

1. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is  $60 \geq$  and  $< 75$ , then the grade is First Division. If aggregate is  $50 \geq$  and  $< 60$ , then the grade is Second Division. If aggregate is  $40 \geq$  and  $< 50$ , then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

- a) 18, 76, 93, 65
- b) 73, 78, 79, 75
- c) 98, 106, 120, 95
- d) 96, 73, -85, 95
- e) 78, 59.8, 76, 79

The screenshot shows the Programiz Online Java Compiler interface. The code in Main.java is as follows:

```
1 import java.util.Scanner;
2 public class GradeCalculator {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5
6         System.out.print("Enter the marks in Python: ");
7         double python = scanner.nextDouble();
8
9         System.out.print("Enter the marks in C Programming: ");
10        double cProgramming = scanner.nextDouble();
11
12        System.out.print("Enter the marks in Mathematics: ");
13        double mathematics = scanner.nextDouble();
14
15        System.out.print("Enter the marks in Physics: ");
16        double physics = scanner.nextDouble();
17
18        double total = python + cProgramming + mathematics + physics;
19        double aggregate = total / 4;
20
21        System.out.printf("Total = %.0f\n", total);
22        System.out.printf("Aggregate = %.1f\n", aggregate);
23
24        if (aggregate > 75) {
25            System.out.println("DISTINCTION");
26        } else if (aggregate >= 60 && aggregate < 75) {
```

The output window shows the following results for the sample input:

```
java -cp /tmp/g7aCpKfIdi/GradeCalculator
Enter the marks in Python: 89
Enter the marks in C Programming: 97
Enter the marks in Mathematics: 67
Enter the marks in Physics: 90
Total = 343
Aggregate = 85.8
DISTINCTION
=== Code Execution Successful ===
```

2. Write a program to calculate tax given the following conditions:

- a. If income is less than or equal to 1,50,000 then no tax
- b. If taxable income is 1,50,001 – 3,00,000 the charge 10% tax

- c. If taxable income is 3,00,001 – 5,00,000 the charge 20% tax
- d. If taxable income is above 5,00,001 then charge 30% tax

Sample Input:

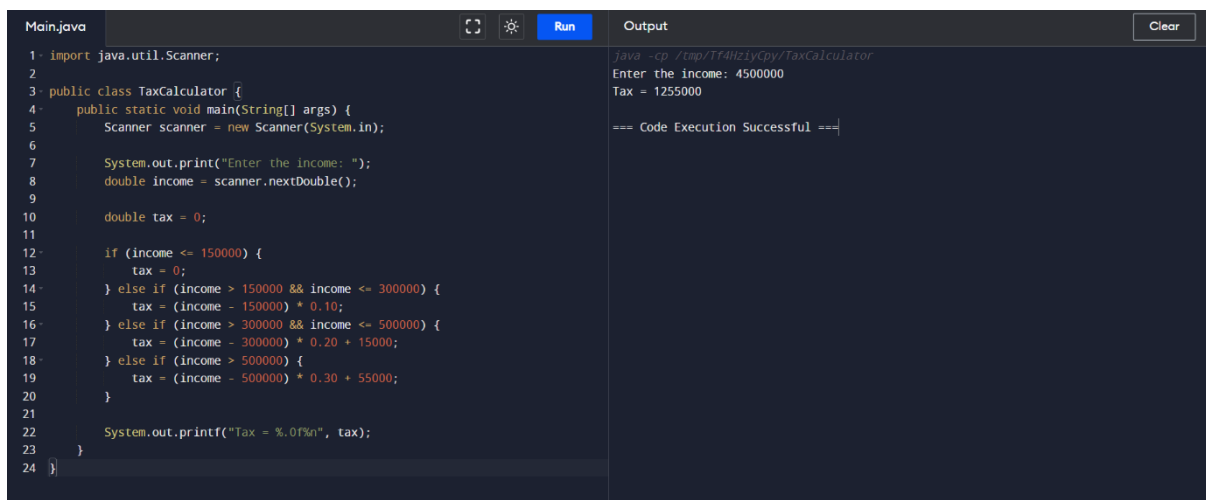
Enter the income:200000

Sample Output:

Tax= 20000

Test cases:

- 1. 400700
- 2. 2789239
- 3. 150000
- 4. 00000
- 5. -125486



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a 'TaxCalculator' class with a 'main' method that uses a 'Scanner' to take input and calculates tax based on income brackets. The output window shows the program was run with an input of 4500000, resulting in a tax of 1255000, and confirms successful execution.

```
1- import java.util.Scanner;
2
3- public class TaxCalculator {
4-     public static void main(String[] args) {
5-         Scanner scanner = new Scanner(System.in);
6
7-         System.out.print("Enter the income: ");
8-         double income = scanner.nextDouble();
9
10        double tax = 0;
11
12        if (income <= 150000) {
13            tax = 0;
14        } else if (income > 150000 && income <= 300000) {
15            tax = (income - 150000) * 0.10;
16        } else if (income > 300000 && income <= 500000) {
17            tax = (income - 300000) * 0.20 + 15000;
18        } else if (income > 500000) {
19            tax = (income - 500000) * 0.30 + 55000;
20        }
21
22        System.out.printf("Tax = %.0f%n", tax);
23    }
24 }
```

Output

```
java -cp ./tmp/TF4HziyCpy/TaxCalculator
Enter the income: 4500000
Tax = 1255000

=== Code Execution Successful ===
```

3. Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)

Sample Input:

N = 3

Sample Output:

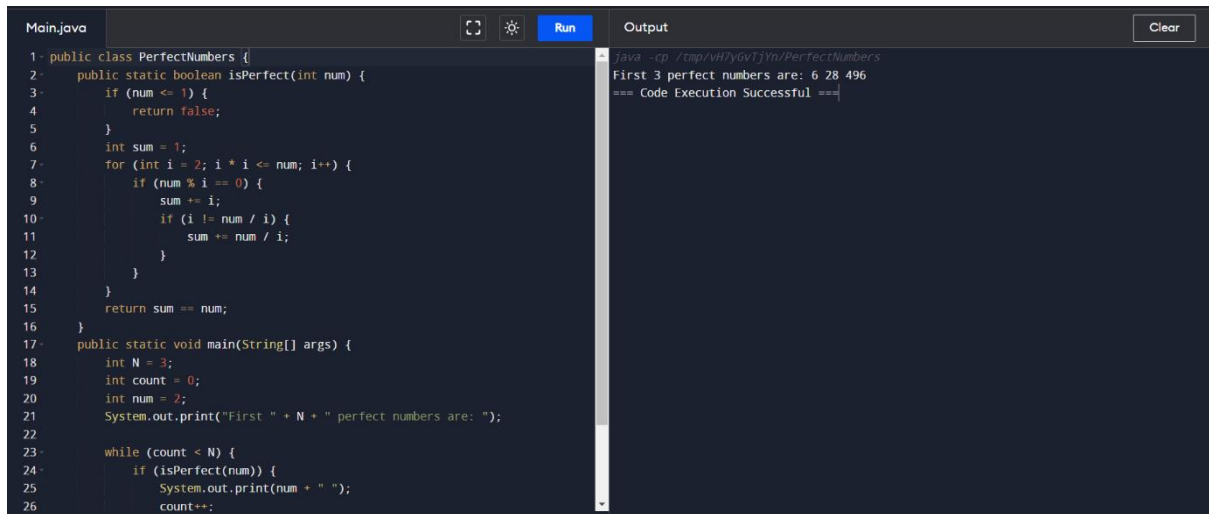
First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

- 1. N = 0
- 2. N = 5
- 3. N = -2

4. N = -5

5. N = 0.2



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'PerfectNumbers' with a static method 'isPerfect' that checks if a number is perfect by summing its divisors. The 'main' method sets N=3, count=0, and num=2, then prints the first N perfect numbers. The output window shows the result: 'First 3 perfect numbers are: 6 28 496' and 'Code Execution Successful'.

```
1- public class PerfectNumbers {
2-     public static boolean isPerfect(int num) {
3-         if (num <= 1) {
4-             return false;
5-         }
6-         int sum = 1;
7-         for (int i = 2; i * i <= num; i++) {
8-             if (num % i == 0) {
9-                 sum += i;
10-                 if (i != num / i) {
11-                     sum += num / i;
12-                 }
13-             }
14-         }
15-         return sum == num;
16-     }
17-     public static void main(String[] args) {
18-         int N = 3;
19-         int count = 0;
20-         int num = 2;
21-         System.out.print("First " + N + " perfect numbers are: ");
22-
23-         while (count < N) {
24-             if (isPerfect(num)) {
25-                 System.out.print(num + " ");
26-                 count++;
```

Output: java -cp /tmp/vH2yGvTjYn/PerfectNumbers  
First 3 perfect numbers are: 6 28 496  
=== Code Execution Successful ===

4. Write a Program to Find the Nth Largest Number in a array.

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases:

1. N = 0

2. N = -5

3. N = 1

4. N = M

5. N = %

```
Main.java
1 public class NthLargestNumber {
2     public static int findNthLargest(int[] arr, int N) {
3         int n = arr.length;
4         if (N <= 0 || N > n) {
5             return -1;
6         }
7         for (int i = 0; i < N; i++) {
8             int maxIndex = i;
9             for (int j = i + 1; j < n; j++) {
10                 if (arr[j] > arr[maxIndex]) {
11                     maxIndex = j;
12                 }
13             }
14             int temp = arr[i];
15             arr[i] = arr[maxIndex];
16             arr[maxIndex] = temp;
17         }
18         return arr[N - 1];
19     }
20     public static void main(String[] args) {
21         int[] arr = {14, 67, 48, 23, 5, 62};
22         int N = 4;
23         int nthLargest = findNthLargest(arr, N);
24         System.out.println(N + "th Largest number: " + nthLargest);
25     }
26 }
```

```
Output
Clear
java -cp /tmp/.JLHeu3gGsh/NthLargestNumber
4th Largest number: 23

=== Code Execution Successful ===
```

5. Write a program to find the number of special characters in the given statement

Sample Input:

Given statement: Modi Birthday @ September 17, #&\$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

```
Main.java
1 public class SpecialCharacterCount {
2
3     public static void main(String[] args) {
4         String statement = "Modi Birthday @ September 17, #&$% is the wishes code
5             for him.";
6         int specialCharacterCount = 0;
7
8         for (int i = 0; i < statement.length(); i++) {
9             char ch = statement.charAt(i);
10            // Check if the character is a special character
11            if (!(Character.isLetterOrDigit(ch) || Character.isWhitespace(ch))) {
12                specialCharacterCount++;
13            }
14        }
15        System.out.println("Number of special Characters: " +
16            specialCharacterCount);
17    }
18 }
```

```
Output
Clear
java -cp /tmp/.55m01Ah0Jg/SpecialCharacterCount
Number of special Characters: 7

=== Code Execution Successful ===
```