

The format of the input file is defined in the project write-up. The easiest way to read it is to read a full line into a character string, then extract the necessary data from that string. Here is a partially defined function to do that:

```
void readData(char* filename, vector<PostNet>& codes)
{
    ifstream input;
    input.open(filename);
    if (!input)
    {
        cout << "Could not open " << filename << endl;
        return;
    }
    char line[1000];
    char b[28];
    char z[6];
    while (true)
    {
        // get the next full line from the input file
        input.getline(line, 1000);
        // if the line starts with 'e', we're finished
        if (line[0] == 'e') break;
        // if the line starts with 'b', rest of line is a bar code
        else if (line[0] == 'b')
        {
            // copy bar code into local array b
            strcpy(b, &(line[2]));
            // build an object and push it into your vector
            // YOUR CODE HERE
        }
        // if the line starts with 'z', rest of line is a zip code
        else if (line[0] == 'z')
        {
            // copy zip code into local array z
            strcpy(z, &(line[2]));
            // build an object and push it into your vector
            // YOUR CODE HERE
        }
        else
        {
            // error check - shouldn't happen on a well-formed
            // input file
            cout << "invalid line: " << line << endl;
        }
    }
    input.close();
}
```

The getline() method is defined for reading from a C++ input stream into a C character array:

<http://www.cplusplus.com/reference/iostream/istream/getline/>

If you prefer to use C++ string objects, there is a similar function for strings:

<http://www.cplusplus.com/reference/string/getline/>

The strcpy() function makes a copy of a C character array:

<http://www.cplusplus.com/reference/cstring/strcpy/>

if you are using an array of objects, rather than an STL vector, modify the function prototype and handle the size of the array yourself:

```
short readData(char* filename, PostNet codes[], short capacity)
{
    short size = 0;
    ifstream input;
    input.open(filename);
    if (!input)
    {
        cout << "Could not open " << filename << endl;
        return;
    }
    char line[1000];
    char b[28];
    char z[6];
    while (true)
    {
        // get the next full line from the input file
        input.getline(line, 1000);
        // if the line starts with 'e', we're finished
        if (line[0] == 'e') break;
        // if the line starts with 'b', rest of line is a bar code
        else if (line[0] == 'b')
        {
            // copy bar code into local array b
            strcpy(b, &(line[2]));
            // build an object and copy it into your array
            // at index position size and increment the size
            // YOUR CODE HERE
            size++;
        }
        // if the line starts with 'z', rest of line is a zip code
        else if (line[0] == 'z')
        {
            // copy zip code into local array z
            strcpy(z, &(line[2]));
            // build an object and copy it into your array
            // at index position size and increment the size
            // YOUR CODE HERE
            size++;
        }
        else
        {
            // error check - shouldn't happen on a well-formed
            // input file
            cout << "invalid line: " << line << endl;
        }
        // check the capacity to prevent overflows:
        if (size == capacity)
        {
            cout << "Array is full. " << endl;
            break;
        }
    }
    input.close();
    return size;
}
```