

Solutions to practice problems on arrays.

(1) Read the entries of an array of 10 integers from a user. Compute x as the average of the 10 entries and then compute the average of those entries that are greater than or equal to x . Print this final average.

```
#include <iostream>
using namespace std;

int main() {
    int a[10], c, count = 0;
    double total1 = 0, average1, total2 = 0, average2;

    cout << "Enter 10 integers for the array" << endl;
    for (c = 0; c < 10; c++) cin >> a[c];

    for (c = 0; c < 10; c++) total1 += a[c];
    average1 = total1 / 10;
    for (c = 0; c < 10; c++)
        if (a[c] >= average1) {
            total2 += a[c];
            count++;
        }
    average2 = total2 / count;
    cout << average2 << endl;
    return 0;
}
```

(2) Write a C++ program that sets up an array of integers with capacity 20. It should then generate the 20 entries randomly in turn. Each entry must be an integer between 1 and 20, however it must also be different from all previous entries in the array. Generate the entries as random numbers and repeatedly make new numbers until a legal entry value is found. The program should finish by printing the list of 20 array values that it has selected.

```
#include <iostream>
using namespace std;

int main() {
    int a[20];
    for (int c = 0; c < 20; c++) {
        bool foundOK = false;
        while (!foundOK) {
            a[c] = rand()%20 + 1;
            foundOK = true;
            for (int b = 0; b < c; b++)
                if (a[c] == a[b]) foundOK = false;
        }
    }

    for (int c = 0; c < 20; c++)
        cout << a[c] << " ";
    cout << endl;
    return 0;
}
```

(3) Eight queens are to be placed on an 8 x 8 chessboard in such a way that one queen is to be in each row. A program will store an array $x[]$ with capacity 8 to

represent such a configuration. If $x[r]$ has value c then in row r there is a queen in column c . Write a program that asks a user to enter the columns that contain queens in the 8 rows. The program then prints the board. For example, if the user enters: 0,3,4,0,0,7,6,7 the program should print:

```
Q.....
...Q....
....Q...
Q.....
Q.....
.....Q
.....Q.
.....Q
```

```
#include <iostream>
using namespace std;
```

```
int main() {
    int x[8];
    cout << "Enter 8 columns for queens:";
    for (int r = 0; r < 8; r++)
        cin >> x[r];
    cout << endl;
    for (int r = 0; r < 8; r++) {
        for (int c = 0; c < 8; c++)
            if (c == x[r]) cout << "Q";
            else cout << ".";
        cout << endl;
    }
    return 0;
}
```

(4) (Similar to problem 3, but the user specifies a row for each column.) Eight queens are to be placed on an 8 x 8 chessboard in such a way that one queen is to be in each column. A program will store an array $x[]$ with capacity 8 to represent such a configuration. If $x[c]$ has value r then in row r there is a queen in column c . Write a program that asks a user to enter the rows that contain queens in the 8 columns. The program then prints the board. For example, if the user enters: 2,3,4,0,1,7,6,5 the program should print:

```
...Q....
....Q...
Q.....
.Q.....
..Q.....
.....Q
.....Q.
.....Q..
```

```
#include <iostream>
using namespace std;
```

```
int main() {
    int x[8];
    cout << "Enter 8 rows for queens:";
    for (int c = 0; c < 8; c++)
        cin >> x[c];
    cout << endl;
```

```

    for (int r = 0; r < 8; r++) {
        for (int c = 0; c < 8; c++)
            if (r == x[c]) cout << "Q";
            else cout << ".";
        cout << endl;
    }
    return 0;
}

```

(5) Again related to problems 3 and 4.

Eight queens are to be placed on an 8 x 8 chessboard in such a way that one queen is to be in each row and one queen is to be in each column. A program will store an array `x[]` with capacity 8 to represent such a configuration. If `x[c]` has value `r` then in row `r` there is a queen in column `c`. Write a program that asks a user to enter the rows that contain queens in the 8 columns. The program then checks whether there is just one queen per row. For example, if the user enters: 2,3,4,0,1,7,6,5 the program should print: OK (because the user has entered the configuration that was entered in problem 3). But if the user enters 0,0,1,2,3,4,5,6 the program should print: No good. (Why?)

```

#include <iostream>
using namespace std;

int main() {
    int x[8];
    cout << "Enter 8 rows for queens:";
    for (int c = 0; c < 8; c++)
        cin >> x[c];
    cout << endl;

    int rowCount[8];
    for (int r = 0; r < 8; r++) rowCount[r] = 0;
    for (int c = 0; c < 8; c++) rowCount[x[c]]++;

    for (int r = 0; r < 8; r++)
        if (rowCount[r] != 1) {
            cout << "No good." << endl;
            return 0;
        }
    cout << "OK" << endl;
    return 0;
}

```

(6) Again related to problems 3, 4, and 5.

Eight queens are to be placed on an 8 x 8 chessboard in such a way that one queen is to be in each row and one queen is to be in each column and so that no two queens share a diagonal. A program will store an array `x[]` with capacity 8 to represent such a configuration. If `x[c]` has value `r` then in row `r` there is a queen in column `c`. Write a program that asks a user to enter the rows that contain queens in the 8 columns. The program then checks whether the configuration is legal. For example, if the user enters: 2,3,4,0,1,7,6,5 the program should print: No good. (Why?)

```

#include <iostream>
using namespace std;

int main() {
    int x[8];

```

```

cout << "Enter 8 rows for queens:";
for (int c = 0; c < 8; c++)
    cin >> x[c];
cout << endl;

bool isOK = true;
for (int c = 0; c < 8; c++)
    for (int b = 0; b < c; b++) {
        if (x[b] == x[c]) isOK = false; // duplicate row
        if ((x[b] - b) == (x[c] - c))
            isOK = false; // duplicate diagonal
        if ((x[b] + b) == (x[c] + c))
            isOK = false; // duplicate diagonal
    }
if (!isOK) cout << "No good." << endl;
else cout << "OK" << endl;
return 0;
}

```

(7)

Write a program to read an array of 11 integers from a user and compute the median entry of the array.

```

#include <iostream>
using namespace std;

int main() {
    int x[11];
    cout << "Enter 11 integers:";
    for (int c = 0; c < 11; c++)
        cin >> x[c];

    int lower[11], higher[11];
    for (int c = 0; c < 11; c++)
        lower[c] = higher[c] = 0;

    for (int c = 0; c < 11; c++)
        for (int b = 0; b < 11; b++) {
            if (x[b] < x[c]) lower[c]++;
            if (x[b] > x[c]) higher[c]++;
        }

    int median;
    for (int c = 0; c < 11; c++)
        if ((lower[c] <= 5) && (higher[c] <= 5))
            median = x[c];
    cout << "The median is " << median << endl;

    return 0;
}

```