

**C++ Program Design/3e**  
**Chapter 9 – Lists**  
**Answers to Self-Check Exercises**

1. answer: Elements.
2. answer: `int Scores[2000];`.
3. answer: Subscripting or indexing.
4. answer: `StackObject StackElements[50];`
5. answer: `float GradePointAvg[500];`
6. answer:
  - (a) `Hits[k] += 6;`
  - (b) `Hits[k+1] = Hits[i];`
  - (c) `Hits[j+5] = Hits[i] + Hits[k];`
7. answer: 0.
8. answer: Uninitialized, since the array definition is given at the local scope.
9. answer: `int Distance[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};`
10. answer:

```
Weights[0] = 3,  
Weights[1] = 4,  
Weights[2] = Weights[3] = Weights[4] = Weights[5] = 0
```
11. answer: 5
12. answer: `float Coefficients[] = {23.1, 34.5, 35.6, 28.0, 35.0, 88.2, 91.3};`
13. answer: `char Title[] = "Madam Bovary";`
14. answer: No.
15. answer: (Note the const value of N should be 19.)

```
void reverse(char A[],char B[],int n) {  
    for(int i=0;i<n;++i)  
        B[n-2-i] = A[i];  
}
```

16. answer:

```
int LessThan(int a[], int n, int v = 0) {
    int ctr = 0;
    for(int i=0; i<n; ++i) {
        if (a[i] < v)
            ++ctr;
    }
    return ctr;
}
```

17. answer:

```
bool Equal(int a[], int b[], int n) {
    bool result = true;
    for(int i=0; i<n; ++i) {
        if (a[i] != b[i])
            result = false;
    }
    return result;
}
```

18. answer:

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

19. answer:

```
#include <iostream>
#include <string>
#include <time.h>

#include "randint.h"

using namespace std;

int main() {
    srand((unsigned int) time(0));
    int n;
    cout << "Enter the number of integers to sort: ";
    cin >> n;
    int A[10000];
    RandomInt U(1,10000);
    for (int i=0; i<n; ++i)
        A[i] = U.Draw();
    for(i=0; i<n; ++i) {
        cout << A[i] << " ";
    }
    cout << endl << endl;
    for(int j=0; j<n;++j) {
        for(int k=n-1; k>j; --k) {
            if (A[k-1] > A[k]) {
                int temp = A[k];
                A[k] = A[k-1];
                A[k-1] = temp;
            }
        }
    }

    for(i=0; i<n; ++i) {
        cout << A[i] << " ";
    }
    cout << endl << endl;
    return 0;
}
```

## 20. answer: Here is one method using a horizontal histogram

```
#include <iostream>    // for file reading and writing
#include <fstream>     // for cout and cerr
#include <string>      // for string type
#include <iomanip>     // for setw, etc.
#include <cmath>      // for sqrt
using namespace std;

// function prototypes

void Histogram(int L[]);

int main() {

    cout << "\nName of file that holds text: ";
    string FileName2;
    cin >> FileName2;
    ifstream fin(FileName2.c_str());
    if (! fin) {
        cout << "Cannot open " << FileName2 << endl;
        exit(1);
    }

    char OrigText[200];
    int i = 0;
    while (fin >> OrigText[i]) {
        if(isalpha(OrigText[i])) {
            ++i;
            if((OrigText[i]>='A') && (OrigText[i]<='Z'))
                OrigText[i] = OrigText[i]-'A'+ 'a';
        }
    }

    int number_of_scores = i;

    // prepare for histogram
    int histogramArray[26] = {0};
    for(i=0;i<number_of_scores;++i) {
        if ((OrigText[i] == 'a') || (OrigText[i] == 'A'))
            ++histogramArray[0];
        else if ((OrigText[i] == 'b') || (OrigText[i] == 'B'))
            ++histogramArray[1];
        else if ((OrigText[i] == 'c') || (OrigText[i] == 'C'))
            ++histogramArray[2];
        else if ((OrigText[i] == 'd') || (OrigText[i] == 'D'))
            ++histogramArray[3];
        else if ((OrigText[i] == 'e') || (OrigText[i] == 'E'))
            ++histogramArray[4];
        else if ((OrigText[i] == 'f') || (OrigText[i] == 'F'))
            ++histogramArray[5];
        else if ((OrigText[i] == 'g') || (OrigText[i] == 'G'))
            ++histogramArray[6];
        else if ((OrigText[i] == 'h') || (OrigText[i] == 'H'))
            ++histogramArray[7];
    }
}
```

```

else if ((OrigText[i] == 'i') || (OrigText[i] == 'I'))
    ++histogramArray[8];
else if ((OrigText[i] == 'j') || (OrigText[i] == 'J'))
    ++histogramArray[9];
else if ((OrigText[i] == 'k') || (OrigText[i] == 'K'))
    ++histogramArray[10];
else if ((OrigText[i] == 'l') || (OrigText[i] == 'L'))
    ++histogramArray[11];
else if ((OrigText[i] == 'm') || (OrigText[i] == 'M'))
    ++histogramArray[12];
else if ((OrigText[i] == 'n') || (OrigText[i] == 'N'))
    ++histogramArray[13];
else if ((OrigText[i] == 'o') || (OrigText[i] == 'O'))
    ++histogramArray[14];
else if ((OrigText[i] == 'p') || (OrigText[i] == 'P'))
    ++histogramArray[15];
else if ((OrigText[i] == 'q') || (OrigText[i] == 'Q'))
    ++histogramArray[16];
else if ((OrigText[i] == 'r') || (OrigText[i] == 'R'))
    ++histogramArray[17];
else if ((OrigText[i] == 's') || (OrigText[i] == 'S'))
    ++histogramArray[18];
else if ((OrigText[i] == 't') || (OrigText[i] == 'T'))
    ++histogramArray[19];
else if ((OrigText[i] == 'u') || (OrigText[i] == 'U'))
    ++histogramArray[20];
else if ((OrigText[i] == 'v') || (OrigText[i] == 'V'))
    ++histogramArray[21];
else if ((OrigText[i] == 'w') || (OrigText[i] == 'W'))
    ++histogramArray[22];
else if ((OrigText[i] == 'x') || (OrigText[i] == 'X'))
    ++histogramArray[23];
else if ((OrigText[i] == 'y') || (OrigText[i] == 'Y'))
    ++histogramArray[24];
else
    ++histogramArray[25];
}

Histogram(histogramArray);

return 0;
}

void Histogram(int L[]) {
    int low=0, high = 25;
    char ch = 'a';
    for(int i=0; i<=25; ++i) {
        cout << ch << " | ";
        for(int j=1; j<=L[i]; ++j)
            cout << '*';
        cout << endl;
        ++ch;
    }
    cout << endl;
}

```

21. answer:

```
int LessThan(vector<int> &A, int n, int v = 0) {
    int ctr = 0;
    for(int i=0; i<n; ++i) {
        if (A[i] < v)
            ++ctr;
    }
    return ctr;
}
```

22. answer:

```
bool Equal(vector<double> &A, vector<double> &B, int n) {
    bool result = true;
    for(int i=0; i<n; ++i) {
        if (A[i] != B[i])
            result = false;
    }
    return result;
}
```

23. answer:

```
bool IsSorted(vector<int> &A, int n) {
    bool result;
    for(int i=1; i<n; ++i) {
        if(A[i] > A[i-1])
            result = false;
    }
    return result;
}
```

24. answer:

```
#include <iostream>
#include <string>
#include <vector>
#include <time.h>

#include "randint.h"

using namespace std;

int main() {
    srand((unsigned int) time(0));
    int n;
    cout << "Enter the number of integers to sort: ";
    cin >> n;
    vector<int> List(n);
    RandomInt U(1,10000);
    for (int i=0; i<n; ++i)
        List[i] = U.Draw();
    for(i=0; i<n; ++i) {
        cout << List[i] << " ";
    }
    cout << endl << endl;
    for(int j=0; j<n;++j) {
        for(int k=n-1; k>j; --k) {
            if (List[k-1] > List[k]) {
                int temp = List[k];
                List[k] = List[k-1];
                List[k-1] = temp;
            }
        }
    }

    for(i=0; i<n; ++i) {
        cout << List[i] << " ";
    }
    cout << endl << endl;
    return 0;
}
```

25. answer: This question is exactly the same as self-check question 9-20.

26. answer:

```
void PrintMatrix(int List[][MaxCols], int m, int n) {
    for (int i=0; i<m; ++i) {
        for (int j=0; j<n; ++j)
            cout << List[i][j] << " ";
        cout << endl;
    }
}
```

27. answer: Here is the code for a square matrix:

```
const int n = 5;

void Swap(int L[n][n], int i, int j) {
    int temp;
    temp = L[i][j];
    L[i][j] = L[j][i];
    L[j][i] = temp;
}

void FlipDiag(int List[n][n]) {
    for (int i=0; i<n; ++i)
        for (int j=0; j<n; ++j)
            if (i<j)
                Swap(List, i, j);
}
```

28. answer:

```
void Scale(double List[rows][cols], double scale[], int r,
           int c) {
    for (int i=0; i<c; ++i)
        for (int j=0; j<r; ++j)
            List[j][i] *= scale[i];
}
```