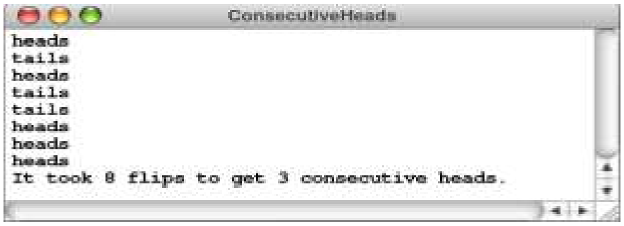
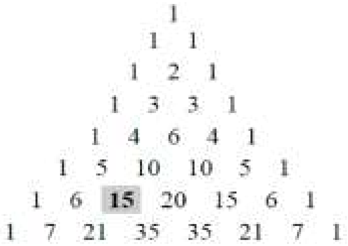
Object Oriented Programming using C++
Assignment II

Name: Submission Deadline:27/02/2017

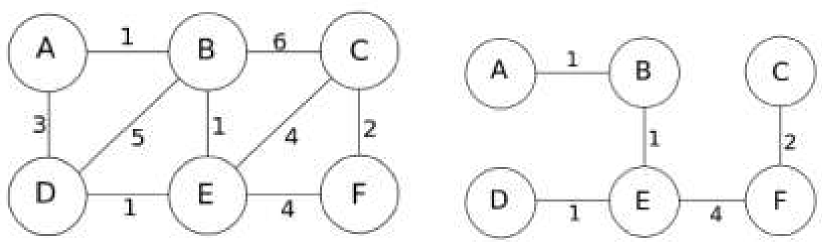
Roll: Group:

Remarks	Problem
	1. Write a program that simulates flipping a coin repeatedly and continues until three consecutive heads are tossed. At that point, your program should display the total number of coin flips that were made. The following is one possible sample run of the program:  heads tails heads tails tails heads heads heads It took 8 flips to get 3 consecutive heads.
	2. Pascal's Triangle has the interesting property that every entry is the sum of the two entries above it, except along the left and right edges, where the values are always 1. Consider, for example, the highlighted entry in the following display of Pascal's Triangle:  1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 **15** 20 15 6 1 1 7 21 35 35 21 7 1 This entry, which corresponds to c(6, 2), is the sum of the two entries—5 and 10—that appear above it to either side. Write a simple test program to demonstrate that your combinations function works. You may also write a program that uses c(n, k) to display the first ten rows of Pascal's Triangle.
	3. In a library, books, and journals are kept. Journals are issued to faculty members only. A student member can have 2 books issued at a time. For faculty members it is 10. For late return student members are charged Rs. 1 per day. Faculties are not charge. For journals additional information like issue no., date of publish, volume no., etc., are to be stored. For any transaction, members are supposed to place transactions slip. After necessary validations, transaction is carried out. Each transaction is to be noted into a register. Implement the system described above after designing the necessary classes. Use necessary **files to store and retrieve** information. Support should be there for date validation.

Object Oriented Programming using C++
Assignment II

Name: Submission Deadline:27/02/2017

Roll: Group:

	4. Implement a Priority Queue with linked lists. Provide functionalities exactly like the **template** provided in C++.
	5. You’re a government consultant and you’re trying to figure out the best way to arrange for internet access in your small island, Sunderban. There are N (3 ≤ N ≤ 250) villages on your island connected by M (N ≤ M ≤ 250) various roads and you can walk between any two villages on the island by traversing some sequence of roads. However, you’ve got a limited budget and have determined that the cheapest way to arrange for internet access is to build some fiber-optic cables along existing roadways. You have a list of the costs of laying fiber-optic cable down along any particular road, and want to figure out how much money you’ll need to successfully complete the project–meaning that, at the end, every village will be connected along some sequence of fiber-optic cables. Luckily, you’re also Sunderban’s resident computer scientist, and you remember hearing about Prim’s algorithm in one of your old programming classes. This algorithm is exactly the solution to your problem, but it requires a priority queue. Here’s the C++ standard template library to the rescue.  A 1 B 6 C 3 5 1 4 2 D 1 E 4 F A 1 B C 1 2 D 1 E 4 F Figure: Graph on the left and minimum spanning tree on the right