# 20BCS402 MOHD ADIL Program 12| Worst fit memory management algorithm.

#include <iostream>

#include <vector>

using namespace std;

struct Process

{

    char Pname[3];

    int memory;

    bool allocated = false;

};

struct Block

{

    int size;

    bool used = false;

    int rem;

    struct Process processAllocated;

};

int main()

{

    cout << "No. of block : ";

    int n;

    cin >> n;

    vector<Block> blocks;

    cout << "Enter Size of the " << n << " Blocks: ";

    for (int i = 0; i < n; i++)

    {

        Block tempBlock;

        cin >> tempBlock.size;

        tempBlock.rem = tempBlock.size;

        blocks.push\_back(tempBlock);

    }

    cout << "No. of Process : ";

    int m;

    cin >> m;

    vector<Process> Processes;

    cout << "Enter Name and size of the Processes: ";

    for (int i = 0; i < m; i++)

    {

        Process tempProcess;

        cin >> tempProcess.Pname;

        cin >> tempProcess.memory;

        Processes.push\_back(tempProcess);

    }

    // memory allocation

    for (int i = 0; i < m; i++)

    {

        bool exist = false;

        int index, max = INT16\_MIN;

        for (int j = 0; j < n; j++)

        {

            if (Processes[i].memory <= blocks[j].rem && blocks[j].used == false && blocks[j].rem > max)

            {

                max = blocks[j].rem;

                exist = true;

                index = j;

            }

        }

        if (exist)

        {

            Processes[i].allocated = true;

            blocks[index].used = true;

            blocks[index].rem = blocks[index].size - Processes[i].memory;

            blocks[index].processAllocated = Processes[i];

        }

    }

    cout << "\tBlock Number\tSize\tProcess Allocated\tInternal Fragmentation" << endl;

    for (int i = 0; i < n; i++)

    {

        if (blocks[i].used == true)

        {

            cout << "\t\t" << i + 1 << "\t" << blocks[i].size << "\t\t" << blocks[i].processAllocated.Pname << "\t\t\t" << blocks[i].rem << endl;

        }

        else

        {

            cout << "\t\t" << i + 1 << "\t" << blocks[i].size << "\t\t"

                 << "---"

                 << "\t\t\t"

                 << "---" << endl;

        }

    }

    bool flag = true;

    for (int i = 0; i < m; i++)

    {

        if (Processes[i].allocated == false)

        {

            flag = false;

            break;

        }

        else

        {

            continue;

        }

    }

    int IF = 0, EF = 0;

    for (int i = 0; i < n; i++)

    {

        if (blocks[i].used == true)

        {

            IF += blocks[i].rem;

        }

        else

        {

            if (flag == false)

            {

                EF += blocks[i].rem;

            }

        }

    }

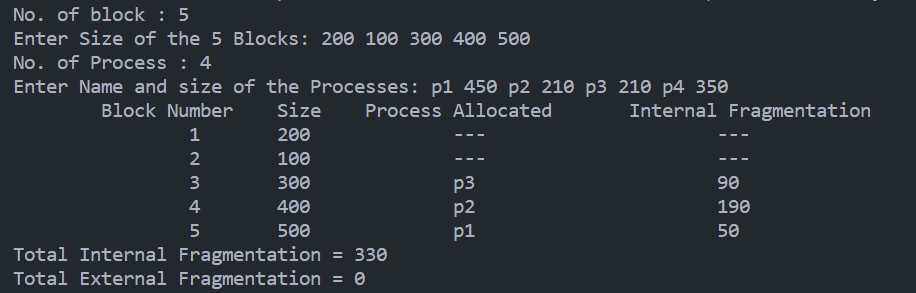
    cout << "Total Internal Fragmentation = " << IF << endl;

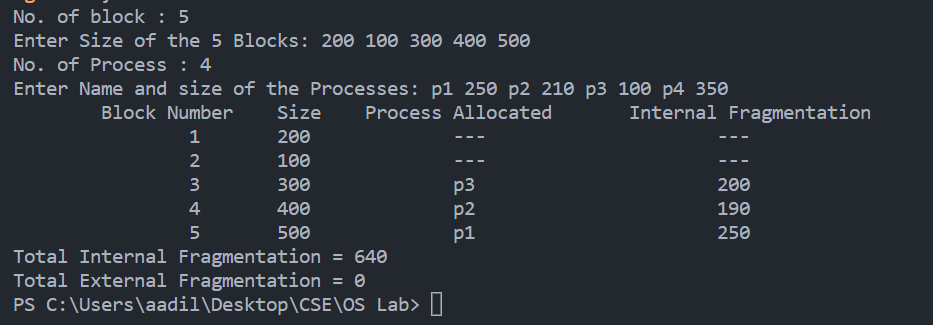
    cout << "Total External Fragmentation = " << EF << endl;

    return 0;

}

**Output**

****

****

Thank you