8

Case Study of Array

Example 1: Matrix Multiplication

- Problem definition
  - **Use arrays to perform matrix multiplication**
  - **The matrix sizes must be reasonable**
    - **Ex:**

$$A \in R^{m \times n}, A \in R^{m \times o} \Rightarrow (AB) \in R^{m \times o}$$

$$(AB)_{i,j} = \sum_{r=1}^{n} A_{i,r} B_{r,j} \qquad 1 \le i \le m \ \ and \ \ 1 \le j \le o$$

```
1  // matrix multiplication
2
3  #include <iostream.h>
4  #include <iomanip.h>
5
6  // define the matrix size
7  #define M 3
8  #define N 4
9  #define O 5
10
11 class matrix_multi
12 {
13   int p[M][N];   // define the matrix size
14   int q[N][O];
15   int r[M][O];   // r[M][N] = p[M][N] * q[N][N]
16
17 public:
18   matrix_multi();
19 };
20
21 matrix_multi::matrix_multi()
22 {
23   int p[M][N]={1,-99,10,2,0,1,-17,6,-1,1,2,3};
24   int q[N][O]={1,1,0,1,-1,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0};
25   int i, j, k;
26   for(i=0;i<M;i++)
27     for(j=0;j<O;j++)
28       r[i][j]=0;
29
30   for(i=0;i<M;i++)
31     for(j=0;j<O;j++)
32       for(k=0;k<N;k++)
33         r[i][j]+=p[i][k]*q[k][j];
34
```

```
35    cout << "\tp[][] = ";
36    for(i=0;i<M;i++)                          // show the matrix p[i][k]
37    {
38      cout  << endl << "\t";
39      for(k=0;k<N;k++)
40        cout << setw(5) <<  p[i][k];
41    }
42    cout << endl << endl;
43
44    cout << "\tq[][] = ";                     // show the matrix q[k][j]
45    for(k=0;k<N;k++)
46    {
47      cout << endl << "\t";
48      for(j=0;j<O;j++)
49        cout << setw(5) << q[k][j];
50    }
51    cout << endl <<endl;
52
53    cout << "\tr[][] = p[][] * q[][]";       // show results of the matrix multiplication
54    for(i=0;i<M;i++)
55    {
56      cout << endl << "\t";
57      for(j=0;j<O;j++)
58        cout << setw(5) << r[i][j];
59    }
60    cout << endl << endl;
61  }
62
63  void main()
64  {
65    matrix_multi obj;
66  }
67
68
```

Example 2 : Check Duplication

- **Problem definition (Exercise 6.12)**
  - **Use a single-subscripted array**
  - **Read in 20 numbers, each of which is between 10 and 100, inclusive**
  - **As each number is read, print it only if it is not a duplicate of a number already read**
  - **Provide for the "worst case" in which all 20 numbers are different**
  - **Use the smallest possible array to solve this problem**

Check Duplication (1/2)

```
1  /* Exercise 6.12 Solution */
2  #include <stdio.h>
3  #define MAX 20
4
5  int main( void )
6  {
7     int a[ MAX ] = { 0 }; /* array for user input */
8     int i;                /* loop counter */
9     int j;                /* loop counter */
10    int k = 0;            /* number of values currently entered */
11    int duplicate;        /* flag for duplicate values */
12    int value;            /* current value */
13
14    printf( "Enter 20 integers between 10 and 100:\n" );
15
16    /* get 20 integers from user */
17    for ( i = 0; i <= MAX - 1; i++ ) {
18       duplicate = 0;
19       scanf( "%d", &value );
20
21       /* test if integer is a duplicate */
22       for ( j = 0; j < k; j++ ) {
23
24          /* if duplicate, raise flag and break loop */
25          if ( value == a[ j ] ) {
26             duplicate = 1;
27             break;
28          } /* end if */
```

Check Duplication (2/2)

```
29
30          } /* end for */
31
32          /* if number is not a duplicate, enter it in array */
33          if ( !duplicate ) {
34             a[ k++ ] = value;
35          } /* end if */
36
37       } /* end for */
38
39       printf( "\nThe nonduplicate values are:\n" );
40
41       /* display array of nonduplicates */
42       for ( i = 0; a[ i ] != 0; i++ ) {
43          printf( "%d  ", a[ i ] );
44       } /* end for */
45
46       printf( "\n" );
47
48       return 0; /* indicate successful termination */
49
50    } /* end main */
```

```
Enter 20 integers between 10 and 100:
10 11 12 13 14 15 16 17 18 19 20 21 10 11 12 13 14 15 16 17

The nonduplicate values are:
10  11  12  13  14  15  16  17  18  19  20  21
```