

Day 15&16

Task 1: Knapsack Problem

Write a function `int Knapsack(int W, int[] weights, int[] values)` in java that determines the maximum value of items that can fit into a knapsack with a capacity `W`. The function should handle up to 100 items. Find the optimal way to fill the knapsack with the given items to achieve the maximum total value. You must consider that you cannot break items, but have to include them whole.

Program:

```
package Assignments.Day15and16;

public class Task1 {

    public static int knapsack(int W, int[] weights, int[] values) {
        int n = weights.length;
        int [][] dp = new int[n + 1][W + 1];

        for (int i = 0; i <= n; i++) {
            for (int w = 0; w <= W; w++) {
                if (i == 0 || w == 0) {
                    dp[i][w] = 0;
                } else if (weights[i - 1] <= w) {
                    dp[i][w] = Math.max(values[i - 1] + dp[i - 1][w - weights[i - 1]],
dp[i - 1][w]);
                } else {
                    dp[i][w] = dp[i - 1][w];
                }
            }
        }

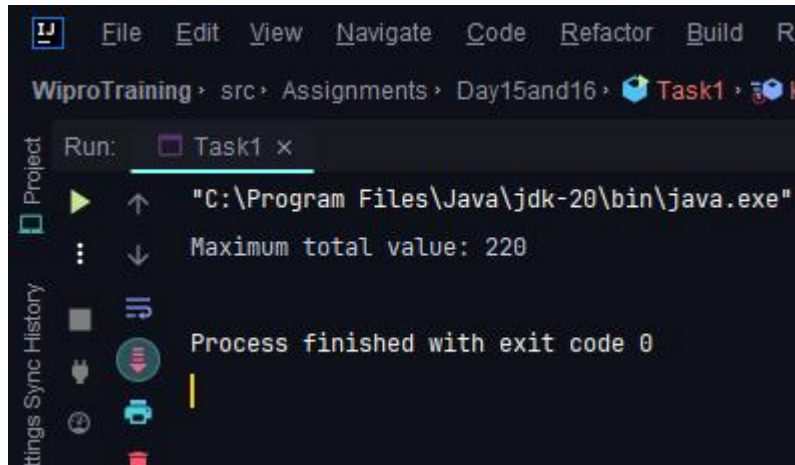
        return dp[n][W];
    }

    public static void main(String[] args) {
        int W = 50;
        int[] weights = {10, 20, 30};
        int[] values = {60, 100, 120};

        int maxTotalValue = knapsack(W, weights, values);
        System.out.println("Maximum total value: " + maxTotalValue);
    }
}
```

```
}  
}
```

Output:



Task 2: Longest Common Subsequence

Implement int LCS(string text1, string text2) to find the length of the longest common subsequence between two strings.

Program:

```
package Assignments.Day15and16;
```

```
public class Task2 {
```

```
    public static int longestCommonSubsequence(String s1, String s2) {
```

```
        int length1 = s1.length();
```

```
        int length2 = s2.length();
```

```
        int[][] dp = new int[length1 + 1][length2 + 1];
```

```
        for (int i = 1; i <= length1; ++i) {
```

```
            for (int j = 1; j <= length2; ++j) {
```

```
                if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
```

```
                    dp[i][j] = dp[i - 1][j - 1] + 1;
```

```
                }
```

```
            } else {
```

```
                dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
```

```
            }
```

```
        }
```

```
    }
```

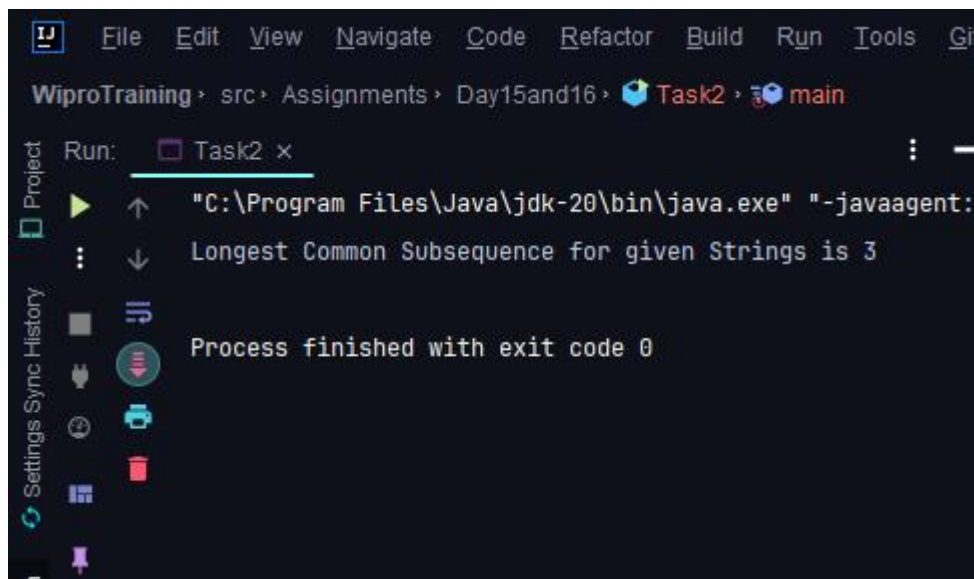
```
    return dp[length1][length2];
```

```
}

public static void main(String[] args) {
    System.out.println("Longest Common Subsequence for given Strings is  
"+longestCommonSubsequence("abcde","abc"));
}

}
```

Output:



The screenshot shows an IDE window with the following content:

- Menu bar: File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git
- Breadcrumb: WiproTraining > src > Assignments > Day15and16 > Task2 > main
- Run configuration: Run: Task2 x
- Run output:
 - "C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:
 - Longest Common Subsequence for given Strings is 3
 - Process finished with exit code 0

