# HUMAN RESOURCES APPLICATION

To compile: `make`
To run: `./hr`

Design:
The human resources application consists of three classes: the `HRApp` class, the `Manager` class, and the `Employee` class.

The `HRApp` class, which is available to the driver of the program (`main.cpp`) represents an instance of the application itself. Its only public member functions are its constructors, a destructor, and a `run()` function. The `HRApp` class maintains a list of all users of the system and their permissions, the admin userID needed to access the system as an administrative user, and a list of managers.

The `Manager` class, which is available to the `HRApp` class, represents a manager. Managers store and maintain lists of employees. Because Managers store a list of employees, all manipulation of employees (viewing, editing, adding) must be done through the Manager class, through public member functions.

The `Employee` class, which is available to the Manager class, represents an employee. The salary, salary history, vacation balance, and annual bonus of the employee are stored as private attributes of the class. Setters and getters exist to access these attributes.

Permissions and authentication system:
The permissions and authentication system is maintained by the `HRApp` class. In order to access the system in a given role, the user must provide a `userID` (similar to a username and password, although for simplicity's sake, just one field will be used in this program). When an admin user creates a new user, they must provide a new `userID` for that user, as well as specifying their role/permissions. The admin `userID` is set when an instance of the app is created through the parameterized `HRApp` constructor. If the default constructor is used, the default admin `userID` will be "admin".

The `run()` function of the `HRApp` class prompts the user for their `userID`, from which it will determine the permissions of that user and permit them to manipulate the data in the program in the appropriate way. For example: a user enters a userID that is associated with the role of "an HR employee" in the list of users. The program will then allow them to view their own information, as well as any other employee's information who is not part of the HR department.

Disadvantages:
- High cost of searching for employees, managers, and users, maintenance of multiple unsorted vectors in the system. This could be optimized by sorting the vectors or using a different data structure.

- UserID of each user must be recorded by the admin and communicated to the new user upon creation of a new user.