Assignment 1:

Input:

Recursion:

```
public class Fibonacci_recursion {

          static int fib(int n)
   {
      if (n <= 1)
         return n;
      return fib(n - 1) + fib(n - 2);
   }
   public static void main(String args[])
   {
      int n = 9;
     System.out.println("DAA Assignment 1");
                          System.out.println();
                          System.out.println();

      System.out.println(fib(n));
   }
}
```

Time complexity: T(2^N)
Space Complexity: T(N)

Non -Recursion:

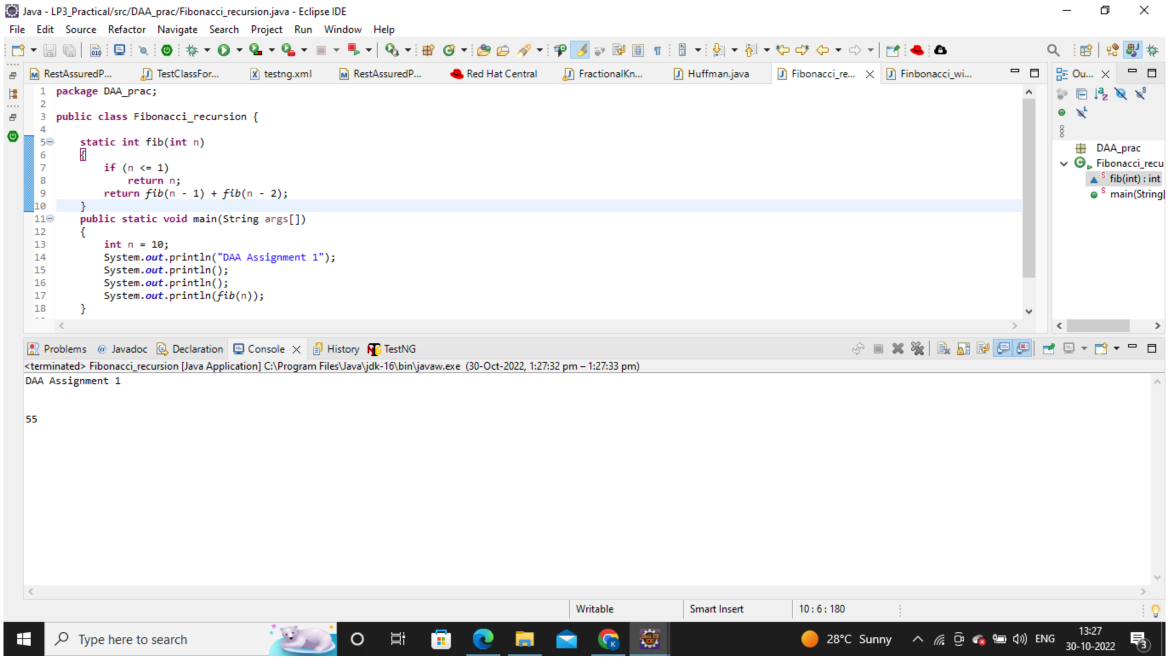
```
public class Fibonacci_wothout_rec {

          public static void main(String args[]) {
                    System.out.println("DAA Assignment 1");
                          System.out.println();
                          System.out.println();

                    fibonacci(10);
                    }
          public static void fibonacci(int number)
          {
                    for(int i=0; i<=number; i++)
          {
                              System.out.print(getFibonacci(i) + " ");
          }
                    }
          public static int getFibonacci(int n)
          { if (n == 0)
          { return 0; }
          if (n == 1)
          { return 1;
          }
```

```
    int first = 0;
    int second = 1;
    int nth = 1;
    for (int i = 2; i <= n; i++)
    { nth = first + second;
    first = second;
    second = nth; }
    return nth;
    }

}
```

Time complexity: T(N)
Space Complexity: O(N)

Output:

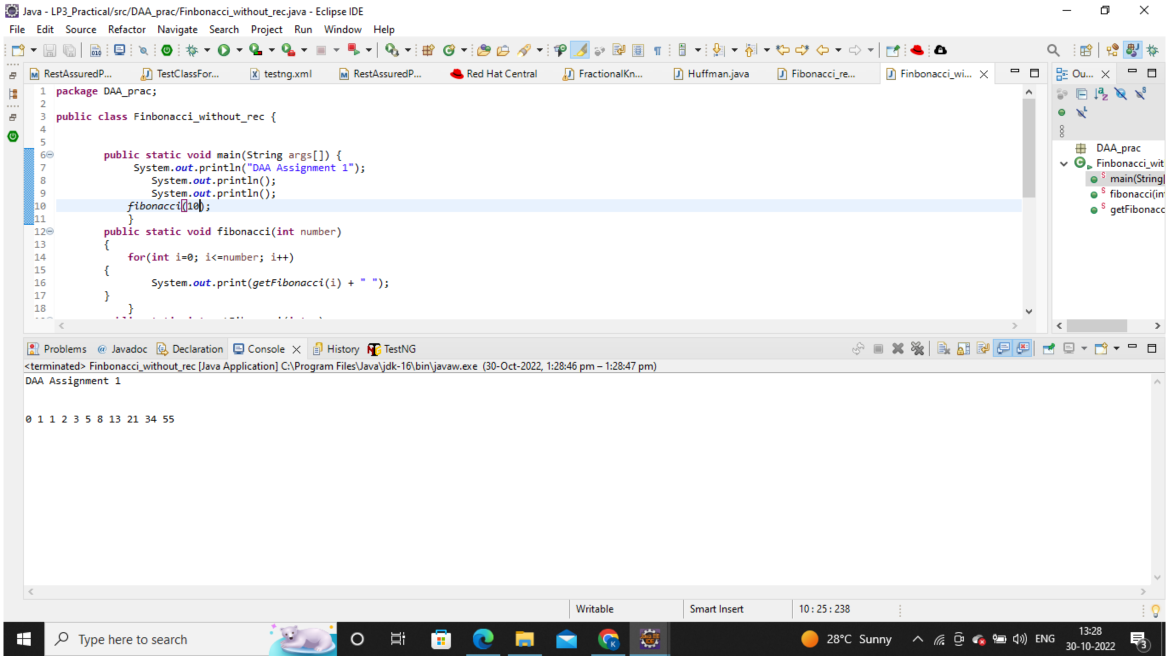
Recursion:


Java - LP3_Practical/src/DAA_prac/Fibonacci_recursion.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RestAssuredP... TestClassFor... testng.xml RestAssuredP... Red Hat Central FractionalKn... Huffman.java Fibonacci_re... Finbonacci_wi...
Problems Javadoc Declaration Console History TestNG
Writable Smart Insert 10 : 6 : 180
Type here to search
28°C Sunny ENG 13:27 30-10-2022

```
package DAA_prac;

public class Fibonacci_recursion {

    static int fib(int n)
    {
        if (n <= 1)
            return n;
        return fib(n - 1) + fib(n - 2);
    }
    public static void main(String args[])
    {
        int n = 10;
        System.out.println("DAA Assignment 1");
        System.out.println();
        System.out.println();
        System.out.println(fib(n));
    }
```

```
<terminated> Fibonacci_recursion [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (30-Oct-2022, 1:27:32 pm – 1:27:33 pm)
DAA Assignment 1


55
```

Non -Recursion:


Java - LP3_Practical/src/DAA_prac/Finbonacci_without_rec.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RestAssuredP... TestClassFor... testng.xml RestAssuredP... Red Hat Central FractionalKn... Huffman.java Fibonacci_re... Finbonacci_wi...
Problems Javadoc Declaration Console History TestNG
Writable Smart Insert 10 : 25 : 238
Type here to search
28°C Sunny ENG 13:28 30-10-2022

```
package DAA_prac;

public class Finbonacci_without_rec {


        public static void main(String args[]) {
             System.out.println("DAA Assignment 1");
                System.out.println();
                System.out.println();
            fibonacci(10);
            }
        public static void fibonacci(int number)
        {
            for(int i=0; i<=number; i++)
        {
                System.out.print(getFibonacci(i) + " ");
        }
            }
```

```
<terminated> Finbonacci_without_rec [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (30-Oct-2022, 1:28:46 pm – 1:28:47 pm)
DAA Assignment 1


0 1 1 2 3 5 8 13 21 34 55
```

Assignment 2

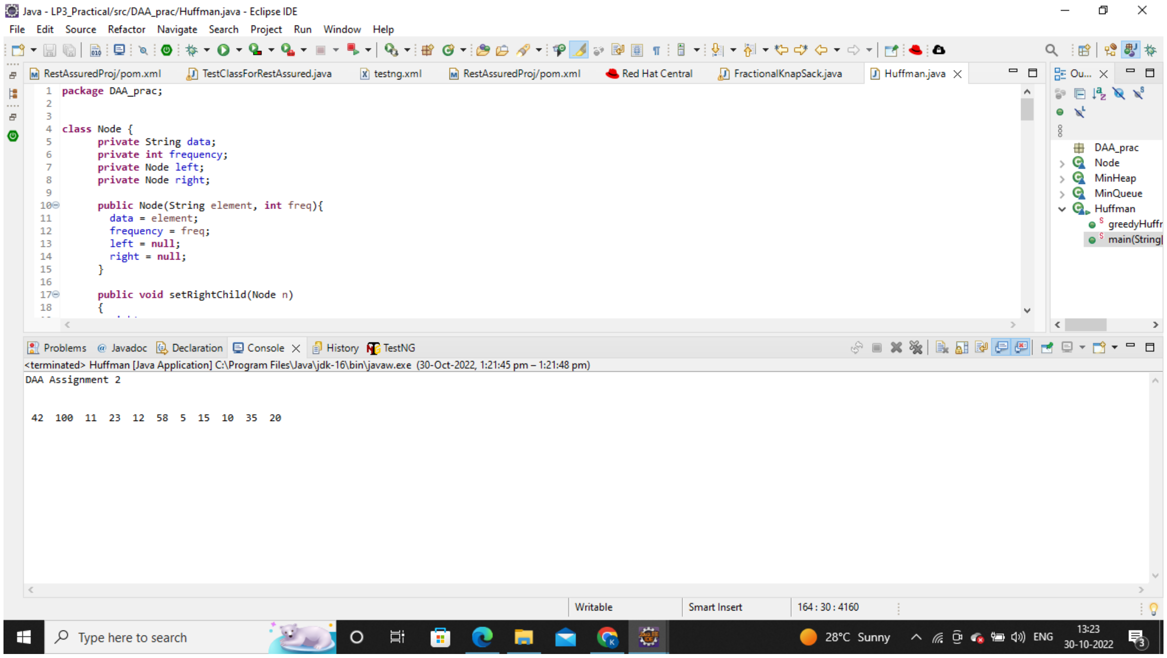
Input:

```
package Day_36;
class Node {
        private String data;
        private int frequency;
        private Node left;
        private Node right;

        public Node(String element, int freq){
          data = element;
          frequency = freq;
          left = null;
          right = null;
        }

        public void setRightChild(Node n)
        {
          right = n;
        }

        public void setLeftChild(Node n){
          left = n;
        }

        public Node getRightChild(){
          return right;
        }

        public Node getLeftChild(){
          return left;
        }

        public String getData(){
          return data;
        }

        public int getFrequency(){
          return frequency;
        }

        public static int getLeftChildIndex(int index) {
```

```
  if(((2*index) <= MinHeap.heapSize) && (index >= 1)) {
   return 2*index;
  }
  return -1;
 }

 public static int getRightChildIndex(int index) {
  if((((2*index)+1) <= MinHeap.heapSize) && (index >= 1)) {
   return (2*index)+1;
  }
  return -1;
 }

 public static int getParentIndex(int index){
  if((index > 1 && (index <= MinHeap.heapSize))) {
   return index/2;
  }
  return -1;
 }

 public static void inorder(Node root) {
  if(root != null) {
   inorder(root.getLeftChild());
   System.out.print(" "+root.getFrequency()+" ");
   inorder(root.getRightChild());
  }
 }
}

class MinHeap {
 public static int heapSize = 0;
 public static final int heapArraySize = 100;
 public static final int INF = 100000;

 public static void minHeapify(Node A[], int index) {
  int leftChildIndex = Node.getLeftChildIndex(index);
  int rightChildIndex = Node.getRightChildIndex(index);

  int smallest = index;

  if ((leftChildIndex <= MinHeap.heapSize) && (leftChildIndex>0)) {
   if (A[leftChildIndex].getFrequency() < A[smallest].getFrequency()) {
    smallest = leftChildIndex;
   }
  }

  if ((rightChildIndex <= MinHeap.heapSize) && (rightChildIndex>0)) {
   if (A[rightChildIndex].getFrequency() < A[smallest].getFrequency()) {
    smallest = rightChildIndex;
```

```
      }
    }

    // smallest is not the node, node is not a heap
    if (smallest != index) {
      Node temp;
      temp = A[index];
      A[index] = A[smallest];
      A[smallest] = temp;
      minHeapify(A, smallest);
    }
  }
}

class MinQueue {

  public static void insert(Node A[], Node a, int key) {
    MinHeap.heapSize++;
    A[MinHeap.heapSize] = a;
    int index = MinHeap.heapSize;
    while((index>1) && (A[Node.getParentIndex(index)].getFrequency() > a.getFrequency())) {
      Node temp;
      temp = A[index];
      A[index] = A[Node.getParentIndex(index)];
      A[Node.getParentIndex(index)] = temp;
      index = Node.getParentIndex(index);
    }
  }

  public static Node[] buildQueue(Node c[], int size) {
    Node[] a = new Node[MinHeap.heapArraySize];
    for(int i=0; i<size; i++) {
      MinQueue.insert(a, c[i], c[i].getFrequency());
    }
    return a;
  }

  public static Node extractMin(Node A[]) {
    Node minm = A[1];
    A[1] = A[MinHeap.heapSize];
    MinHeap.heapSize--;
    MinHeap.minHeapify(A, 1);
    return minm;
  }
}

class Huffman {
  public static Node greedyHuffmanCode(Node C[]) {
    Node[] minQueue = MinQueue.buildQueue(C, 6);
    while(MinHeap.heapSize > 1) {
      Node h = MinQueue.extractMin(minQueue);
      Node i = MinQueue.extractMin(minQueue);
      Node z = new Node("NONE", h.getFrequency()+i.getFrequency());
      z.setLeftChild(h);
      z.setRightChild(i);
      MinQueue.insert(minQueue, z, z.getFrequency());
    }
    return MinQueue.extractMin(minQueue);
  }

  public static void main(String[] args) {
    Node a = new Node("a", 42);
    Node b = new Node("b", 20);
    Node c = new Node("c", 5);
    Node d = new Node("d", 10);
    Node e = new Node("e", 11);
    Node f = new Node("f", 12);
  System.out.println("DAA Assignment 2");
                     System.out.println();
                     System.out.println();

    Node[] C = {a, b, c, d, e ,f};

    Node z = Huffman.greedyHuffmanCode(C);
    Node.inorder(z);
    System.out.println("");
  }
}
```

Output:



```java
package DAA_prac;


class Node {
      private String data;
      private int frequency;
      private Node left;
      private Node right;

      public Node(String element, int freq){
        data = element;
        frequency = freq;
        left = null;
        right = null;
      }

      public void setRightChild(Node n)
      {
```

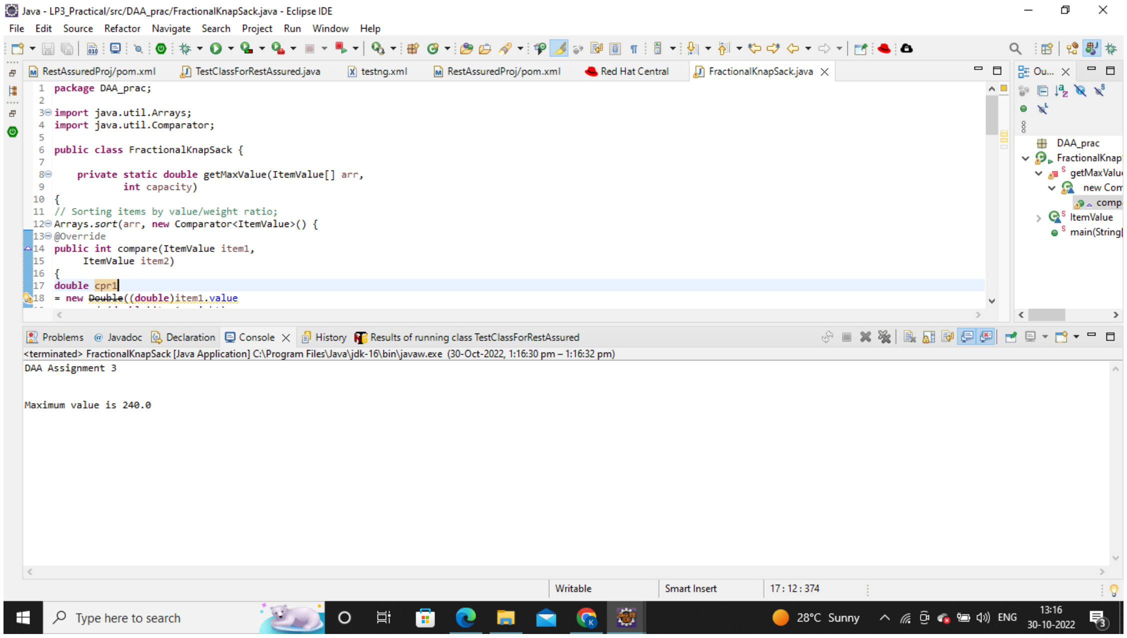
```
<terminated> Huffman [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe  (30-Oct-2022, 1:21:45 pm – 1:21:48 pm)
DAA Assignment 2


 42  100  11  23  12  58  5  15  10  35  20
```

Assignment 3

Input:

```
package DAA_prac;

import java.util.Arrays;
import java.util.Comparator;

public class FractionalKnapSack {

        private static double getMaxValue(ItemValue[] arr,
       int capacity)
{
// Sorting items by value/weight ratio;
Arrays.sort(arr, new Comparator<ItemValue>() {
@Override
public int compare(ItemValue item1,
   ItemValue item2)
{
double cpr1
= new Double((double)item1.value
    / (double)item1.weight);
double cpr2
= new Double((double)item2.value
    / (double)item2.weight);

if (cpr1 < cpr2)
return 1;
else
return -1;
}
});

double totalValue = 0d;

for (ItemValue i : arr) {

int curWt = (int)i.weight;
int curVal = (int)i.value;

if (capacity - curWt >= 0) {

// this weight can be picked while
capacity = capacity - curWt;
totalValue += curVal;
}
else {

// Item cant be picked whole
double fraction
= ((double)capacity / (double)curWt);
totalValue += (curVal * fraction);
capacity
= (int)(capacity - (curWt * fraction));
break;
}
}

return totalValue;
}

// Item value class
static class ItemValue {

int value, weight;

// Item value function
public ItemValue(int val, int wt)
{
this.weight = wt;
this.value = val;
}
}

// Driver code
public static void main(String[] args)
{

ItemValue[] arr = { new ItemValue(60, 10),
  new ItemValue(100, 20),
  new ItemValue(120, 30) };

int capacity = 50;

double maxValue = getMaxValue(arr, capacity);

// Function call
System.out.println("DAA Assignment 3                                          ");
System.out.println();
System.out.println();
System.out.println("Maximum value is "+maxValue);
}

}
```

Output:



Java - LP3_Practical/src/DAA_prac/FractionalKnapSack.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RestAssuredProj/pom.xml
TestClassForRestAssured.java
testng.xml
RestAssuredProj/pom.xml
Red Hat Central
FractionalKnapSack.java
Ou...
DAA_prac
FractionalKnap
getMaxValu
new Com
comp
ItemValue
main(String[

```java
package DAA_prac;

import java.util.Arrays;
import java.util.Comparator;

public class FractionalKnapSack {

    private static double getMaxValue(ItemValue[] arr,
            int capacity)
{
// Sorting items by value/weight ratio;
Arrays.sort(arr, new Comparator<ItemValue>() {
@Override
public int compare(ItemValue item1,
     ItemValue item2)
{
double cpr1
= new ~~Double~~((double)item1.value
```

Problems
Javadoc
Declaration
Console
History
Results of running class TestClassForRestAssured

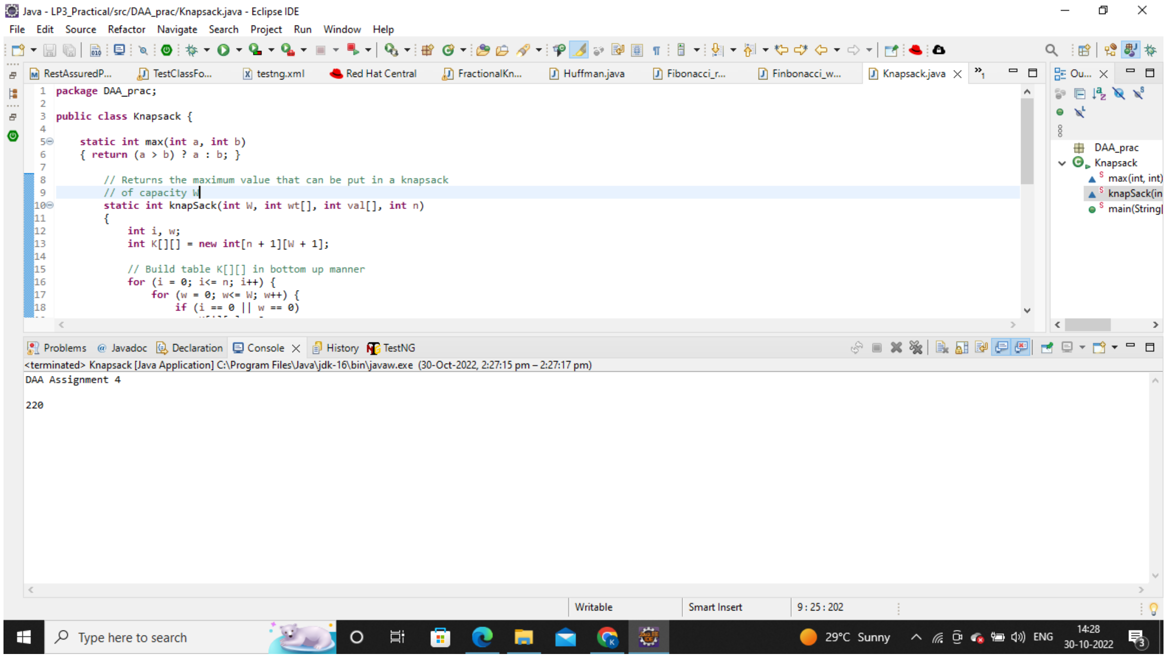
```
<terminated> FractionalKnapSack [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe  (30-Oct-2022, 1:16:30 pm – 1:16:32 pm)
DAA Assignment 3


Maximum value is 240.0
```

Assignment no 4

Input:

```
package DAA_prac;

public class Knapsack {

        static int max(int a, int b)
        { return (a > b) ? a : b; }

          // Returns the maximum value that can be put in a knapsack
          // of capacity W
          static int knapSack(int W, int wt[], int val[], int n)
          {
             int i, w;
             int K[][] = new int[n + 1][W + 1];

             // Build table K[][] in bottom up manner
             for (i = 0; i<= n; i++) {
                for (w = 0; w<= W; w++) {
                   if (i == 0 || w == 0)
                      K[i][w] = 0;
                   else if (wt[i - 1]<= w)
                      K[i][w] = max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
                   else
                      K[i][w] = K[i - 1][w];
                }
             }

             return K[n][W];
          }

          // Driver program to test above function
          public static void main(String args[])
          {
             int val[] = new int[] { 60, 100, 120 };
             int wt[] = new int[] { 10, 20, 30 };
             int W = 50;
             int n = val.length;
             System.out.println("DAA Assignment 4");
             System.out.println();

             System.out.println(knapSack(W, wt, val, n));
          }

}
```

Output:


Java - LP3_Practical/src/DAA_prac/Knapsack.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RestAssuredP... TestClassFo... testng.xml Red Hat Central FractionalKn... Huffman.java Fibonacci_r... Finbonacci_w... Knapsack.java
Ou...
DAA_prac
Knapsack
max(int, int)
knapSack(in
main(String[
Problems Javadoc Declaration Console History TestNG
Writable Smart Insert 9 : 25 : 202
Type here to search
29°C Sunny
ENG
14:28
30-10-2022

```java
package DAA_prac;

public class Knapsack {

    static int max(int a, int b)
    { return (a > b) ? a : b; }

        // Returns the maximum value that can be put in a knapsack
        // of capacity W
        static int knapSack(int W, int wt[], int val[], int n)
        {
            int i, w;
            int K[][] = new int[n + 1][W + 1];

            // Build table K[][] in bottom up manner
            for (i = 0; i<= n; i++) {
                for (w = 0; w<= W; w++) {
                    if (i == 0 || w == 0)
```

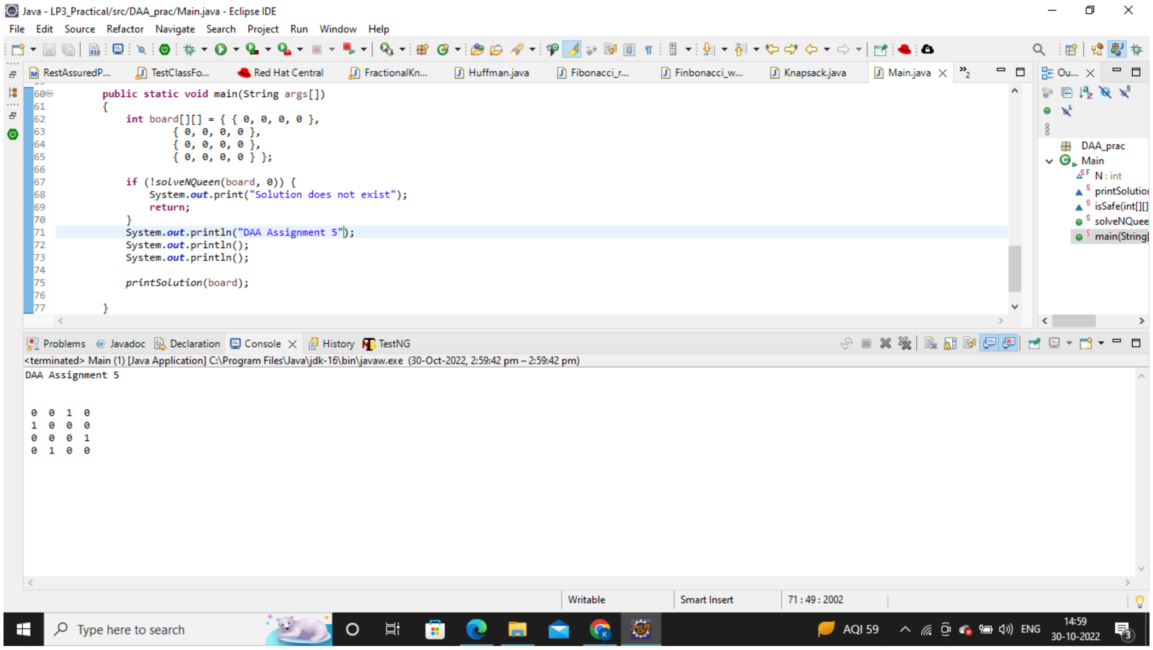
```
<terminated> Knapsack [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (30-Oct-2022, 2:27:15 pm – 2:27:17 pm)
DAA Assignment 4

220
```

Assignment no 5

Input:

```
package DAA_prac;

public class Main {


        static final int N = 4;

       // print the final solution matrix
        static void printSolution(int board[][])
        {
            for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++)
                    System.out.print(" " + board[i][j]
                            + " ");
                System.out.println();
            }
        }

        // function to check whether the position is safe or not
        static boolean isSafe(int board[][], int row, int col)
        {
            int i, j;
            for (i = 0; i < col; i++)
                if (board[row][i] == 1)
                    return false;


            for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
                if (board[i][j] == 1)
                    return false;

            for (i = row, j = col; j >= 0 && i < N; i++, j--)
                if (board[i][j] == 1)
                    return false;

            return true;
        }

        // The function that solves the problem using backtracking
        public static boolean solveNQueen(int board[][], int col)
        {
            if (col >= N)
                return true;
        for (int i = 0; i < N; i++) {
            //if it is safe to place the queen at position i,col -> place it
            if (isSafe(board, i, col)) {
                board[i][col] = 1;

                if (solveNQueen(board, col + 1))
                    return true;

                //backtrack if the above condition is false
                board[i][col] = 0;
            }
        }
        return false;
    }

    public static void main(String args[])
    {
        int board[][] = { { 0, 0, 0, 0 },
                { 0, 0, 0, 0 },
                { 0, 0, 0, 0 },
                { 0, 0, 0, 0 } };

        if (!solveNQueen(board, 0)) {
            System.out.print("Solution does not exist");
            return;
        }
        System.out.println("DAA Assignment 5");
        System.out.println();
        System.out.println();

        printSolution(board);

    }
}
```

Output:


Java - LP3_Practical/src/DAA_prac/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RestAssuredP... TestClassFo... Red Hat Central FractionalKn... Huffman.java Fibonacci_r... Finbonacci_w... Knapsack.java Main.java
Ou...
DAA_prac
Main
N : int
printSolutio
isSafe(int[][]
solveNQuee
main(String
Problems Javadoc Declaration Console History TestNG
Writable
Smart Insert
71 : 49 : 2002
Type here to search
AQI 59
ENG
14:59
30-10-2022

```java
    public static void main(String args[])
    {
        int board[][] = { { 0, 0, 0, 0 },
                { 0, 0, 0, 0 },
                { 0, 0, 0, 0 },
                { 0, 0, 0, 0 } };

        if (!solveNQueen(board, 0)) {
            System.out.print("Solution does not exist");
            return;
        }
        System.out.println("DAA Assignment 5");
        System.out.println();
        System.out.println();

        printSolution(board);

    }
```

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe  (30-Oct-2022, 2:59:42 pm – 2:59:42 pm)
DAA Assignment 5


 0  0  1  0
 1  0  0  0
 0  0  0  1
 0  1  0  0
```

Mini project

Input:

Matrix multiplication

```
package DAA_prac;

public class Matrix_multiplication {

        public static void main(String[] args) {
                        //creating two matrices
                   System.out.println("......MINI PROJECT.....");
                   System.out.println();
                        int a[][]={{1,1,1},{2,2,2},{3,3,3}};
                        int b[][]={{1,1,1},{2,2,2},{3,3,3}};
                        int c[][]=new int[3][3];
                        for(int i=0;i<3;i++){
                        for(int j=0;j<3;j++){
                        c[i][j]=0;
                        for(int k=0;k<3;k++)
                        {
                        c[i][j]+=a[i][k]*b[k][j];
                        }
                        System.out.print(c[i][j]+" ");
                        }
                        System.out.println();
                        }
                        }

        }
```

Time complexity: O(N3)
Auxiliary Space: O(M1 * N2)

Multithreaded matrix multiplication:

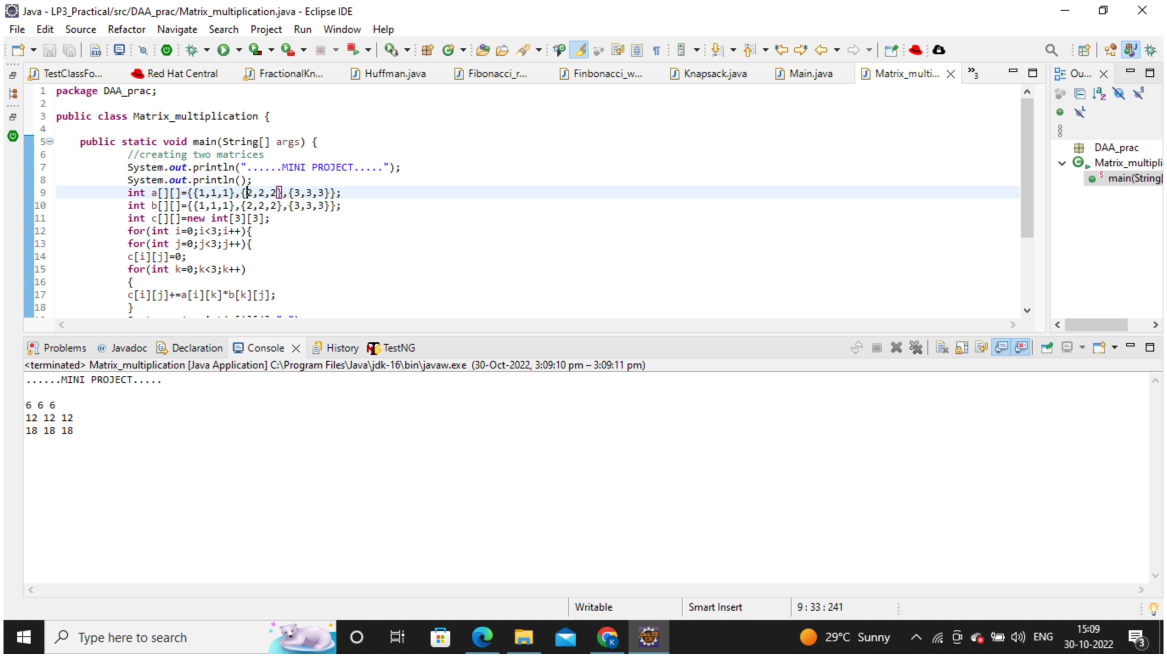
```
package DAA_prac;



        import java.io.BufferedReader;
        import java.io.InputStreamReader;

        public class Multi_matrix extends Thread{
                 static int in1[][];
           static int in2[][];
           static int out[][];
           static int n=2;
```

```
int row;
Multi_matrix(int i)
{
    row=i;
    this.start();
}
public void run()
{
    int i,j;
    for(i=0;i<n;i++)
    {
        out[row][i]=0;
        for(j=0;j<n;j++)
            out[row][i]=out[row][i]+in1[row][j]*in2[j][i];
    }
}
public static void main(String args[])
{
    int i,j;
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.print("Enter the order of Matrix : ");
    try
    {
      n=Integer.parseInt(br.readLine());
    }catch(Exception e){}
    in1=new int[n][n];
    in2=new int[n][n];
    out=new int[n][n];
    System.out.println("Enter the First Matrix : ");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            try
            {
                in1[i][j]=Integer.parseInt(br.readLine());
            }catch(Exception e){}
        }
    }
    System.out.println("Enter the Second Matrix : ");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            try
            {
                in2[i][j]=Integer.parseInt(br.readLine());
            }catch(Exception e){}
        }
```

```
                }
                Multi_matrix mat[]=new Multi_matrix[n];
                for(i=0;i<n;i++)
                        mat[i]=new Multi_matrix(i);
                try
                {
                        for(i=0;i<n;i++)
                                mat[i].join();
                }catch(Exception e){}
                System.out.println("OUTPUT :");
                for(i=0;i<n;i++)
                        for(j=0;j<n;j++)
                                System.out.println(out[i][j]);
        }
}
```

Output:

Matrix multiplication

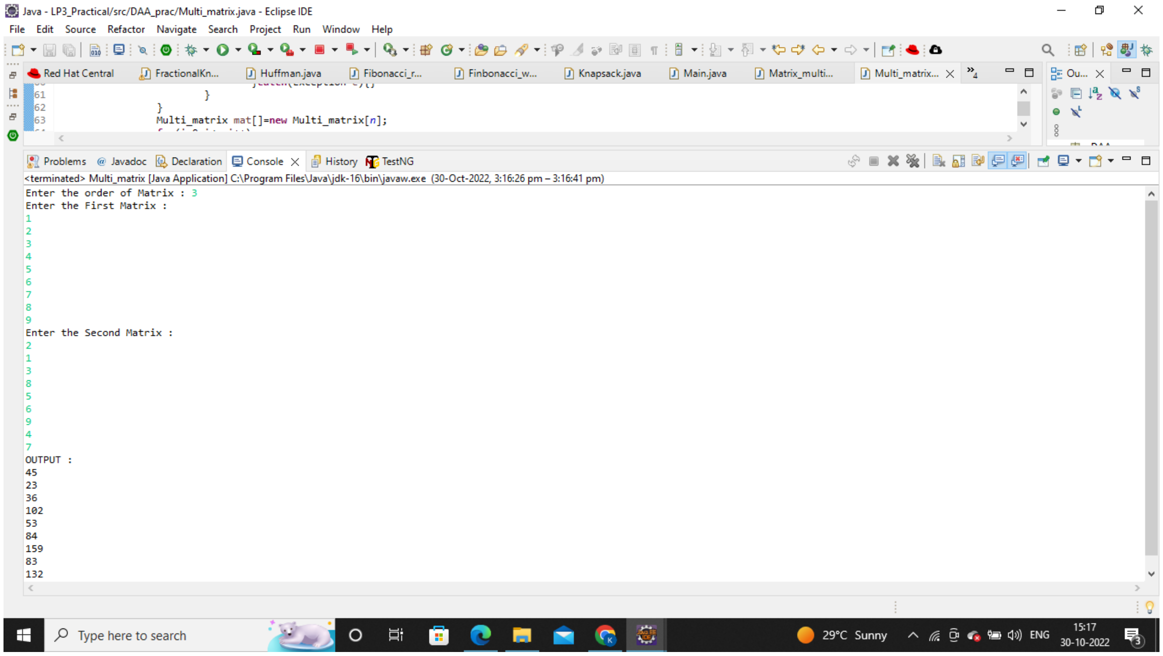


Java - LP3_Practical/src/DAA_prac/Matrix_multiplication.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
TestClassFo... Red Hat Central FractionalKn... Huffman.java Fibonacci_r... Finbonacci_w... Knapsack.java Main.java Matrix_multi...
Ou...
DAA_prac
Matrix_multipli
main(String[
Problems @ Javadoc Declaration Console History TestNG
Writable Smart Insert 9 : 33 : 241
Type here to search
29°C Sunny ENG 15:09 30-10-2022

```
package DAA_prac;

public class Matrix_multiplication {

    public static void main(String[] args) {
            //creating two matrices
            System.out.println("......MINI PROJECT.....");
            System.out.println();
            int a[][]={{1,1,1},{2,2,2},{3,3,3}};
            int b[][]={{1,1,1},{2,2,2},{3,3,3}};
            int c[][]=new int[3][3];
            for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
            c[i][j]=0;
            for(int k=0;k<3;k++)
            {
            c[i][j]+=a[i][k]*b[k][j];
            }
```

```
<terminated> Matrix_multiplication [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe  (30-Oct-2022, 3:09:10 pm – 3:09:11 pm)
......MINI PROJECT.....

6 6 6
12 12 12
18 18 18
```



```
                        }
                }
                Multi_matrix mat[]=new Multi_matrix[n];
```

<terminated> Multi_matrix [Java Application] C:\Program Files\Java\jdk-16\bin\javaw.exe (30-Oct-2022, 3:16:26 pm – 3:16:41 pm)

```
Enter the order of Matrix : 3
Enter the First Matrix :
1
2
3
4
5
6
7
8
9
Enter the Second Matrix :
2
1
3
8
5
6
9
4
7
OUTPUT :
45
23
36
102
53
84
159
83
132
```