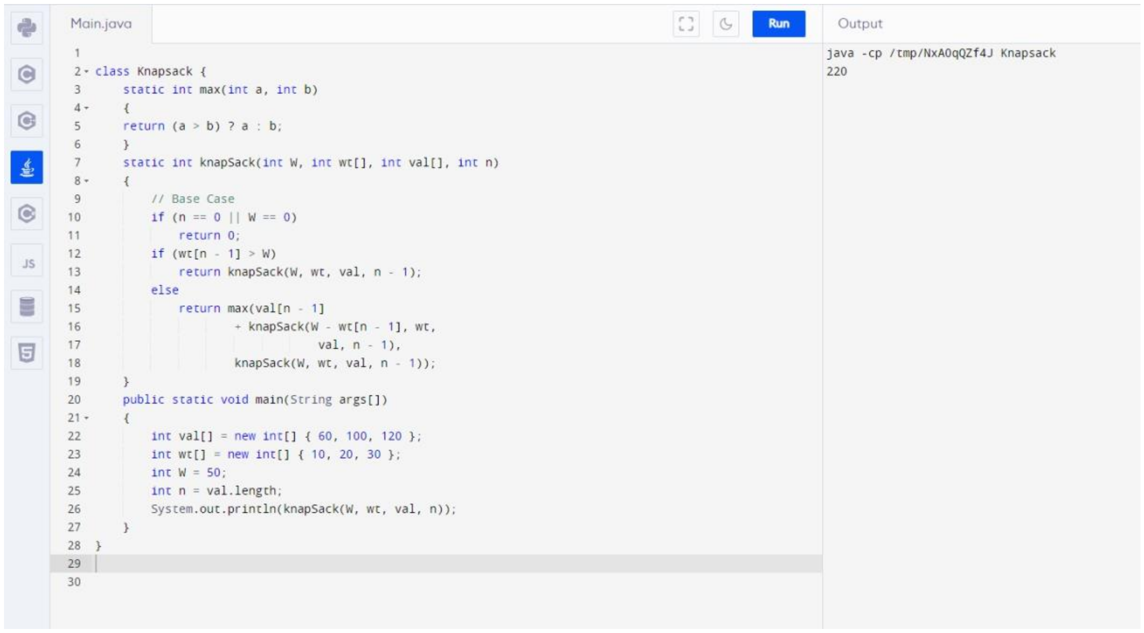
Week 4

Name- Akanksha Yadav

Roll no- 1900290120008

1. 0/1 knapsack

```
class Knapsack {
    static int max(int a, int b)
    {
    return (a > b) ? a : b;
    }
    static int knapSack(int W, int wt[], int val[], int n)
    {
        // Base Case
        if (n == 0 || W == 0)
            return 0;
        if (wt[n - 1] > W)
            return knapSack(W, wt, val, n - 1);
        else
            return max(val[n - 1]
                        + knapSack(W - wt[n - 1], wt,
                                    val, n - 1),
                        knapSack(W, wt, val, n - 1));
    }
    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(knapSack(W, wt, val, n));
    }
}
```

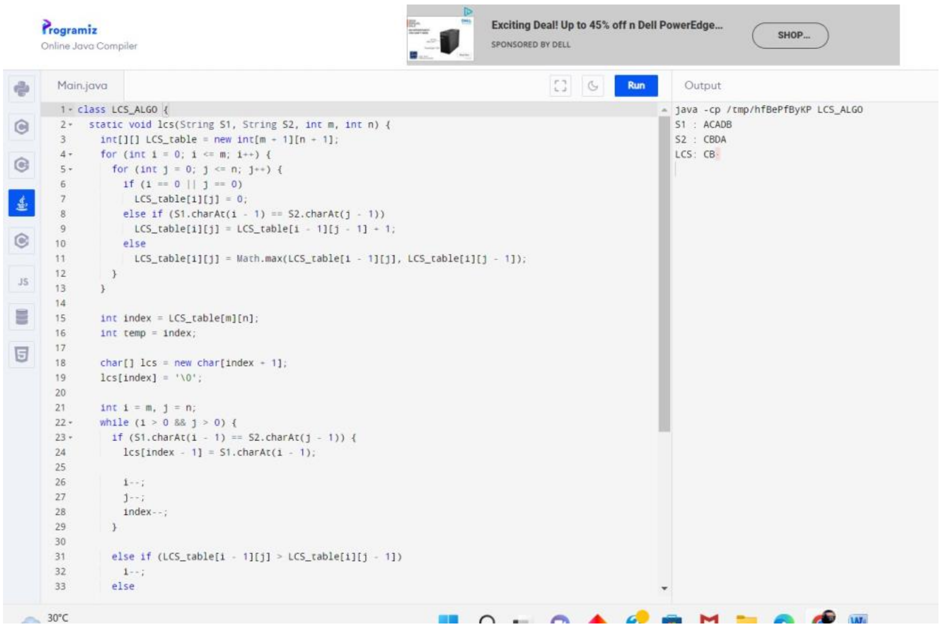


```java

class Knapsack {
    static int max(int a, int b)
    {
    return (a > b) ? a : b;
    }
    static int knapSack(int W, int wt[], int val[], int n)
    {
        // Base Case
        if (n == 0 || W == 0)
            return 0;
        if (wt[n - 1] > W)
            return knapSack(W, wt, val, n - 1);
        else
            return max(val[n - 1]
                    + knapSack(W - wt[n - 1], wt,
                               val, n - 1),
                    knapSack(W, wt, val, n - 1));
    }
    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(knapSack(W, wt, val, n));
    }
}
```

```
java -cp /tmp/NxA0qQZf4J Knapsack
220
```

2. LCS algo

```java
class LCS_ALGO {
  static void lcs(String S1, String S2, int m, int n) {
    int[][] LCS_table = new int[m + 1][n + 1];
    for (int i = 0; i <= m; i++) {
      for (int j = 0; j <= n; j++) {
      if (i == 0 || j == 0)
          LCS_table[i][j] = 0;
        else if (S1.charAt(i - 1) == S2.charAt(j - 1))
        LCS_table[i][j] = LCS_table[i - 1][j - 1] + 1;
        else
          LCS_table[i][j] = Math.max(LCS_table[i - 1][j], LCS_table[i][j - 1]);
      }
    }

    int index = LCS_table[m][n];
    int temp = index;

    char[] lcs = new char[index + 1];
    lcs[index] = '\0';

    int i = m, j = n;
    while (i > 0 && j > 0) {
      if (S1.charAt(i - 1) == S2.charAt(j - 1)) {
        lcs[index - 1] = S1.charAt(i - 1);
        i--;
        j--;
        index--;
      }

      else if (LCS_table[i - 1][j] > LCS_table[i][j - 1])
        i--;
      else
        j--;
    }
    System.out.print("S1 : " + S1 + "\nS2 : " + S2 + "\nLCS: ");
    for (int k = 0; k <= temp; k++)
      System.out.print(lcs[k]);
    System.out.println("");
  }

  public static void main(String[] args) {
    String S1 = "ACADB";
    String S2 = "CBDA";
    int m = S1.length();
    int n = S2.length();
    lcs(S1, S2, m, n);
  }
}
```

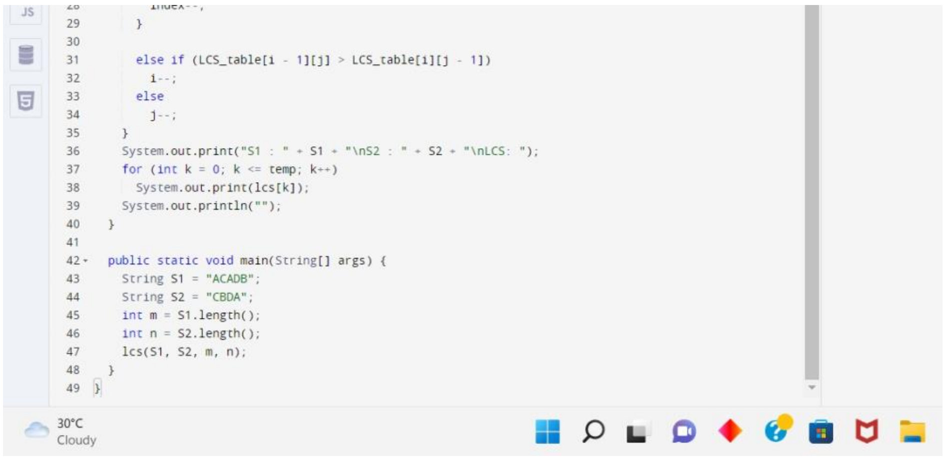


Programiz
Online Java Compiler

Exciting Deal! Up to 45% off n Dell PowerEdge...
SPONSORED BY DELL

```
class LCS_ALGO {
  static void lcs(String S1, String S2, int m, int n) {
    int[][] LCS_table = new int[m + 1][n + 1];
    for (int i = 0; i <= m; i++) {
      for (int j = 0; j <= n; j++) {
        if (i == 0 || j == 0)
          LCS_table[i][j] = 0;
        else if (S1.charAt(i - 1) == S2.charAt(j - 1))
          LCS_table[i][j] = LCS_table[i - 1][j - 1] + 1;
        else
          LCS_table[i][j] = Math.max(LCS_table[i - 1][j], LCS_table[i][j - 1]);
      }
    }

    int index = LCS_table[m][n];
    int temp = index;

    char[] lcs = new char[index + 1];
    lcs[index] = '\0';

    int i = m, j = n;
    while (i > 0 && j > 0) {
      if (S1.charAt(i - 1) == S2.charAt(j - 1)) {
        lcs[index - 1] = S1.charAt(i - 1);

        i--;
        j--;
        index--;
      }

      else if (LCS_table[i - 1][j] > LCS_table[i][j - 1])
        i--;
      else
```

```
java -cp /tmp/hfBePfByKP LCS_ALGO
S1 : ACADB
S2 : CBDA
LCS: CB
```

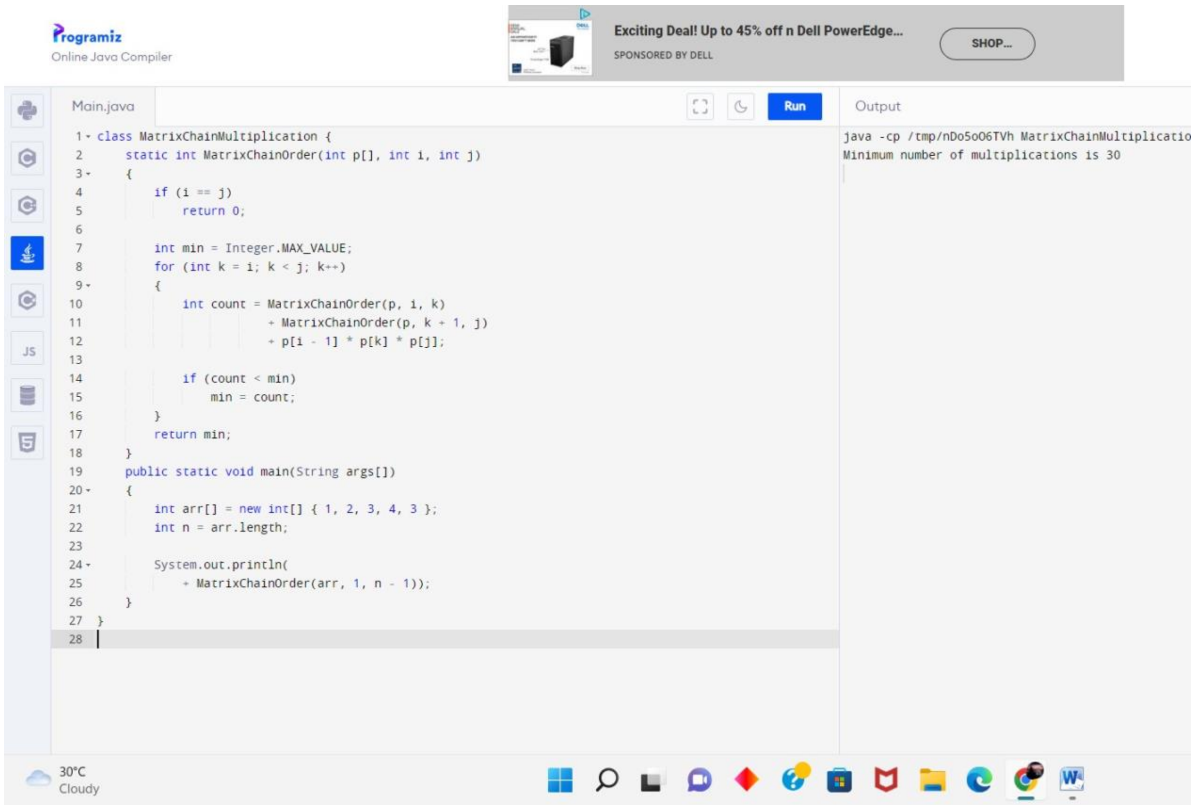


```
            index--;
        }

        else if (LCS_table[i - 1][j] > LCS_table[i][j - 1])
          i--;
        else
          j--;
      }
      System.out.print("S1 : " + S1 + "\nS2 : " + S2 + "\nLCS: ");
      for (int k = 0; k <= temp; k++)
        System.out.print(lcs[k]);
      System.out.println("");
    }

    public static void main(String[] args) {
      String S1 = "ACADB";
      String S2 = "CBDA";
      int m = S1.length();
      int n = S2.length();
      lcs(S1, S2, m, n);
    }
}
```

30°C
Cloudy

3. MCM algo

```
class MatrixChainMultiplication {
    static int MatrixChainOrder(int p[], int i, int j)
    {
        if (i == j)
            return 0;

        int min = Integer.MAX_VALUE;
        for (int k = i; k < j; k++)
        {
            int count = MatrixChainOrder(p, i, k)
                        + MatrixChainOrder(p, k + 1, j)
                        + p[i - 1] * p[k] * p[j];

            if (count < min)
                min = count;
        }
        return min;
    }
    public static void main(String args[])
    {
        int arr[] = new int[] { 1, 2, 3, 4, 3 };
        int n = arr.length;

        System.out.println(
            + MatrixChainOrder(arr, 1, n - 1));
    }
}
```



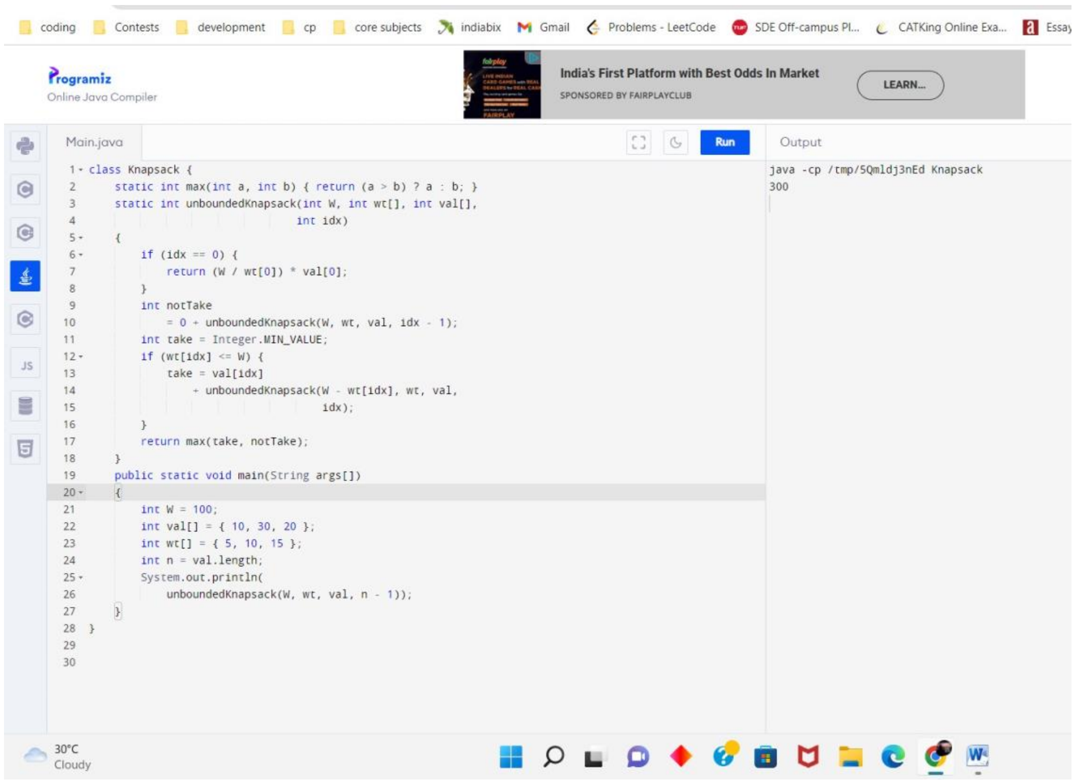
```
class MatrixChainMultiplication {
    static int MatrixChainOrder(int p[], int i, int j)
    {
        if (i == j)
            return 0;

        int min = Integer.MAX_VALUE;
        for (int k = i; k < j; k++)
        {
            int count = MatrixChainOrder(p, i, k)
                        + MatrixChainOrder(p, k + 1, j)
                        + p[i - 1] * p[k] * p[j];

            if (count < min)
                min = count;
        }
        return min;
    }
    public static void main(String args[])
    {
        int arr[] = new int[] { 1, 2, 3, 4, 3 };
        int n = arr.length;

        System.out.println(
            + MatrixChainOrder(arr, 1, n - 1));
    }
}
```

```
java -cp /tmp/nDo5o06TVh MatrixChainMultiplicatio
Minimum number of multiplications is 30
```

4. Unbounded knapsack

```
class Knapsack {
        static int max(int a, int b) { return (a > b) ? a : b; }
        static int unboundedKnapsack(int W, int wt[], int val[],
                                                                int idx)
        {
                if (idx == 0) {
                        return (W / wt[0]) * val[0];
                }
                int notTake
                        = 0 + unboundedKnapsack(W, wt, val, idx - 1);
                int take = Integer.MIN_VALUE;
                if (wt[idx] <= W) {
                        take = val[idx]
                                + unboundedKnapsack(W - wt[idx], wt, val,
                                                                        idx);
                }
                return max(take, notTake);
        }
        public static void main(String args[])
        {
                int W = 100;
                int val[] = { 10, 30, 20 };
                int wt[] = { 5, 10, 15 };
```

```
		int n = val.length;
		System.out.println(
			unboundedKnapsack(W, wt, val, n - 1));
	}
}
```



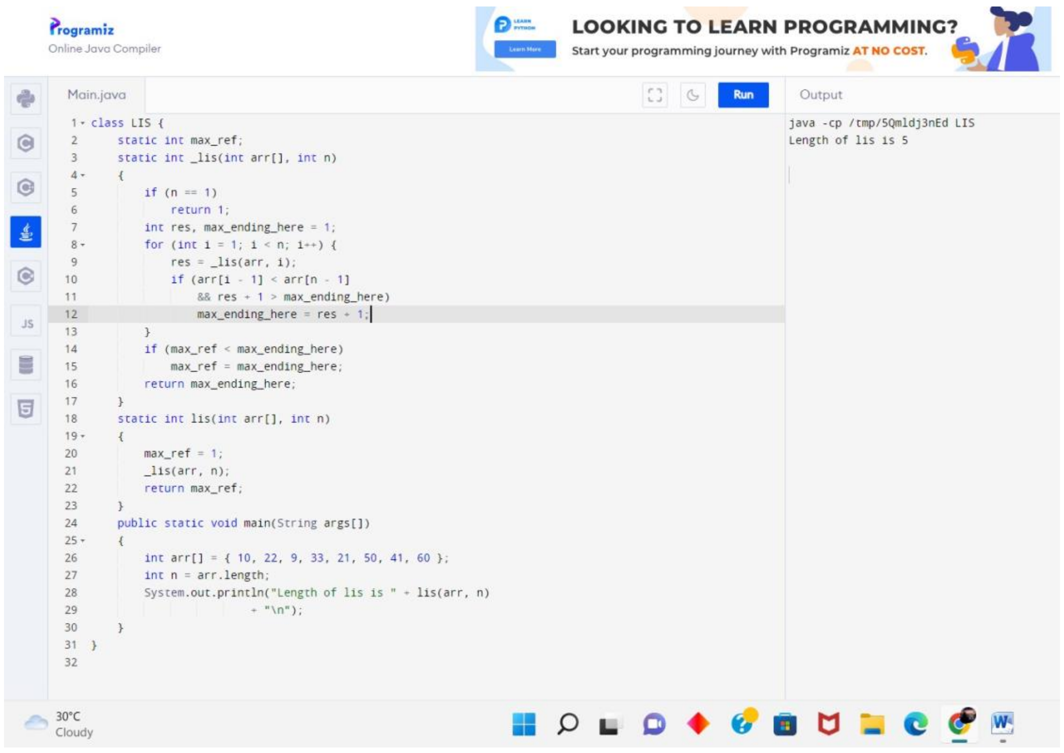
```
class Knapsack {
    static int max(int a, int b) { return (a > b) ? a : b; }
    static int unboundedKnapsack(int W, int wt[], int val[],
                                 int idx)
    {
        if (idx == 0) {
            return (W / wt[0]) * val[0];
        }
        int notTake
            = 0 + unboundedKnapsack(W, wt, val, idx - 1);
        int take = Integer.MIN_VALUE;
        if (wt[idx] <= W) {
            take = val[idx]
                + unboundedKnapsack(W - wt[idx], wt, val,
                                    idx);
        }
        return max(take, notTake);
    }
    public static void main(String args[])
    {
        int W = 100;
        int val[] = { 10, 30, 20 };
        int wt[] = { 5, 10, 15 };
        int n = val.length;
        System.out.println(
            unboundedKnapsack(W, wt, val, n - 1));
    }
}
```

```
java -cp /tmp/5Qmldj3nEd Knapsack
300
```

5. LIS Algo

```
class LIS {
	static int max_ref;
	static int _lis(int arr[], int n)
	{
		if (n == 1)
			return 1;
		int res, max_ending_here = 1;
		for (int i = 1; i < n; i++) {
			res = _lis(arr, i);
			if (arr[i - 1] < arr[n - 1]
				&& res + 1 > max_ending_here)
				max_ending_here = res + 1;
		}
		if (max_ref < max_ending_here)
			max_ref = max_ending_here;
		return max_ending_here;
	}
	static int lis(int arr[], int n)
	{
```

```
            max_ref = 1;
            _lis(arr, n);
            return max_ref;
        }
        public static void main(String args[])
        {
            int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };
            int n = arr.length;
            System.out.println("Length of lis is " + lis(arr, n)
                                                   + "\n");
        }
}
```


Programiz
Online Java Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Main.java
Run
Output
30°C
Cloudy

```
class LIS {
    static int max_ref;
    static int _lis(int arr[], int n)
    {
        if (n == 1)
            return 1;
        int res, max_ending_here = 1;
        for (int i = 1; i < n; i++) {
            res = _lis(arr, i);
            if (arr[i - 1] < arr[n - 1]
                && res + 1 > max_ending_here)
                max_ending_here = res + 1;
        }
        if (max_ref < max_ending_here)
            max_ref = max_ending_here;
        return max_ending_here;
    }
    static int lis(int arr[], int n)
    {
        max_ref = 1;
        _lis(arr, n);
        return max_ref;
    }
    public static void main(String args[])
    {
        int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };
        int n = arr.length;
        System.out.println("Length of lis is " + lis(arr, n)
                        + "\n");
    }
}
```

```
java -cp /tmp/5Qmldj3nEd LIS
Length of lis is 5
```