Problem 1: A large company pays its salespeople on a commission basis. The salespeople each receive Rs. 2,000 per week plus 9% of their gross sales for that week. For example, a salesperson who sells Rs. 50,000 worth of chemicals in a week receives Rs. 2,000 plus 9% of Rs. 50,000, or a total of Rs. 6500. Develop the algorithm of the program that uses a while statement to input each salesperson's gross sales for last week and calculates and displays that salesperson's earnings. Process one salesperson's figures at a time. The program should terminate if the users enters a negative value as his gross sales for last week.

```
Enter sales in rupees (negative value to exit): 50000
Salary is: Rs. 6500
Enter sales in rupees (negative value to exit): 60000

Salary is: Rs. 7400

Enter sales in rupees (negative value to exit): 70000

Salary is: Rs. 8300

Enter sales in rupees (negative value to exit): -1
```

Problem 2: Write the algorithm of a program that prompts the user for entering a number, say x, and prints 1, 2, 3, …, x-1, x, x-1, …, 3, 2, 1. Assume that x > 1. For example, if the input is 5, the output should be 1 2 3 4 5 4 3 2 1, similarly if the input is 2, the output should be 1 2 1.

Problem 3: Write the algorithm of program that evaluates the series 5-10+7+15+9-20+11+25+… up to x terms, where the value of x is taken as input from the user. For example, the sum of the given series up to 3 terms should give 2 as output (5-10+7=2), similarly, the sum of this series up to 4 terms should give 17 as output (5-10+7+15=17), etc.

Problem 4: Write the algorithm of a program that receives a number as input and returns the square root of that number accurate to 5 decimal places. For example, square_root(25) should return 5.00000, squar_root(50) should return 7.07107, etc. Remember that you cannot use any built-in functions like sqrt, pow, etc. in your program.

Problem 5: Write the algorithm of a program that receives a number *i* as input and prints the i^{th} Fibonacci number. Remember that Fibonacci numbers are: 1, 1, 2, 3, 5, 8, 13, 21, …
Therefore, the output corresponding to an input of 4 should be 3 (the fourth Fibonacci number), similarly, the output corresponding to an input of 7 should be 13 (the seventh Fibonacci number), etc.

Problem 6: A country club, which currently charges $2,500 per year for membership, has announced it will increase its membership fee by 4% each year for the next six years. Write a program that uses a loop to display the projected rates for the next six years.

Problem 7: You want to buy a laptop worth Rs.30000/-. But your current savings are only Rs.5000/-. Given that the cost price of laptop increases by 2% every month and you can increase your savings by 7% monthly, how many months would you need to wait to buy that laptop. Try the problem for user given monthly increases as well. Identify first which looping construct suits your needs best.

Problem 8: Program to ask the user to enter the positive integer divisible by 6. Display the Message "Invalid Input, Enter again?" if the number entered is not divisible by 6 and prompt the user to enter the number again. Continue to do so until the user enters either a valid input or attempt-limit is reached. Attempt-limit is five consecutive wrong attempts. Display the Message "Five consecutive invalid attempts. Program Exited." and terminate your program in case the user enters five consecutive invalid inputs try break. When User has Successfully entered the number, say N, print the sum of series:

SUM = 1 – 2 + 3 – 4 + 5 . . . . . upto N Terms

Problem 9: Create a program that takes as input the length and breadth of a rectangle. And outputs a rectangle using a asterisk (*).

Suppose if you input 7 and 4 the output should be like

```
*******
*     *
*     *
*******
```

Lists:

Given following list of 31 elements, solve following problems.

l= [13, 99, 6, 76, 11, 83, 27, 84, 28, 67, 66, 22, 96, 46, 63, 21, 65, 48, 8, 14 , 84, 22, 28, 11, 83, 87, 11, 76, 6,83,27]

You can print on the same line using ",". For example following code will print the list l on the same even if we are using separate print statement in each iteration of while loop.

```
i=0

while i < len(l): # len is a function that returns us total number of element in the list

print l[i], # notice the trailing comma in the print statement

i=i+1
```

Problem 10: Write a python code to initialize the given list and print its length/ count of elements.

Problem 11: Write a python code to find the Maximum element of the list l.

Problem 12: Write a python code to find the Minimum element of the list l.

Problem 13: Write a python code to find the range of the values of list l. Note that range of values is equal to max - min.

Problem 14: Write a python code to find the average of elements in the list l.

Problem 15: Write a python code to search a given element (take as input) in the list.

Problem 16: Write a python code to swap first and last element of the list l.