**ASSIGNMENT NO:08-Group B**

Name:Mrunal R Somankar

Roll No:TCOA46

package FIT;

import java.util.Scanner;

public class FirstFit {

    static int job[];

    static int block[];

    static int js, bs;

    static Scanner input = new Scanner(System.in);

    static int Allocation[];

    public static void main(String args[]) {

        FirstFit MA = new FirstFit();

        while (true) {

            System.out.println("Menu:");

            System.out.println("\n1.Read Data - Job No. & Size, Block No. & Size \n2.First Fit \n3.Best Fit \n4.Worst Fit \n5.Next Fit\n6.Exit");

            System.out.println("Enter Your Choice:");

            int ch = Integer.parseInt(input.nextLine());

            switch (ch) {

                case 1:

                    System.out.println("\nEnter total no. of jobs:");

                    js = Integer.parseInt(input.nextLine());

                    System.out.println("\nEnter total no. of blocks:");

                    bs = Integer.parseInt(input.nextLine());

                    job = new int[js];

                    block = new int[bs];

                    MA.ReadData(js, bs);

                    break;

                case 2:

                    MA.Firstfit();

                    break;

                case 3:

                    MA.BestFit();

                    break;

                case 4:

                    MA.WorstFit();

                    break;

                case 5:

                    MA.NextFit();

                    break;

                case 6:

                    System.exit(0);

                    break;

            }

        }

    }

    void ReadData(int n, int m) {

        for (int i = 0; i < n; i++) {

            System.out.println("Enter Job Size:");

            job[i] = Integer.parseInt(input.nextLine());

        }for (int i = 0; i < m; i++) {

            System.out.println("Enter Block Size:");

            block[i] = Integer.parseInt(input.nextLine());

        }

    }

    void Firstfit() {

        int flag = 0;

        Allocation = new int[js];

        for (int i = 0; i < Allocation.length; i++) {

            Allocation[i] = -1;

        }for (int i = 0; i < js; i++) {

            for (int j = 0; j < bs; j++) {

                flag = 0;

                if (block[j] >= job[i]) {

                    for (int k = 0; k < js; k++) {

                        if (Allocation[k] == j)

                            flag = 1;

                    }

                    if (flag == 0) {

                        Allocation[i] = j;

                        break;

                    }

                }

            }

        }

        Display();

    }

    void Display() {

        System.out.println("\tJob No.\tJob Size\tBlock No.\tFragment");

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

            System.out.print("\t" + (i+1) + "\t\t" + job[i] + "\t\t");

            if (Allocation[i] != -1) {

                System.out.println(Allocation[i] + "\t\t" + (block[Allocation[i]] - job[i]));

            } else {

                System.out.println("Not Allocated");

            }

        }

    }

    void BestFit() {

        int flag = 0;

        Allocation = new int[js];

        for (int i = 0; i < Allocation.length; i++) {

            Allocation[i] = -1;

        }

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

            int BestInd = -1;

            for (int j = 0; j < bs; j++) {

                flag = 0;

                if (block[j] >= job[i]) {

                    for (int k = 0; k < js; k++) {

                        if (Allocation[k] == j) {

                            flag = 1;

                            break;

                        }

                    }

                    if (BestInd == -1 && flag == 0) {

                        BestInd = j;

                    } else if (flag == 0 && block[BestInd] > block[j]) {

                        BestInd = j;

                    } else {

                        continue;

                    }

                }

            }

            if (BestInd != -1) {

                Allocation[i] = BestInd;

            }

        }

        Display();

    }

    void WorstFit() {

        Allocation = new int[js];

        for (int i = 0; i < Allocation.length; i++) {

            Allocation[i] = -1;

        }

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

            int worstBlockIndex = -1;

            for (int j = 0; j < bs; j++) {

                if (block[j] >= job[i]) {

                    if (worstBlockIndex == -1 || block[j] > block[worstBlockIndex]) {

                        worstBlockIndex = j;

                    }

                }

            }

            if (worstBlockIndex != -1) {

                Allocation[i] = worstBlockIndex;

                block[worstBlockIndex] -= job[i];

            }

        }

        Display();

    }

    void NextFit() {

        int flag = 0;

        Allocation = new int[js];

        for (int i = 0; i < Allocation.length; i++) {

            Allocation[i] = -1;

        }

        int lastAllocatedBlockIndex = -1;

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

            for (int j = lastAllocatedBlockIndex + 1; j < bs; j++) {

                flag = 0;

                if (block[j] >= job[i]) {

                    lastAllocatedBlockIndex = j;

                    for (int k = 0; k < js; k++) {

                        if (Allocation[k] == j) {

                            flag = 1;

                            break;

                        }

                    }

                    if (flag == 0) {

                        Allocation[i] = j;

                        break;

                    }

                }

            }

        }

        Display();

    }

}

**Output:**

**PS C:\Users\soman\Desktop\5th SEM\SPOS\spos lab> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\soman\AppData\Roaming\Code\User\workspaceStorage\4f18fa9c53191b0f16505535b0496562\redhat.java\jdt\_ws\spos lab\_9706026e\bin' 'FIT.FirstFit'**

**Menu:**

**1.Read Data - Job No. & Size, Block No. & Size**

**2.First Fit**

**3.Best Fit**

**4.Worst Fit**

**5.Next Fit**

**6.Exit**

**Enter Your Choice:**

**1**

**Enter total no. of jobs:**

**5**

**Enter total no. of blocks:**

**5**

**Enter Job Size:**

**80**

**Enter Job Size:**

**60**

**Enter Job Size:**

**25**

**Enter Job Size:**

**15**

**Enter Job Size:**

**30**

**Enter Block Size:**

**100**

**Enter Block Size:**

**20**

**Enter Block Size:**

**80**

**Enter Block Size:**

**30**

**Enter Block Size:**

**40**

**Menu:**

**1.Read Data - Job No. & Size, Block No. & Size**

**2.First Fit**

**3.Best Fit**

**4.Worst Fit**

**5.Next Fit**

**6.Exit**

**Enter Your Choice:**

**2**

**Job No. Job Size Block No. Fragment**

**1 80 0 20**

**2 60 2 20**

**3 25 3 5**

**4 15 1 5**

**5 30 4 10**

**Menu:**

**1.Read Data - Job No. & Size, Block No. & Size**

**2.First Fit**

**3.Best Fit**

**4.Worst Fit**

**5.Next Fit**

**6.Exit**

**Enter Your Choice:**

**5**

**Job No. Job Size Block No. Fragment**

**1 80 0 20**

**2 60 2 20**

**3 25 3 5**

**4 15 4 25**

**5 30 Not Allocated**

**Menu:**

**1.Read Data - Job No. & Size, Block No. & Size**

**2.First Fit**

**3.Best Fit**

**4.Worst Fit**

**5.Next Fit**

**6.Exit**

**Enter Your Choice:**

**3**

**Job No. Job Size Block No. Fragment**

**1 80 2 0**

**2 60 0 40**

**3 25 3 5**

**4 15 1 5**

**5 30 4 10**

**Menu:**

**1.Read Data - Job No. & Size, Block No. & Size**

**2.First Fit**

**3.Best Fit**

**4.Worst Fit**

**5.Next Fit**

**6.Exit**

**Enter Your Choice:**

**4**

**Job No. Job Size Block No. Fragment**

**1 80 0 -60**

**2 60 2 -40**

**3 25 4 -10**

**4 15 3 0**

**5 30 Not Allocated**