**Bestfit:**

import java.util.\*;

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)

{

int blockSize[] = {100, 500, 200, 300, 600};

int processSize[] = {212, 417, 112, 426};

int m = blockSize.length;

int n = processSize.length;

bestFit(blockSize, m, processSize, n);

}

}

**Output**

****

**Firstfit:**

import java.util.\*;

public class FirstFit {

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println("Enter the Number of Memmory Blocks: ");

int block\_no=sc.nextInt();

int i=0,j;

int block[]=new int[10];

int flag[]=new int[10];

int display[]= new int [10];

int total=0,sum=0;

Arrays.fill(flag,0);

Arrays.fill(display,0);

for(i=0;i<block\_no;i++)

{

System.out.println("Enter the memory capacity of B"+(i+1));

block[i]=sc.nextInt();

total=total+block[i];

}

System.out.println("Enter the Number of Proesses: ");

int process\_no=sc.nextInt();

int process[]=new int[10];

for(i=0;i<process\_no;i++)

{

System.out.println("Enter the memory required for Process"+(i+1));

process[i]=sc.nextInt();

}

System.out.println("Blockm Distribution: ");

for(i=0;i<block\_no;i++)

{

System.out.println("B"+(i+1)+"\t"+block[i]);

}

System.out.println("\n\nProcess Distribution: ");

for(i=0;i<process\_no;i++)

{

System.out.println("P"+(i+1)+"\t"+process[i]);

}

for(i=0;i<process\_no;i++)

{

for(j=0;j<block\_no;j++)

{

if(process[i]<block[j])

{

if(flag[j]==0)

{

sum=sum+process[i];

display[j]=i+1;

flag[j]=1;

break;

}

}

}

}

System.out.println("\n\nProccess Alocation In blocks");

for(i=0;i<block\_no;i++){

if(display[i]==0){

System.out.println("B"+(i+1)+"\t"+"0");

}

else{

System.out.println("B"+(i+1)+"\t"+"P"+display[i]);

}

}

double efficiency = (sum\*100)/total;

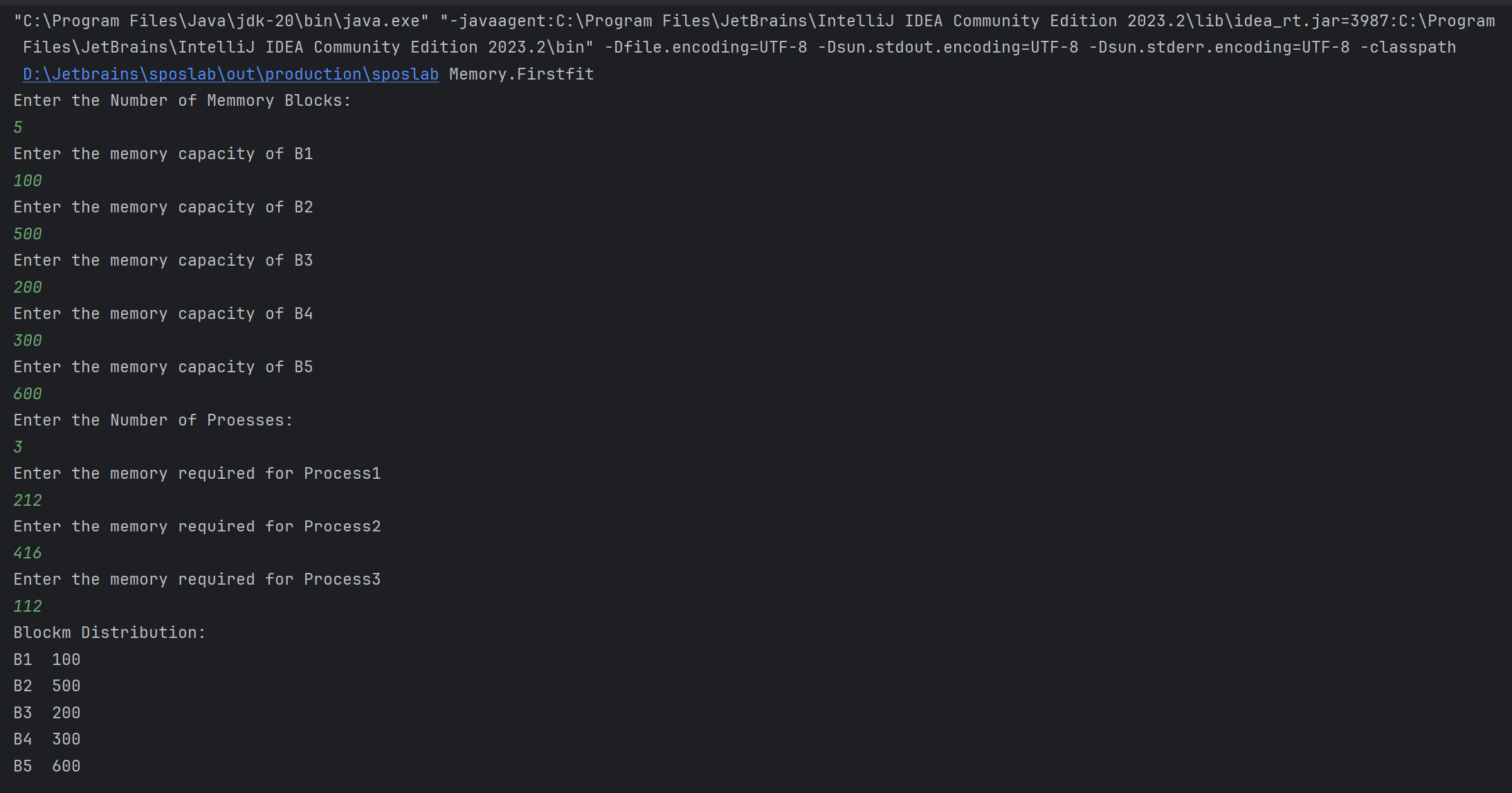
System.out.println("\n \nEfficiency :"+efficiency);

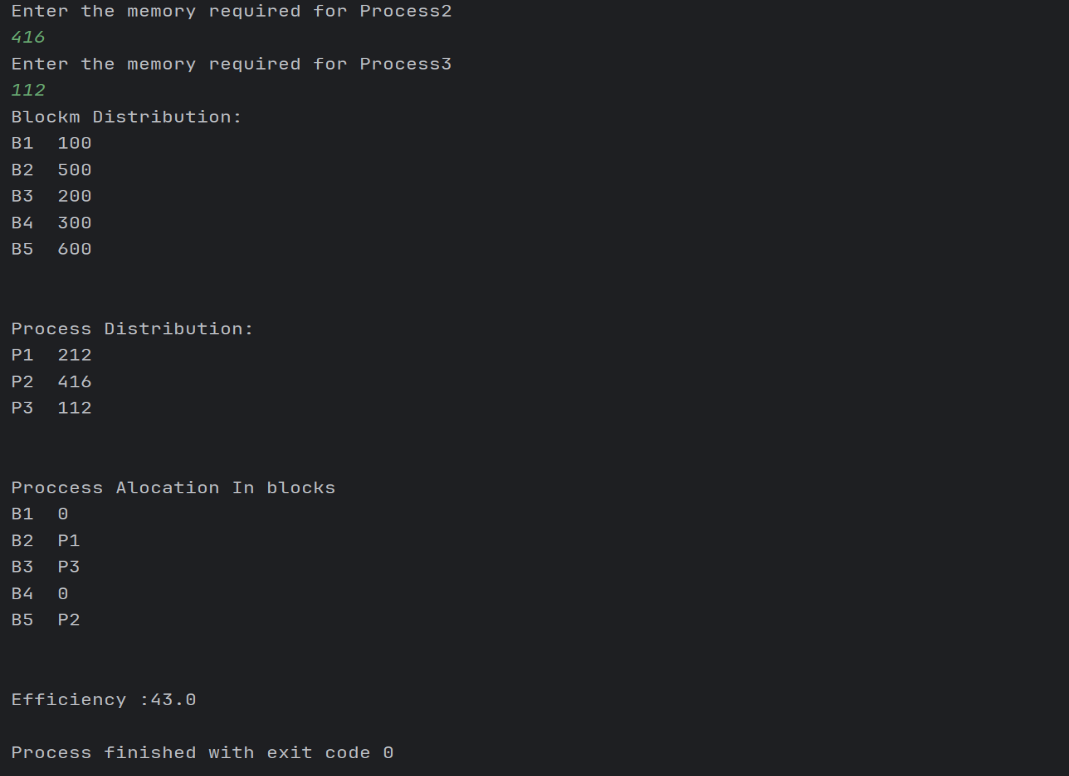
sc.close();

}

}

**Output:**

****

****

**Nextfit:**

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\t\t" + processSize[i]

+ "\t\t\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) {

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

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

int m = blockSize.length;

int n = processSize.length;

NextFit(blockSize, m, processSize, n);

}

}

**Output:**

