanner.close();
    }
}

```

Output:

```

Enter the number of processes: 4
Enter the process sizes:
212
417
112
426
Enter the number of blocks: 5
Enter the block sizes:
100
500
200
300
600

```

Process No.	Process Size	Block no.
1	212	5
2	417	2
3	112	5

4

426

Not Allocated

Q