**Group : B**

**Assignment No. : 6**

**Title : Write a program to implement Memory Allocation Strategies.**

**Roll No. : 2365**

**-------------------------------------------------------------------------------------------**

import java.util.\*;

public class MATrial

{

public static void main(String args[])

{

Scanner scan=new Scanner(System.in);

int nBlock,nProcess;

int block[]=new int[10];

int process[]=new int[10];

int pBlock[]=new int[10];

int bFlag[]=new int[10];

int pFlag[]=new int[10];

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

nBlock=scan.nextInt();

System.out.println("\nEnter Block Size:");

for(int i=0;i<nBlock;i++)

{

block[i]=scan.nextInt();

bFlag[i]=0;

}

System.out.print("\nEnter no. of processes:");

nProcess=scan.nextInt();

System.out.println("\nEnter process size:");

for(int i=0;i<nProcess;i++)

{

process[i]=scan.nextInt();

pFlag[i]=0;

}

int choice;

do

{

System.out.println("\nMenu\n1.First Fit\n2.Best Fit\n3.Worst Fit\n4.Exit");

System.out.print("\nEnter Choice:");

choice=scan.nextInt();

switch(choice)

{

case 1:

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

for(int j=0;j<nBlock;j++)

{

if(process[i]<=block[j] && bFlag[j]==0)

{

pBlock[i]=block[j];

bFlag[j]=1;

pFlag[i]=1;

break;

}

}

}

}

System.out.println("\n[Size]:[Block]\n");

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==1)

{

System.out.println("["+process[i]+"]:["+pBlock[i]+"]");

}

}

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

System.out.println("Process "+process[i]+" is not allocated a block.");

}

}

for(int i=0;i<nBlock;i++)

{

pFlag[i]=0;

bFlag[i]=0;

}

break;

case 2:

for(int i=0;i<nBlock;i++)

for(int j=i+1;j<nBlock;j++)

if(block[i]>block[j])

{

int temp=block[i];

block[i]=block[j];

block[j]=temp;

}

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

for(int j=0;j<nBlock;j++)

{

if(process[i]<=block[j] && bFlag[j]==0)

{

pBlock[i]=block[j];

bFlag[j]=1;

pFlag[i]=1;

break;

}

}

}

}

System.out.println("\n[Size]:[Block]\n");

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==1)

{

System.out.println("["+process[i]+"]:["+pBlock[i]+"]");

}

}

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

System.out.println("Process "+process[i]+" is not allocated a block.");

}

}

for(int i=0;i<nBlock;i++)

{

pFlag[i]=0;

bFlag[i]=0;

}

break;

case 3:

for(int i=0;i<nBlock;i++)

for(int j=i+1;j<nBlock;j++)

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

{

int temp=block[i];

block[i]=block[j];

block[j]=temp;

}

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

for(int j=0;j<nBlock;j++)

{

if(process[i]<=block[j] && bFlag[j]==0)

{

pBlock[i]=block[j];

bFlag[j]=1;

pFlag[i]=1;

break;

}

}

}

}

System.out.println("\n[Size]:[Block]\n");

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==1)

{

System.out.println("["+process[i]+"]:["+pBlock[i]+"]");

}

}

for(int i=0;i<nProcess;i++)

{

if(pFlag[i]==0)

{

System.out.println("Process "+process[i]+" is not allocated a block.");

}

}

for(int i=0;i<nBlock;i++)

{

bFlag[i]=0;

pFlag[i]=0;

}

break;

case 4:

System.out.println("Thankyou");

break;

default:System.out.println("Invalid Choice");

}

}while(choice!=4);

}

}

/\*

Enter no. of blocks:5

Enter Block Size:

100

500

200

300

600

Enter no. of processes:4

Enter process size:

112

417

312

426

Menu

1.First Fit

2.Best Fit

3.Worst Fit

4.Exit

Enter Choice:1

[Size]:[Block]

[112]:[500]

[417]:[600]

Process 312 is not allocated a block.

Process 426 is not allocated a block.

Menu

1.First Fit

2.Best Fit

3.Worst Fit

4.Exit

Enter Choice:2

[Size]:[Block]

[112]:[200]

[417]:[500]

[312]:[600]

Process 426 is not allocated a block.

Menu

1.First Fit

2.Best Fit

3.Worst Fit

4.Exit

Enter Choice:3

[Size]:[Block]

[112]:[600]

[417]:[500]

Process 312 is not allocated a block.

Process 426 is not allocated a block.

Menu

1.First Fit

2.Best Fit

3.Worst Fit

4.Exit

Enter Choice:4

Thankyou

\*/