Compilation instructions

1. **Writing Code**: Open an editor and write your code in it. Save the file with a proper name such as "area.cpp"

2. **Compiling:** To compile the code call the g++ command with -c flag and input file as arguments, i.e.

```
g++ -c area.cpp
```

If there are no errors, compiler will generate an object file names 'area.o'.

3. **Linking:** Finally you will call the linker to link the missing definitions with the object file and output exe names as arguments.

```
g++ -o exe area.o
```

4. **Running the code:** Simply call the exe file with './', i.e.

```
./exe
```

Exercise

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

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

Problem 10: Analyze the following input and output, and then write that a program that prints such a pattern given the input.

```
1 1 1 1 1
```

```
2 2 2 2
3 3 3
4 4
5
```

Problem 11: Analyze the following input and output, and then write that a program that prints such a pattern given the input.

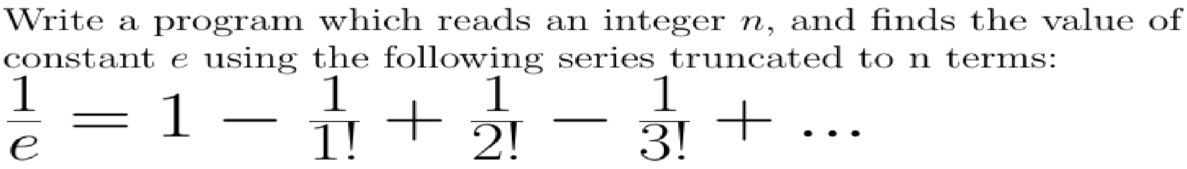
```
Input:3                      Input:4
Output:                      Output:
4 5 6                        7 8 9 10
2 3                          4 5 6
   1                         2 3
                             1
```

Problem 12: Write a program that displays following output

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

Problem 13: Write a program that reads a list of numbers n and displays largest, second largest and third largest.

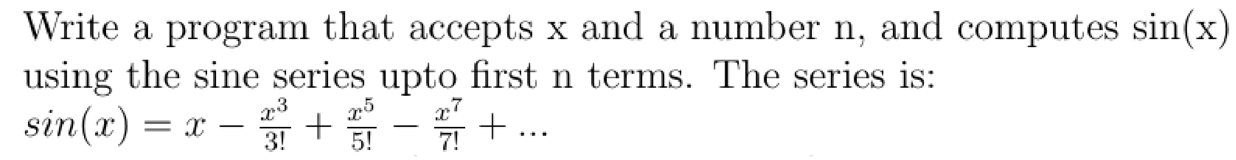
Problem 14:



Write a program which reads an integer n, and finds the value of constant e using the following series truncated to n terms:

$$\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots$$

Problem 15:



Write a program that accepts x and a number n, and computes sin(x) using the sine series upto first n terms. The series is:

$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ...$