FAIZAN CHOUDHARY

20BCS021

OS LAB

7th April 2022

CODE: (code pasted in this format for readability)

```
#include <iostream>
#include <limits.h>
using namespace std;
int n, no;
// array to store process indices for each block index
int allocation_block[100] = {-1};
int totIntFrag=0, totExtFrag=0;
// temp array to store size of blocks for display
int temp[100];
// array to store internal fragmentation of each block
int intFrag[100] = {0};
// array to store the occupancy status of each block
bool occupied_block[100] = {false};
// counter to keep track of allocated processes
int counter=0;


void display (int *s_b, int *s_p) {
    cout<<"\nEntered block sizes:\n";
    cout<<"| ";
    for (int i=0; i<n; i++)
        cout<<temp[i]<<" | ";
    cout<<endl;
    cout<<"Entered process sizes:\n";
    cout<<"| ";
    for (int i=0; i<no; i++)
        cout<<s_p[i]<<" | ";
    cout<<endl;
    cout<<"\nAfter allocation:\n";
    cout<<"\nBLOCK ID\tBLOCK SIZE\tPROCESS\t\tINTERNAL FRAGMENTATION\n";
    for (int i=0; i<n; i++) {
        cout<<i+1<<"\t\t  "<<temp[i]<<"\t\t";
        // if block is actually allocated a process
        if (occupied_block[i] == false || allocation_block[i] == -1)
            cout<<"--\t\t\t--";
        else if (allocation_block[i] != -1) {
            cout<<s_p[allocation_block[i]]<<" (P"<<allocation_block[i] + 1<<")\t\t";
            cout<<intFrag[i];
        }
        cout<<endl;
    }
    cout<<"\nTotal Internal Fragmentation: "<<totIntFrag;
```

```
    cout<<"\nTotal External Fragmentation: "<<totExtFrag<<endl<<endl;

}

void worstFit (int *s_b, int *s_p) {
    for (int i=0; i<n; i++)
        temp[i] = s_b[i];

    for (int i=0; i<no; i++) {
        // to store the index of the worst fit
        int idx = -1;
        for (int j=0; j<n; j++) {
            if (s_b[j] >= s_p[i] && (idx == -1 || s_b[idx] < s_b[j]) && occupied_block[j]
== false)
                idx = j;
        }

        // for a successful worst fit
        if (idx != -1) {
            counter++;
            allocation_block[idx] = i;
            occupied_block[idx] = true;
            intFrag[idx] = s_b[idx] - s_p[i];
            s_b[idx] -= s_p[i];
        }
    }

    for (int i=0; i<n; i++) {
        // cout<<allocation_block[i]<<endl;
        if (occupied_block[i] == true)
            totIntFrag += intFrag[i];
        if (occupied_block[i] == false && counter < no)
            totExtFrag += s_b[i];
    }
}

int main() {
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
    cout<<"\nWorst Fit Memory Management\n";

    cout<<"\nEnter number of memory blocks: ";
    cin>>n;

    int size_blocks[100];
    cout<<"\nEnter the size of each block:\n";
    for (int i=0; i<n; i++)
        cin>>size_blocks[i];

    cout<<"\nEnter number of processes: ";
    cin>>no;

    int size_processes[100];
    cout<<"\nEnter the size of each process:\n";
    for (int i=0; i<no; i++)
```

```
        cin>>size_processes[i];

    worstFit (size_blocks, size_processes);
    display (size_blocks, size_processes);
    return 0;
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

Worst Fit Memory Management

Enter number of memory blocks: 5

Enter the size of each block:
100 500 200 300 600

Enter number of processes: 4

Enter the size of each process:
212 417 112 426

Entered block sizes:
| 100 | 500 | 200 | 300 | 600 |
Entered process sizes:
| 212 | 417 | 112 | 426 |
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 100           --                      --
2                 500           417 (P2)                83
3                 200           --                      --
4                 300           112 (P3)                188
5                 600           212 (P1)                388

Total Internal Fragmentation: 659
Total External Fragmentation: 300
```

```
FAIZAN CHOUDHARY
20BCS021

Worst Fit Memory Management

Enter number of memory blocks: 5

Enter the size of each block:
200 100 300 400 500

Enter number of processes: 4

Enter the size of each process:
250 200 100 350

Entered block sizes:
| 200 | 100 | 300 | 400 | 500 |
Entered process sizes:
| 250 | 200 | 100 | 350 |
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 200           --                      --
2                 100           --                      --
3                 300           100 (P3)                200
4                 400           200 (P2)                200
5                 500           250 (P1)                250

Total Internal Fragmentation: 650
Total External Fragmentation: 300
```

```
FAIZAN CHOUDHARY
20BCS021

Worst Fit Memory Management

Enter number of memory blocks: 5

Enter the size of each block:
200 100 300 400 500

Enter number of processes: 4

Enter the size of each process:
450 210 210 350

Entered block sizes:
| 200 | 100 | 300 | 400 | 500 |
Entered process sizes:
| 450 | 210 | 210 | 350 |
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 200           --                      --
2                 100           --                      --
3                 300           210 (P3)                90
4                 400           210 (P2)                190
5                 500           450 (P1)                50

Total Internal Fragmentation: 330
Total External Fragmentation: 300
```