Vector Practice

- Create a vector to hold 40 names of students.

```
vector <string> names (40);
```

- Initialize a vector called "grades" to the value: 100

```
//this answer assumes it has been created with room for # integers
//vector<int> grades (#);
for(unsigned i=0; i<grades.size(); ++i) {
    grades.at(i) = 100;
}
```

- Compute the average of an entire vector of doubles.

```
double sum = 0;
for(unsigned i=0; i<grades.size(); ++i) {
    sum += grades.at(i);
}
double average = sum / grades.size();
```

- Create a vector with 20 strings. Allow the user to input this vector. Randomly choose two elements from the vector. Output their concatenation.

```
vector<string> words(20);
for(unsigned i=0; i<20; i++) {
    string s;
    cin>>s;
    words.push_back(s);
}
int r1=rand()%20;
int r2=rand()%20;
cout << words.at(r1)<< words.at(r2);
//you can also add strings like:
// string s= words.at(r1)+words.at(r2);
// cout << s;
```

- Given a vector named fairies. Randomly choose two elements from the vector and output their sum.

```
int hogwarts = rand()%fairies.size();
int labyrinth = rand()%fairies.size();
//this code assumes fairies is a vector of integers.
//
int sum = fairies.at(hogwarts)+fairies.at(labyrinth);
```

- Show how to output every other element of a vector called numbers.

```
for(unsigned i=0; i<numbers.size(); i+=2){
    cout << numbers.at(i) << " ";
}
cout <<endl;
// Does a horizontal print
```

- Show how to fill a vector of 30 integers with only even numbers.

```
//Assume Different even #s are required
// Assume vector is called nums & has no "spare"
for(unsigned i=0; i<30; ++i){
    // 2*i is always even!
    nums.push_back(2*i);
}
```

- Show how to fill a vector of 30 integers with random numbers from 1-10.

```
// Assume vector is called nums & has no "spare"
for(unsigned i=0; i<30; ++i){
    int r = rand()%10+1;
    nums.push_back(r);
}
```

- Show how to take a vector and sum all of its elements.

```
//we did this earlier w/ the grades vector
double sum = 0;
for(unsigned i = 0; i < grades.size(); ++i) {
    sum += grades.at(i);
}
```

- Show how to take a vector of doubles and increase every value by 10.

```
//Assume vector is called salaries
for(unsigned i = 0; i < salaries.size(); ++i) {
    salaries.at(i) = salaries.at(i) + 10;
}
```

- Show how to take a vector of doubles and replace each item with its value squared.

```
//Assume vector is called sq.
for(unsigned i=0; i<sq.size(); ++i){
    sq.at(i) = sq.at(i) * sq.at(i);
}
```

- Show how to take a vector of doubles and replace each item with its square root.

```
//Assume vector is called sq.
for(unsigned i=0; i<sq.size(); ++i){
    sq.at(i) = Sqrt(sq.at(i));
}
```

- Show how to rotate a vector such that the second becomes first, the third becomes second and so on. The first element should be placed in the last slot. Assume this is a vector of doubles (although the same logic should work for any vector)

```
// Example [7, 9, 8, 4]
// Becomes [9, 8, 4, 7]
//Assume vector is called v
double temp = v.at(0);//save the first value
for(unsigned i=0; i<v.size()-1 ; ++i){ //notice the -1 here
    v.at(i) = v.at(i+1); //so v.at(0) = v.at(1) & then @1=@2.
}
// @ this point only the last element is unchanged, b/c of the -1 in the for loop
v.at(v.size()-1) = temp;
```

- Show how to rotate a vector such that the first becomes second, the second becomes third and so on. The last element should be placed in the first slot. Assume this is a vector of strings (although the same logic should work for any vector)

```
//Ex [7, 9, 8]
//Becomes [4, 7, 9, 8]
// Assume vector is called v
double temp = v.at(v.size()-1); //Save the last value
for(unsigned i = v.size()-1; i>0; ++i){ //we must do this in the
                                        //reverse order b/c, we want to
                                        // move things from left to right
//  ↗ starts @ the last, ends w/ i=1
  v.at(i) = v.at(i-1);
}
v.at(0) = temp; // First comes from the last value, but that was
                // changed so we get its value from our temp va-
```