// Java program for next fit

// memory management algorithm

import java.util.Arrays;

public class nextFit {

// Function to allocate memory to blocks as per Next fit

// algorithm

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

// Stores block id of the block allocated to a

// process

int allocation[] = new int[n], j = 0;

// Initially no block is assigned to any process

Arrays.fill(allocation, -1);

// pick each process and find suitable blocks

// according to its size ad assign to it

for (int i = 0; i < n; i++) {

// Do not start from beginning

int count =0;

while (j < m) {

count++; //makes sure that for every process we traverse through entire array maximum once only.This avoids the problem of going into infinite loop if memory is not available

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

// allocate block j to p[i] process

allocation[i] = j;

// Reduce available memory in this block.

blockSize[j] -= processSize[i];

break;

}

// mod m will help in traversing the blocks from

// starting block after we reach the end.

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("");

}

}

// Driver program

static public void main(String[] args) {

int blockSize[] = {5, 10, 20};

int processSize[] = {10, 20, 5};

int m = blockSize.length;

int n = processSize.length;

NextFit(blockSize, m, processSize, n);

}

}

// This code is contributed by Rajput-Ji

Output:

Process No. Process Size Block no.

1 10 2

2 20 3

3 5 1