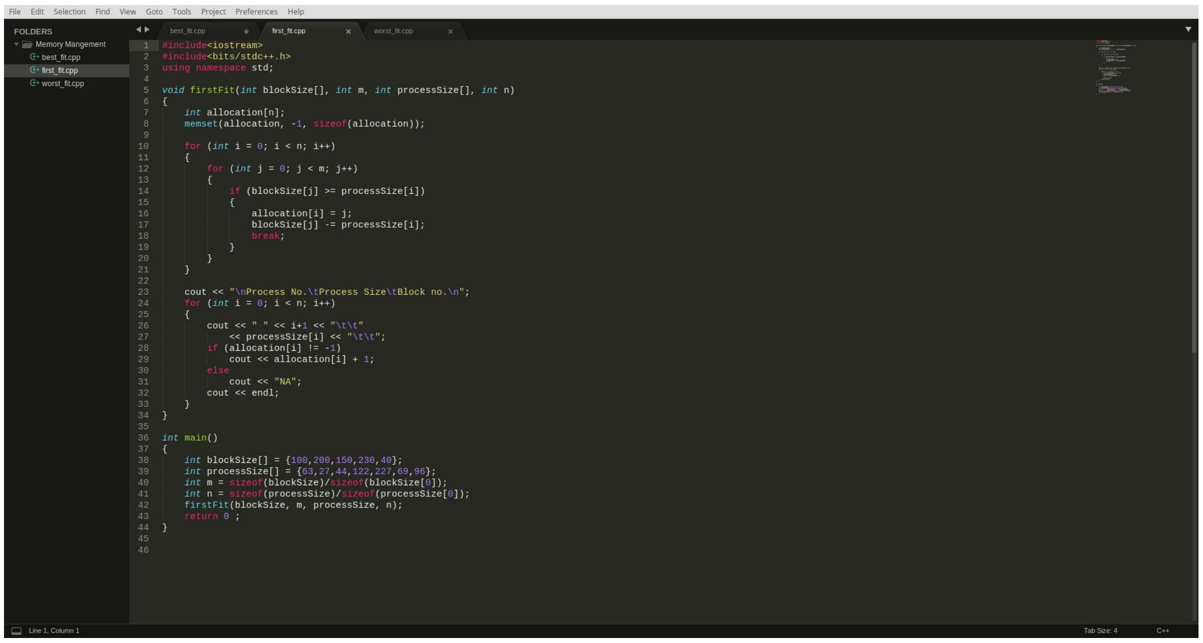
LAB MANUAL

JATIN KARTHIK TRIPATHY

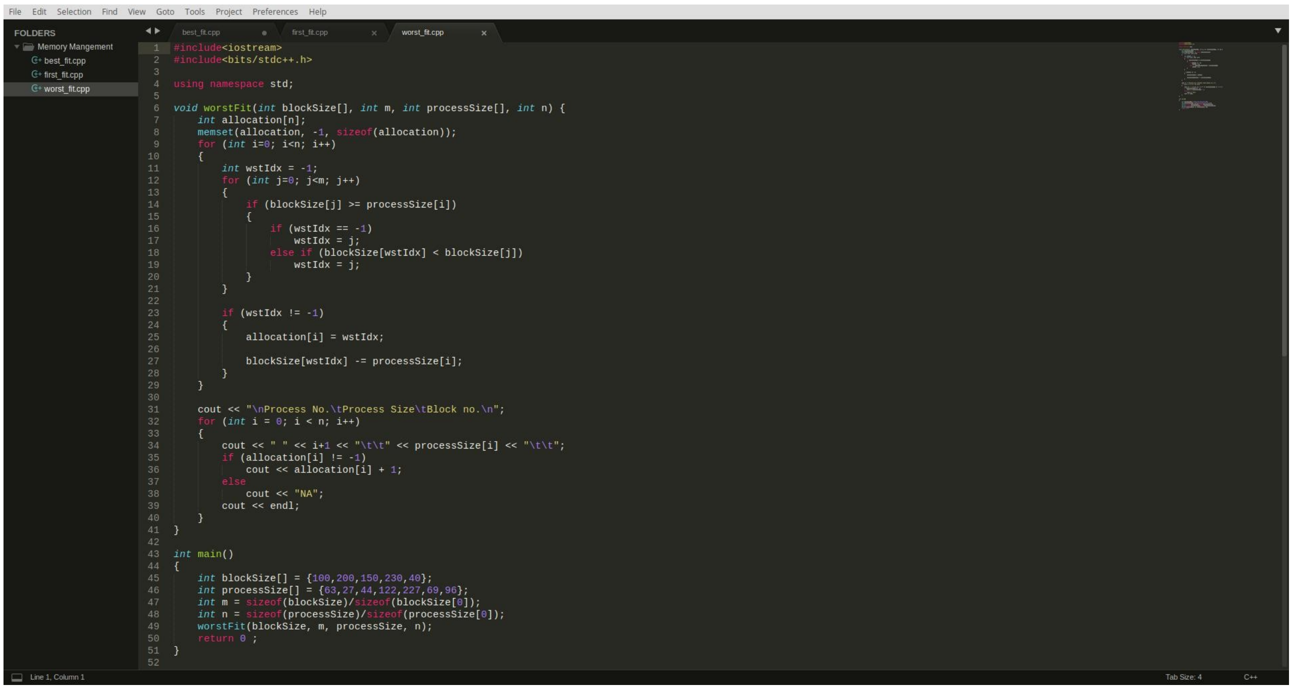
(17BCE7106)

1. First Fit



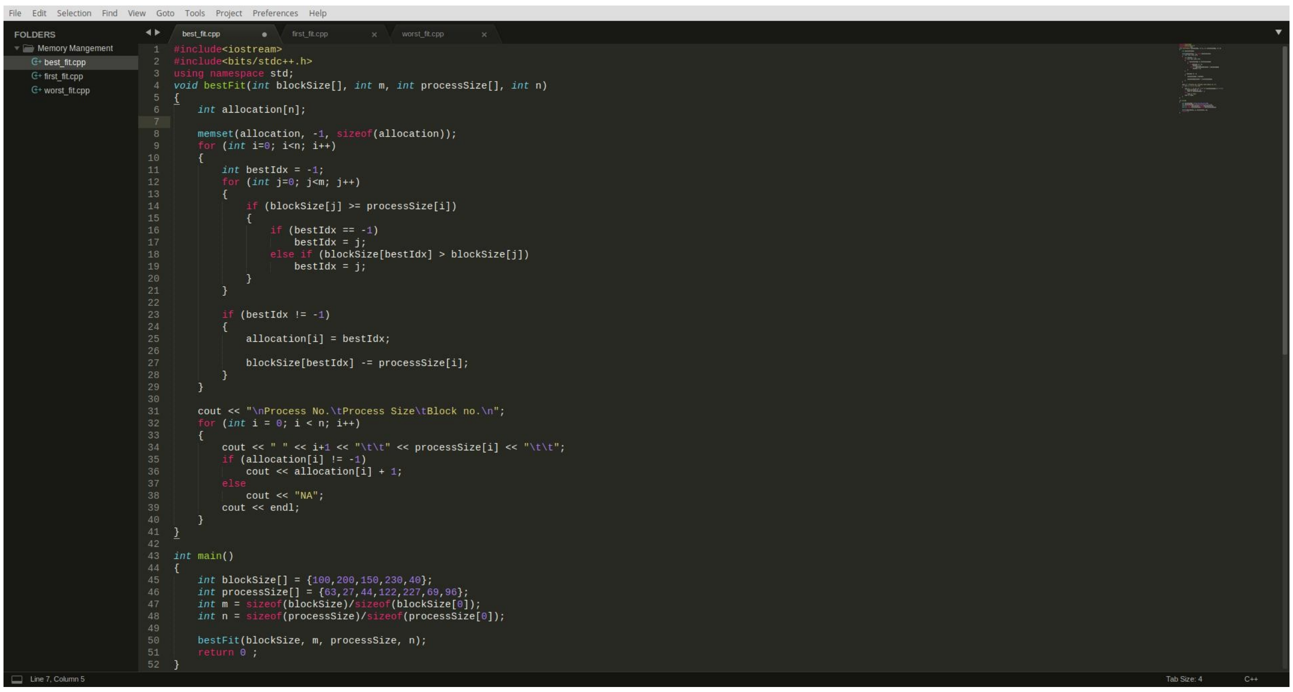
```cpp
#include<iostream>
#include<bits/stdc++.h>
using namespace std;

void firstFit(int blockSize[], int m, int processSize[], int n)
{
    int allocation[n];
    memset(allocation, -1, sizeof(allocation));

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (blockSize[j] >= processSize[i])
            {
                allocation[i] = j;
                blockSize[j] -= processSize[i];
                break;
            }
        }
    }

    cout << "\nProcess No.\tProcess Size\tBlock no.\n";
    for (int i = 0; i < n; i++)
    {
        cout << " " << i+1 << "\t\t"
            << processSize[i] << "\t\t";
        if (allocation[i] != -1)
            cout << allocation[i] + 1;
        else
            cout << "NA";
        cout << endl;
    }
}

int main()
{
    int blockSize[] = {100,200,150,230,40};
    int processSize[] = {63,27,44,122,227,69,96};
    int m = sizeof(blockSize)/sizeof(blockSize[0]);
    int n = sizeof(processSize)/sizeof(processSize[0]);
    firstFit(blockSize, m, processSize, n);
    return 0 ;
}


```

2. Worst Fit



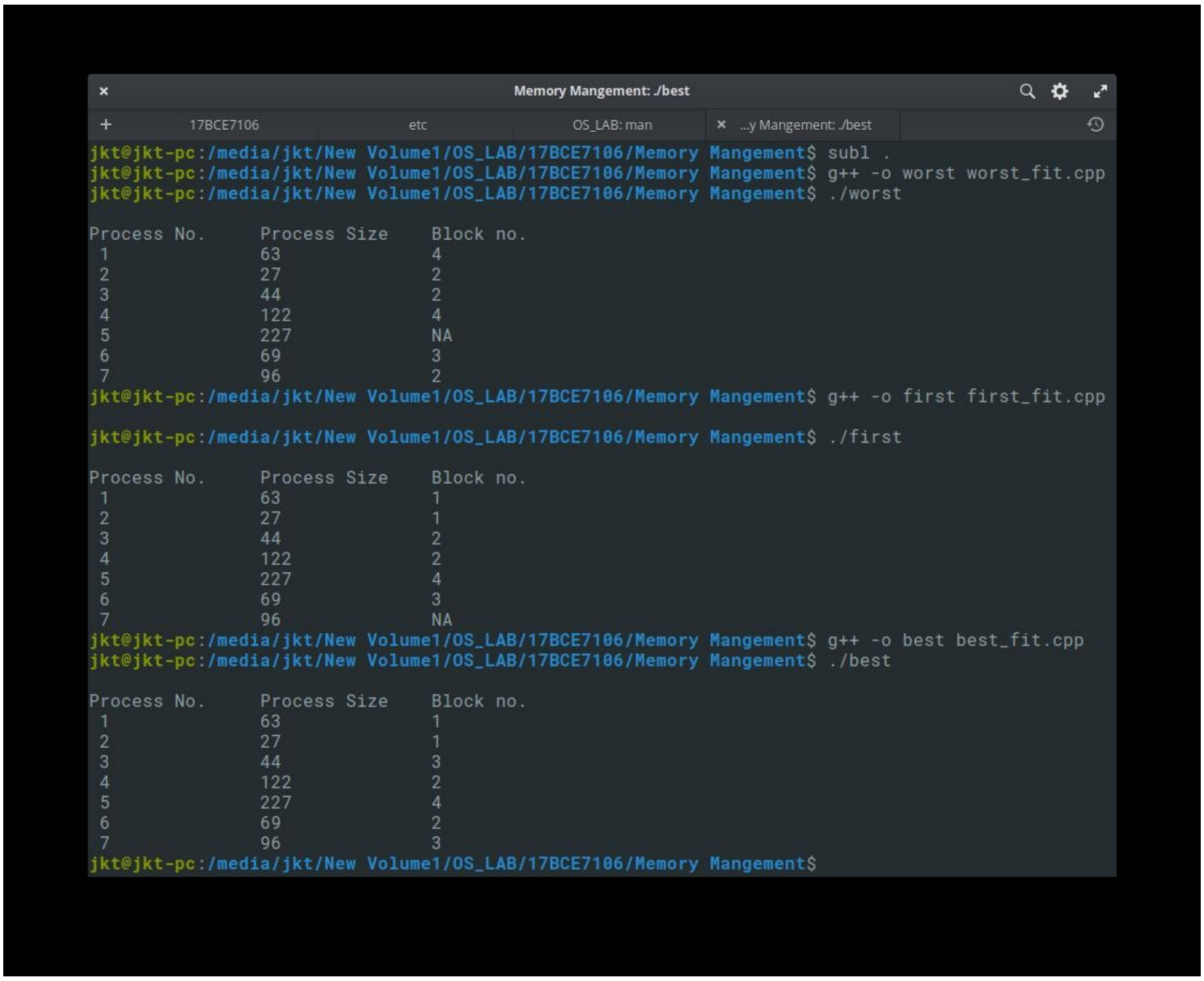
```cpp
#include<iostream>
#include<bits/stdc++.h>

using namespace std;

void worstFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];
    memset(allocation, -1, sizeof(allocation));
    for (int i=0; i<n; i++)
    {
        int wstIdx = -1;
        for (int j=0; j<m; j++)
        {
            if (blockSize[j] >= processSize[i])
            {
                if (wstIdx == -1)
                    wstIdx = j;
                else if (blockSize[wstIdx] < blockSize[j])
                    wstIdx = j;
            }
        }

        if (wstIdx != -1)
        {
            allocation[i] = wstIdx;

            blockSize[wstIdx] -= processSize[i];
        }
    }

    cout << "\nProcess No.\tProcess Size\tBlock no.\n";
    for (int i = 0; i < n; i++)
    {
        cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
        if (allocation[i] != -1)
            cout << allocation[i] + 1;
        else
            cout << "NA";
        cout << endl;
    }
}

int main()
{
    int blockSize[] = {100,200,150,230,40};
    int processSize[] = {63,27,44,122,227,69,96};
    int m = sizeof(blockSize)/sizeof(blockSize[0]);
    int n = sizeof(processSize)/sizeof(processSize[0]);
    worstFit(blockSize, m, processSize, n);
    return 0 ;
}

```

3. Best Fit



```cpp
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
void bestFit(int blockSize[], int m, int processSize[], int n)
{
    int allocation[n];

    memset(allocation, -1, sizeof(allocation));
    for (int i=0; i<n; i++)
    {
        int bestIdx = -1;
        for (int j=0; j<m; j++)
        {
            if (blockSize[j] >= processSize[i])
            {
                if (bestIdx == -1)
                    bestIdx = j;
                else if (blockSize[bestIdx] > blockSize[j])
                    bestIdx = j;
            }
        }

        if (bestIdx != -1)
        {
            allocation[i] = bestIdx;

            blockSize[bestIdx] -= processSize[i];
        }
    }

    cout << "\nProcess No.\tProcess Size\tBlock no.\n";
    for (int i = 0; i < n; i++)
    {
        cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";
        if (allocation[i] != -1)
            cout << allocation[i] + 1;
        else
            cout << "NA";
        cout << endl;
    }
}

int main()
{
    int blockSize[] = {100,200,150,230,40};
    int processSize[] = {63,27,44,122,227,69,96};
    int m = sizeof(blockSize)/sizeof(blockSize[0]);
    int n = sizeof(processSize)/sizeof(processSize[0]);

    bestFit(blockSize, m, processSize, n);
    return 0 ;
}
```

Output



```
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ subl .
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ g++ -o worst worst_fit.cpp
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ ./worst

Process No.     Process Size    Block no.
 1              63              4
 2              27              2
 3              44              2
 4              122             4
 5              227             NA
 6              69              3
 7              96              2
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ g++ -o first first_fit.cpp

jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ ./first

Process No.     Process Size    Block no.
 1              63              1
 2              27              1
 3              44              2
 4              122             2
 5              227             4
 6              69              3
 7              96              NA
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ g++ -o best best_fit.cpp
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$ ./best

Process No.     Process Size    Block no.
 1              63              1
 2              27              1
 3              44              3
 4              122             2
 5              227             4
 6              69              2
 7              96              3
jkt@jkt-pc:/media/jkt/New Volume1/OS_LAB/17BCE7106/Memory Mangement$
```