FAIZAN CHOUDHARY

20BCS021

OS LAB

24th March 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<<"\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 nextFit (int *s_b, int *s_p) {
    for (int i=0; i<n; i++)
        temp[i] = s_b[i];

    int j=0;

    for (int i=0; i<no; i++) {
```

```
        while (j<n) {
            if (s_b[j] >= s_p[i]) {

                if (occupied_block[j] == false) {
                    counter++;
                    allocation_block[j] = i;
                    occupied_block[j] = true;

                    intFrag[j] = s_b[j] - s_p[i];
                    // cout<<intFrag[j]<<endl;
                    // subtracting the value of memory that has been allocated
                    s_b[j] -= s_p[i];
                    j = (j+1) % n;
                }
                break;
            }
            // to maintain the property of the next fit
            j = (j+1) % n;
        }
    }

    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<<"\nNext 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];

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

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

Next Fit Memory Management

Enter number of memory blocks: 3

Enter the size of each block:
5 10 20

Enter number of processes: 3

Enter the size of each process:
10 20 5
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 5             5 (P3)          0
2                 10            10 (P1)         0
3                 20            20 (P2)         0

Total Internal Fragmentation: 0
Total External Fragmentation: 0
```

```
FAIZAN CHOUDHARY
20BCS021

Next Fit Memory Management

Enter number of memory blocks: 5

Enter the size of each block:
100 500 200 450 600

Enter number of processes: 4

Enter the size of each process:
212 417 112 426
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 100           --                      --
2                 500           212 (P1)                288
3                 200           --                      --
4                 450           417 (P2)                33
5                 600           112 (P3)                488

Total Internal Fragmentation: 809
Total External Fragmentation: 300
```

```
FAIZAN CHOUDHARY
20BCS021

Next 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
```

```
After allocation:

BLOCK ID        BLOCK SIZE      PROCESS         INTERNAL FRAGMENTATION
1                 200           --                      --
2                 100           --                      --
3                 300           250 (P1)                50
4                 400           200 (P2)                200
5                 500           100 (P3)                400

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