The materials included in these files are intended for use by AP teachers for course and exam preparation in the classroom; permission for any other use must be sought from the Advanced Placement Program®. Teachers may reproduce them, in whole or in part, in limited quantities, for face-to-face teaching purposes but may not mass distribute the materials, electronically or otherwise. These materials and any copies made of them may not be resold, and the copyright notices must be retained as they appear here. This permission does not apply to any third-party copyrights contained herein.
Complete function CalculateModes below.

apvector<int> CalculateModes(const apvector<int> & tally)
// precondition: tally.length() > 0
// postcondition: returns an apvector that contains the mode(s);
// the apvector’s length equals the number of modes

int mode = FindMax(tally);
apvector<int> modes(tally.length());
int nummode = 0;
int k;
for (k = 0; k < tally.length(); k++)
  if (mode == tally[k])
    { modes[nummode] = k;
      nummode++;
    }

mode.resize(nummode);
return modes;

GO ON TO THE NEXT PAGE.
(b) You will write the function `KthDataValue`, which is described as follows. `KthDataValue` returns the kth data value when the data values are considered in sorted order. Recall that the indexes of the array represent possible data values and that each array location contains the frequency of the value corresponding to its index.

In the example reprinted below, the first ten data values are 2, the next five data values are 3, and the next ten data values are 4. `KthDataValue(tally, 1)` returns 2, `KthDataValue(tally, 14)` returns 3, `KthDataValue(tally, 15)` returns 3, and `KthDataValue(tally, 16)` returns 4.

<table>
<thead>
<tr>
<th>Value</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Complete function `KthDataValue` below.

```c
int KthDataValue(const apvector<int> & tally, int k)
{ // precondition: tally.length() > 0;
  // 0 < k ≤ total number of values in the data collection
  // postcondition: returns the kth value in the data collection
  // represented by tally
  int x;
  for (x = 0; x < tally.length(); x++)
  {
    int k = k - tally[x];
    if (k <= 0)
      return x;
  }
```

GO ON TO THE NEXT PAGE.
Complete function CalculateModes below.

```cpp
apvector<int> CalculateModes(const apvector<int> & tally)
// precondition: tally.length() > 0
// postcondition: returns an apvector that contains the mode(s);
//                 the apvector's length equals the number of modes
{
    apvector<int> calc(0);
    for (int i = 0; i < tally.length(); i++)
        if (tally[i] == FindMax(tally))
            calc.resize(calc.length()+1);
            calc[calc.length()-1] = i;
    return calc;
}
```

GO ON TO THE NEXT PAGE.
(b) You will write the function \texttt{KthDataValue}, which is described as follows. \texttt{KthDataValue} returns the \textit{k}th data value when the data values are considered in sorted order. Recall that the indexes of the array represent possible data values and that each array location contains the frequency of the value corresponding to its index.

In the example reprinted below, the first ten data values are 2, the next five data values are 3, and the next ten data values are 4. \texttt{KthDataValue(tally, 1)} returns 2, \texttt{KthDataValue(tally, 14)} returns 3, \texttt{KthDataValue(tally, 15)} returns 3, and \texttt{KthDataValue(tally, 16)} returns 4.

<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
</tr>
</tbody>
</table>

Complete function \texttt{KthDataValue} below.

```cpp
int KthDataValue(const apvector<int> & tally, int k) {
  // precondition: \texttt{tally.length}() > 0;
  // postcondition: \texttt{returns the kth value in the data collection represented by tally}
  for (int i = 0; i < tally.length(); i++) {
    for (int j = 0; j < tally[i]; j++)
      counter++;
    if (counter == k)
      return i;
  }
}
```

Go on to the next page.
Complete function CalculateModes below.

apvector<int> CalculateModes(const apvector<int> & tally)
// precondition:  tally.length() > 0
// postcondition: returns an apvector that contains the mode(s);
//                the apvector's length equals the number of modes
{
    apvector<int> modes;
    int j = 0, numModes;

    for (k = 0; k < tally.length(); k++)
    {
        if (tally[k2] == FindMax(tally))
        {
            numModes++;
            modes.reserve(numModes);
            modes[j2 = tally[k2];
            j++;
        }
    }

    return modes;
}
You will write the function \texttt{KthDataValue}, which is described as follows. \texttt{KthDataValue} returns the \textit{kth} data value when the data values are considered in sorted order. Recall that the indexes of the array represent possible data values and that each array location contains the frequency of the value corresponding to its index.

In the example reprinted below, the first ten data values are 2, the next five data values are 3, and the next ten data values are 4. \texttt{KthDataValue(tally, 1)} returns 2, \texttt{KthDataValue(tally, 14)} returns 3, \texttt{KthDataValue(tally, 15)} returns 3, and \texttt{KthDataValue(tally, 16)} returns 4.

\begin{center}
\begin{tabular}{c|cccccccccccccc}
\textbf{Value} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
\hline
\textbf{Frequency} & 0 & 0 & 10 & 5 & 10 & 0 & 7 & 1 & 0 & 6 & 0 & 10 & 3 & 0 & 0 & 1 \\
\end{tabular}
\end{center}

Complete function \texttt{KthDataValue} below.

\begin{verbatim}
int KthDataValue(const apvector<int> & tally, int k)
// precondition:  tally.length() > 0;
// postcondition: returns the kth value in the data collection
// represented by tally

int j;
for (j = 0; j < tally.length(); j++)
    if (tally[j] == k)
        return j;

\end{verbatim}