```
    if (a[i] == x) return i;
  return n;
}
index(44,a,7) = 4
index(50,a,7) = 7
```

6.7 Example 6.15 modified so that that it determines whether the array is nonincreasing:

```
bool isNonincreasing(int a[], int n)
{ for (int i=1; i<n; i++)
    if (a[i]>a[i-1]) return false;
  return true;
}
```

6.8

```
float min(float a[], int n)
{ assert(n >= 0);
  float min=a[0];
  for (int i=1; i<n; i++)
    if (a[i] < min) min = a[i];
  return min;
}
```

6.9

```
int minIndex(float a[], int n)
{ assert(n >= 0);
  int j=0;
  for (int i=1; i<n; i++)
    if (a[i] < a[j]) j = i;
  return j;
}
```

6.10

```
void getExtremes(float& min, float& max, float a[], int n)
{ assert(n >= 0);
  min = max = a[0];
  for (int i=1; i<n; i++)
    if (a[i] < min) min = a[i];
    else if (a[i] > max) max = a[i];
}
```

6.11

```
void largest(float& max1, float& max2, float a[], int n)
{ assert(n >= 1);
  if (n == 1) return a[0];
  int i1=0, i2;
  for (int i=1; i<n; i++)
    if (a[i] > a[i1]) i1 = i;
  max1 = a[i1];
  i2 = ( i1 == 0 ? 1 : 0 );
  for (int i=i2+1; i<n; i++)
    if (i != i1 && a[i] > a[i2]) i2 = i;
  max2 = a[i2];
}
```

6.12

```
void remove(float a[], int& n, int i)
{ for (int j=i+1; j<n; j++)
    a[j-1] = a[j];
  --n;
}
```

6.13

```
bool removeFirst(float a[], int& n, float x)
{ for (int i=0; i<n; i++)
    if (a[i] == x)
```