

Kotlin -Assignment

Set 1

1. WAP to calculate Income tax for the given slab of income as mentioned below.

Input – Gross Income in INR (with Std. Deduction – INR 75000)

Output – Total tax deduction, Net Income

Net Income Range	Income Tax rates	Education Cess	Secondary and higher education cess
Up to 4,00,000	Nil	Nil	Nil
INR 4,00,001 - 800000	5% of (total income-300000)	2% of I.T	1% of I.T
INR 800001 - 1200000	INR 20000+10%of(Total Income – 800000)	2% of I.T	1% of I.T
INR 1200001 - 1600000	INR 60000+15% of (Total Income – 1200000)	2% of I.T	1% of I.T
INR 1600001 - 2000000	INR 120000+ 20% of (Total Income – 1600000)	2% of I.T	1% of I.T
INR 2000001 - 2400000	INR 200000 + 25% of (Total Income – 2000000)	2% of I.T	1% of I.T
INR 2400001 and above	INR 300000 + 30% of (Total Income – 2400000)	2% of I.T	1% of I.T

2. WAP to calculate Electricity bill based on the following unit consumption slab

State	Fixed Monthly Charges(INR)	Slab Low (units)	Slab High (units)	Rate/Unit(INR)
Rajasthan	230	1	50	5.00
	230	51	150	6.50
	275	151	300	7.50
	345	301	500	7.85
	400	501	above	8.00

--	--	--	--	--

And the calculation is based on the following:

Take no of units consumed as input (in KWh)

Calculate the total billing amount – based on the table above (including the fixed charge)

Output – Electricity bill amount, Number of units consumed.

3. Given a number as input – calculate if it is prime
4. Given a range of numbers as input print all the prime numbers existing within the range. (Use Sieve of Eratosthenes technique).
5. Given a range of numbers as input print all the perfect numbers existing within the range.
6. Given a range of numbers as input print all the set of pair numbers existing within the range.
7. WAP that calculates the number of triangles possible out of a given set of n numbers (each number representing length of the sides).
8. WAP that calculates the day of the week for any date in the past or future.

Following is a simple function suggested by Sakamoto, Lachman, Keith and Craver to calculate day.

Understanding the Maths:

Given day :14/09/1998

dd=14

mm=09

yyyy=1998 //non-leap year

Step 1: Information to be remembered.

- Magic Number Month array.
For Year: {0,3,3,6,1,4,6,2,5,0,3,5}
- DAY array starting from 0-6: {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}
- Century Year Value: 1600-1699 = 6
1700-1799 = 4
1800-1899 = 2

1900-1999 = 0

2000-2099 = 6.. Step 2: Calculations as per the steps

Last 2 digits of the year: 98

Divide the above by 4: 24

Take the date(dd): 14

Take month value from array: 5 (for September month number 9)

Take century year value: 0 (1998 is in the range 1900-1999 thus 0)

Sum: 141

Divide the Sum by 7 and get the remainder: $141 \% 7 = 1$

Check the Day array starting from index 0: Day[1] = Monday

**If leap year it will be the (remainder-1)

9. Find the largest and second largest as well as smallest and the second smallest element in an array of n unsorted numbers. (use a single loop)