Sample Program 11.1

About Arrays

Problem statement. Write a declaration statement to create an uninitialized `int` array named `a`, of fixed size 100.

Solution #1.

```
int a[100];
```

Solution #2.

```
const int SIZE = 100;
int a[SIZE];
```

Problem statement. Write a code block to sort the above array (based on Solution #2) from lo-to-hi, and output the minimum and maximum values.

Solution.

```
sort(a, a + SIZE);  // requires #include <algorithm>
cout << "The minimum value is " << a[0] << endl;
cout << "The maximum value is " << a[SIZE - 1] << endl;
```

Problem statement. Write a code block to output the above sorted array, with labels.

Solution.

```
for (int i = 0; i < SIZE; i++)
  cout << "a[" << i << "] = " << a[i] << endl;
```

Problem statement. Write a code block to search the above sorted array for any negative value.

Solution.

```
bool found = false;
for (int i = 0; i < SIZE; i++)
{
  if (a[i] < 0)
  {
    found = true;
    break;
  }
}

if (found)
  cout << "At least one negative value found in the array 'a'." << endl;
else
  cout << "No negative values found in the array 'a'." << endl;
```

Problem statement. Write a declaration statement to create an uninitialized dynamic array of strings, named `s`.

Solution.

```
int size;
cout << "What size do you want for the array of strings [1 or greater]? ";
cin >> size;
cin.ignore(1000, 10);

string* s = new string[size]; // requires #include <string>
```

Problem statement. Write a code block to sort the above array alphabetically, and output the minimum and maximum values.

Solution.

```
sort(s, s + size);  // requires #include <algorithm>
cout << "The minimum value is " << s[0] << endl;
cout << "The maximum value is " << s[size - 1] << endl;
```

Problem statement. Write a code block to output the above sorted array, with labels.

Solution.

```
for (int i = 0; i < size; i++)
  cout << "s[" << i << "] = " << s[i] << endl;
```

Problem statement. Write a code block to count the number of blanks in the above array.

Solution.

```
int nBlanks = 0;
for (int i = 0; i < size; i++)
  if (s[i].length() == 0)
    nBlanks++;

cout << "#of blanks found in the array 's': " << nBlanks << endl;
```