

MemoryAllo-Next Fit – 9

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

```
class NextFit {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Enter number of memory blocks: ");  
        int nb = sc.nextInt();  
        int block[] = new int[nb], rem[] = new int[nb];  
  
        System.out.println("Enter block sizes:");  
        for(int i = 0; i < nb; i++) {  
            block[i] = sc.nextInt();  
            rem[i] = block[i];  
        }  
  
        System.out.print("Enter number of processes: ");  
        int np = sc.nextInt();  
        int process[] = new int[np], alloc[] = new int[np];  
  
        System.out.println("Enter process sizes:");  
        for(int i = 0; i < np; i++) {  
            process[i] = sc.nextInt();  
            alloc[i] = -1;  
        }  
  
        int lastPos = 0; // remembers last allocated block  
  
        // Next Fit Allocation  
        for(int i = 0; i < np; i++) {
```

```

int count = 0;

boolean allocated = false;

while(count < nb) {
    int j = (lastPos + count) % nb;

    if(rem[j] >= process[i]) {
        alloc[i] = j;
        rem[j] -= process[i];
        lastPos = j; // update pointer
        allocated = true;
        break;
    }
    count++;
}

// Output Allocation Table
System.out.println("\nProcess\tSize\tBlock Allocated");
for(int i = 0; i < np; i++) {
    if(alloc[i] != -1)
        System.out.println("P" + (i+1) + "\t" + process[i] + "\tBlock " + (alloc[i]+1));
    else
        System.out.println("P" + (i+1) + "\t" + process[i] + "\tNot Allocated");
}

// Fragmentation Details
System.out.println("\nBlock\tInitial Size\tRemaining\tFragmentation");
int totalFrag = 0;
for(int i = 0; i < nb; i++) {
    System.out.println("B" + (i+1) + "\t" + block[i] + "\t\t" + rem[i] + "\t\t" + rem[i]);
}

```

```
        totalFrag += rem[i];
    }

    System.out.println("\nTotal Fragmented/Unused Memory: " + totalFrag);
}
}
```