ALGORITHMIZATION AND PROGRAMMING

Part 2
STRUCTURED DATA PROGRAMMING

Methodical instructions for laboratory training and exercises

Odessa 2019
Compilers: Y. V. Prokop, O. G. Trofymenko, L. N. Bukata

These instructions for laboratory training and exercises contain theoretical information with examples of programs in C++ and variants of individual problems. Methodical instructions will be useful for students of the Academy of Telecommunications who are studying in English, fixing theoretical material, preparing to laboratory training and exercises in the discipline of "Algorithmization and programming" during the first semester.

It is intended for the acquisition of skills in operation on a personal computer and programming by students of the academy studying in English, with the purpose of further usage of these skills in daily professional work. Also, it will be useful for users of personal computers wishing to learn programming in Visual C++ environment.

APPROVED
by the sub-faculty IT meeting and recommended for publication.

Minutes № from

APPROVED
by the Methodology Council
the Academy of Communications.

Minutes № from
Introduction

The Visual C++ language and development tools help you to develop native Universal Windows apps, native desktop and server applications, cross-platform libraries that run on Android and iOS as well as Windows, and managed apps that run on the .NET Framework.

Visual C++ supports two distinct, but closely related flavors of the C++ language, ISO/IEC standard C++ (native C++) and C++/CLI. We’ll cover both of them in this course.

There are two principal types of applications, which we are going to create: console application (it executes in “black screen” in character mode) and application with graphical interface with Windows forms.

There’s a lot of code even in a simple Windows program, and it’s very important not to be distracted by the complexities of Windows while learning the ins and outs of C++.

These instructions present a brief theoretical information, examples of how to create software projects using Visual C++ for the calculation of linear, branched and cyclic structures, control questions and variants of individual problems to the nine laboratory exercises. Each of the proposed laboratory exercises contains problems of different complexity levels. The student himself or under the teacher’s instructions selects problem of this or that complexity level according to a student’s log number. Problems of basic level are obvious.

Before realization of the laboratory training the student should do the following:

– to select individual tasks with the teacher;
– to study relevant sections of the theoretical course in accordance with lecture notes and academic literature;
– to develop flowcharts for solving the individual problems;
– to write code in C++;
– to prepare a report of the laboratory exercise and submit it to the teacher for checking.

The students may perform Laboratory exercise only after they prepared a report.

The content of the report of the laboratory exercise:

– the title of the topic and the purpose of the laboratory exercise;
– answers to control questions;
– flowcharts for solving of the individual problems;
– programming code of his problems in C++;
– results of calculations on the computer.

The correctness of execution of the program and obtained results should be checked by the teacher.
Lab № 6

One-dimensional array. Operating with arrays in functions

Goal: to get practical skills of operating with 1-dimensional arrays in Visual C++.

Examples of programs

Example 6.1. Enter array of 8 real numbers and calculate the sum of elements.
The program code:
#include <iostream>
using namespace std;
int main()
{
  double a[8];       // Declaration array
  double sum = 0;
  cout << "Enter 8 numbers" << endl;
  for (int i = 0; i < 8; i++)
  { cin >> a[i];     // Input element of array
    sum += a[i];    // The sum of elements
  }
  cout << "Array sum = " << sum << endl;
  system("pause");
  return 0;
}
Results:
Enter 10 numbers
5  -2  8  1.7  -1  -4  -4.2  7
Array sum = 10.5

Example 6.2. Enter array of 10 integer numbers and calculate the product of non-zero elements.
The program code:
#include <iostream>
using namespace std;
int main()
{
  const int n = 10;   // The count of elements
  int a[n], p = 1;
  cout << "Enter " << n << " integer numbers\n";
  for (int i = 0; i < n; i++)
  { cin >> a[i];     // Input element of array
    if (a[i] != 0)  // If element is non-zero
      p *= a[i];   // that calculate the product
  }
  cout << "Product p = " << p << endl;
  system("pause");
  return 0;
}
Results:
Enter 10 integer numbers
2 8 -4 0 1 0 3 -2 0 1
Product p = 384
Example 6.3. Enter array of 10 real numbers and calculate the minimal element and its index.

The program code:
```
#include <iostream>
using namespace std;
int main()
{
    const int n = 10; int i, ind;
    double a[n], min;
    cout << "Enter " << n << " real numbers:\n";
    for (i = 0; i < n; i++)
        cin >> a[i];
    min = a[0]; ind = 0;
    for (i = 1; i < n; i++)
        if (a[i] < min)
            { min = a[i];
              ind = i;
            }
    cout << "min = " << min;
    cout << " ind = " << ind << endl;
    system("pause");
    return 0;
}
```

Results:
Enter 9 real numbers:
13 25.6 -8 3 1 -14 5 -1 -4
min = -14 ind = 5

Example 6.4. Enter array of 12 integer numbers and calculate the quantity of odd elements from the interval (~10, 30).

The program code:
```
#include <iostream>
using namespace std;
int main()
{
    int a[12], i, k = 0;
    cout << "Enter 12 integer numbers:\n";
    for (i = 0; i < 12; i++)
        { cin >> a[i];
          if (a[i] % 2 != 0 && a[i] > -10 && a[i] < 30)
              k++;
        }
    cout << "k = " << k << endl;
    system("pause");
    return 0;
}
```

Results:
Enter 12 integer numbers:
29 -817 44 16 -75 1 230 -83 19 -34 8 5
k = 4
**Example 6.5.** Fill array of 10 elements under the formula:

\[ a_i = (-1)^i \frac{\sin(i^2)}{\sin(i+1)} \]

and define the biggest negative element of array.

The program code:
```c
#include <math.h>

int main()
{
    double a[10], max = 0; int i;
    printf("Array: \n");
    for (i = 0; i < 10; i++)
    {
        a[i] = pow(-1,i) * sin(i*i) / sin(i+1);
        printf("%5.2f", a[i]);
        if (a[i] < 0)
            if (max == 0 || max < 0 && max < a[i])
                max = a[i];
    }
    printf("\nmax = %5.2f\n", max);
    system("pause");
    return 0;
}
```

Results:
Array:
0.00 -0.93 -5.36 0.54 0.30 -0.47 -1.51 0.96 2.23 -1.16
max = -0.47

**Example 6.6.** Enter array of 10 integers. Create a new array with elements computed by dividing each element of the original array by the sum of its elements with odd indices.

The program code:
```c
#include <iostream>
using namespace std;

int main()
{
    const int N = 10;
    int a[N], i, sum = 0;
    double b[N];
    cout << "Enter 10 integer numbers:\n";
    for (i = 0; i < N; i++)
        cin >> a[i];
    for (i = 1; i < N; i += 2)
        sum += a[i];
    cout << "Sum = " << sum << endl;
    cout << "New array: \n";
    for (i = 0; i < N; i++)
    {
        b[i] = 1.*a[i] / sum;
        cout << b[i] << " ";
    }
    cout << endl;
}
```
system("pause");
    return 0;
}

Results:
Enter 10 integer numbers:
-3 9 12 6 -5 -2 4 7 0 1
Sum = 21
New array:
-0.1428 0.4285 0.5714 0.2857 -0.2380 -0.09523 0.1904 0.3333 0 0.0476

Example 6.7. Enter array of 12 real numbers. Sort it on ascending order.
The program code:
#include <iostream>
using namespace std;
int main()
{  
    const int n = 12;
    double a[n];
    cin >> "Enter " << n << " real numbers:\n";
    for (int i = 0; i < n; i++)
        cin >> a[i];
    for (int i = 0; i < n - 1; i++)
        for (int j = 0; j < n - 1; j++)
            if (a[j] > a[j + 1])
            {
                double temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
    cout << "Sorted array:\n";
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << endl;
    system("pause");
    return 0;
}

Results:
Enter 12 real numbers:
5 0 -2 -6 1 -3 8 9 23 5 -1 3
Sorted array:
-6 -3 -2 -1 0 1 3 5 5 8 9 23

Step-by-step array changing:
Inputted array:
5 0 -2 -6 1 -3 8 9 23 5 -1 3
Pass #: 0
0 -2 -6 1 -3 5 8 9 5 -1 3 23
Pass #: 1
-2 -6 0 -3 1 5 8 5 -1 3 9 23
Pass #: 2
-6 -2 -3 0 1 5 5 -1 3 8 9 23
Example 6.8. Enter array of 8 real numbers and delete all negative elements from it.

The program code:
```cpp
#include <iostream>
using namespace std;
int main()
{
    double a[8]; int i, j, n = 8;
    cout << "Enter array of 8 real numbers: \n ";
    for (i = 0; i < n; i++)
        cin >> a[i];
    for (i = 0; i < n; i++)
        if (a[i] < 0)
            {
            for (j = i; j < n; j++)
                a[j] = a[j + 1];
            n--; i--;
        }
    cout << "\nChanged array consists of " << n << " elements:\n ";
    for (i = 0; i < n; i++)
        cout << a[i] << "\t";
    cout << endl;
    system("pause");
    return 0;
}
```

Results:
Enter array of 8 real numbers:
6 2.5 -3 7 1 0.1 -2 5

Changed array consists of 6 elements:
6 2.5 7 1 0.1 5

Example 6.9. Create a function for calculating an average of non-zero elements of an integer array. Use this function to find an average of non-zero
elements for array of 10 integers: first for all elements, and then for this first half of the array.

The flowcharts of the avg function and of the main function:

![Flowchart of function avg()](image1)

![Flowchart of the main function](image2)

The program code:

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

double avg(int a[], int n)
{
    double s = 0; int i, kol = 0;
    for (i = 0; i < n; i++)
        if (a[i] != 0)
            {
                s += a[i];
                kol++;
            }
    if (kol) s /= kol;
    return s;
}

int main()
{
    int a[10], i;
    cout << "Enter 10 integer numbers:\n";
```
for (i = 0; i < 10; i++) cin >> a[i];
cout << "\nAverage of 10 elements = " << avg(a, 10) << endl;
cout << "\nAverage of the first 5 elements = " << avg(a, 5) << endl;
system("pause");
return 0;
}

Results:
Enter 10 integer numbers:
-3 5 0 -8 0 2 3 -1 0 6
Average of 10 elements = 0.571429
Average of the first 5 elements = -2

Example 6.10. Create a function to define maximum and minimum elements of an array of 10 real numbers and call the function for the entered array.
The flowcharts of the MinMax function and of the main function:

Flowchart of function MinMax()

Flowchart of the main function

The program code:
#include <iostream>

using namespace std;

void MinMax(double a[], int n, double& min, double& max)
{
    min = max = a[0];
    for (int i = 1; i < n; i++)
    {
        if (a[i] < min) min = a[i];
        if (a[i] > max) max = a[i];
    }
}
int main()
{
    double a[10], min, max;
    cout << "Enter 10 numbers:\n";
    for (int i = 0; i < 10; i++)
        cin >> a[i];
    MinMax(a, 10, min, max);
    cout << "Minimum: " << min << endl;
    cout << "Maximum: " << max << endl;
    system("pause");
    return 0;
}

Results:
Enter 10 numbers:
5.3 2 -8 4 -3 7 0.5 -1.04 2.5 3
Minimum: -8
Maximum: 7

Control questions

1) What is array?
2) Write the declaration of array of 25 double numbers.
3) Assign zero to the first element of array A of 25 double numbers.
4) Which variable is the sum of array?
   a) s1=0;  b) s2=1;  c) s3=0;
   for(i=0;i<10;i++) for(i=0;i<10;i++) for(i=0;i<10;i++)
       s1 += a[i];  s2 *= a[i];  if(a[i]>0) s3++;
5) Which variable is the number of positive elements of array?
   a) S1=0;  b) S2=1;  c) S3=0;
   for(i=0;i<10;i++) for(i=0;i<10;i++) for(i=0;i<10;i++)
       S1 += a[i];  S2 *= a[i];  if(a[i]>0) S3++;
6) Which variable is the biggest element of array?
   a) S1=0;  b) S2=a[0];  c) S3=0;
   for(i=0;i<10;i++) for(i=1;i<10;i++) for(i=0;i<10;i++)
       S1 += a[i];  if(a[i]>S2)S2=a[i];  if(a[i]>0)S3=S3+1;
7) How are arrays passed to functions?
8) What type has a function, which sorts array? Why?
9) What information do the following headers give about the functions:
   int fun(double x[], int n);
   void fun(double x[], int n);
   double fun(double x[], int n);
   double fun(double x[], int n, double& z);

Individual task

1. Create projects with array (Tables 6.1 and 6.2). Draw the flowcharts.
2. Create project with function (Table 6.3). Create function with one or two
   results and use it in the main function. Draw the flowcharts.
### Table 6.1

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Size of array</th>
<th>Data type</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>int</td>
<td>Calculate the quantity and the sum of even elements of array</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>int</td>
<td>Calculate an average of positive elements</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>int</td>
<td>Calculate factorial of the last element of array</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>double</td>
<td>Calculate the product of elements which values are smaller than 6</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>int</td>
<td>Calculate an average of odd elements</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>int</td>
<td>Calculate the sum of elements which are multiple to 3</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>double</td>
<td>Calculate the sum of elements which absolute value is not bigger than 10</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>int</td>
<td>Calculate a quantity of elements from (0, 7]</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>double</td>
<td>Calculate an average of elements with absolute value bigger than 12</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>double</td>
<td>Calculate a product of elements from (-4, 5)</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>int</td>
<td>Calculate an average of the minimum and the maximum of array elements</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>double</td>
<td>Output the quantity of elements which values are greater than the value of the first element</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>int</td>
<td>Find the indices of the minimum and the maximum of array</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>double</td>
<td>Calculate the sum of array elements, which values belong to the interval [3, 6]</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>int</td>
<td>Calculate the product of odd array elements</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>double</td>
<td>Find minimum and maximum of array elements</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>int</td>
<td>Calculate the product of odd elements with even indices</td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td>double</td>
<td>Find the difference between the first element and the sum of array</td>
</tr>
</tbody>
</table>

### Table 6.2

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Size of array</th>
<th>Data type</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>double</td>
<td>Swap the smallest element and the next to the last element</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>int</td>
<td>Calculate the sum of positive odd elements and change even elements with this sum</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>int</td>
<td>Define, whether array is sorted in ascending order</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>double</td>
<td>Calculate factorial of the first element of array with the value smaller than 8</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>int</td>
<td>Swap the biggest element and the first element</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>double</td>
<td>Calculate the new array as a difference between elements of the inputted array and their average</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>int</td>
<td>Change all zero elements with the last element</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>double</td>
<td>Calculate the new array as a sum of elements of the inputted array and its minimum</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>int</td>
<td>Change all elements with odd indexes to array average</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>int</td>
<td>Define, whether array is sorted in descending order</td>
</tr>
<tr>
<td>№ var</td>
<td>Size of array</td>
<td>Data type</td>
<td>Problems</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>double</td>
<td>Calculate the new array as a sum between elements of the inputted array and its maximum</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>int</td>
<td>Swap the maximal element and the minimal element</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>int</td>
<td>Calculate the factorial of the absolute value of maximal element</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>double</td>
<td>Swap the first half of the array with the second half</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>double</td>
<td>Change the minimal and maximal elements to the average of all elements</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>double</td>
<td>Swap the pairs of elements: 0&lt;sup&gt;th&lt;/sup&gt; and 1&lt;sup&gt;st&lt;/sup&gt;, 2&lt;sup&gt;nd&lt;/sup&gt; and 3&lt;sup&gt;rd&lt;/sup&gt; etc.</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>int</td>
<td>Increase all even elements by 3, change all zero elements to array maximum</td>
</tr>
<tr>
<td>18</td>
<td>7</td>
<td>int</td>
<td>Swap the first and the last positive elements</td>
</tr>
</tbody>
</table>

**Table 6.3**

<table>
<thead>
<tr>
<th>№ var</th>
<th>Size of array</th>
<th>Data type</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>int</td>
<td>Calculate the factorial of the first element of array which is smaller than 10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>int</td>
<td>Calculate the number of array elements which are placed after the first zero element</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>int</td>
<td>Calculate the factorial of the first positive element of array</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>double</td>
<td>Calculate the smallest negative element of array</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>int</td>
<td>Calculate the number and the sum of even elements of array</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>double</td>
<td>Calculate the sum of array elements which absolute value is not bigger than 10</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>int</td>
<td>Calculate the sum and the number of array elements which are multiple to 3</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>double</td>
<td>Calculate the biggest positive element of array</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>int</td>
<td>Calculate the factorial of the last even element of array</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>double</td>
<td>Calculate the quantity of elements which are bigger than average</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>double</td>
<td>Find the indices of the minimal and the maximal elements of the array</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>int</td>
<td>Calculate the absolute value of the sum of all negative elements, and the sum of all positive</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>double</td>
<td>Calculate the sum of negative elements placed after the maximal element of the array</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>int</td>
<td>Calculate the difference between the sum of all positive elements and the sum of absolute values of all negative elements</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>double</td>
<td>Calculate the product of single-digit elements of an array</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>int</td>
<td>Define, whether array is sorted in ascending order</td>
</tr>
<tr>
<td>17</td>
<td>7</td>
<td>double</td>
<td>Define, whether array contains more positive elements than negative ones</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>int</td>
<td>Calculate the sum of even elements in the second half of array</td>
</tr>
</tbody>
</table>
Lab № 7

**Two-dimensional arrays**

**Goal:** to get practical skills of operating with 2-dimensional arrays in Visual C++.

**Examples of programs**

**Example 7.1.** Enter a matrix of 3×4 integers and compute the quantity of negative elements.

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

int main()
{
    int a[3][4], i, j; int k = 0;
    cout << "Input matrix 3x4: " << endl;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 4; j++)
            cin >> a[i][j]; // Input element of matrix
    for (i = 0; i < 3; i++)
        for (j = 0; j < 4; j++)
            if (a[i][j] < 0) k++;
    cout << "\nQuantity of negative elements - " << k << endl;
    system("pause");
    return 0;
}
```

**Results:**

Input matrix 3x4:

```
1 8 0 -1
3 4 -5 2
0 -1 -7 5
```

Quantity of negative elements - 4

**Example 2.** Enter the matrix 4×6 of real numbers and swap the maximal and minimal elements.

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

int main()
{
    double a[4][6], min, max;
    int i, j, imin, jmin, imax, jmax;
    cout << "Input matrix 4x6: " << endl;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 6; j++)
            cin >> a[i][j];
    min = max = a[0][0];
    imin = jmin = imax = jmax = 0;
```
for (i = 0; i < 4; i++)
  for (j = 0; j < 6; j++)
  {
    if (a[i][j] < min)
    {
      min = a[i][j];
      imin = i;
      jmin = j;
    }
    if (a[i][j] > max)
    {
      max = a[i][j];
      imax = i;
      jmax = j;
    }
  }
  a[imin][jmin] = max;
  a[imax][jmax] = min;
  cout << "\nMatrix with min and max swapped:" << endl;
  for (i = 0; i < 4; i++)
  {
    for (j = 0; j < 6; j++)
      cout << a[i][j] << "\t";
    cout << endl;
  }
  system("pause");
  return 0;
}

Results:
Input matrix 4x6:
6 9 -3 0 0 1
1 1 2 -3 0 -1
8 -3 -4 0 0 1
5 -2 -7 0 1 0

Matrix with min and max swapped:
6 -7 -3 0 0 1
1 1 2 -3 0 -1
8 -3 -4 0 0 1
5 -2 9 0 1 0

Example 7.3. Enter a matrix 5x7 of integers and create a vector of sums of the negative elements of matrix columns.

#include <iostream>
using namespace std;
int main()
{
  int a[5][7], x[7], i, j, s;
  cout << "Input matrix 5x7:" << endl;
  for (i = 0; i < 5; i++)
    for (j = 0; j < 7; j++)
      cin >> a[i][j];
  cout << "Vector:\n";
Lab 7. Two-dimensional arrays

for (j = 0; j < 7; j++)
{
    s = 0;
    for (i = 0; i < 5; i++)
        if (a[i][j] < 0)
            s += a[i][j];
    x[j] = s;
}
for (i = 0; i < 7; i++)
cout << x[i] << "\t";
cout << endl;

Results:
Input matrix 5x7:
7 1 0 0 2 -4 1
4 2 -2 0 0 0 0
-1 2 3 4 5 -1 -1
-2 -4 -5 0 -1 -1 6
0 0 1 1 2 2 3
Vector:
-3 -4 -7 0 -1 -6 -1

Example 7.4. Input matrix 7x5 of integers. Form a vector of averages of even elements of odd (1, 3, and 5) matrix rows.

#include <iostream>
using namespace std;
int main()
{
    int i, j, a[7][5]; double v[3];
cout << "Input matrix 7x5:"<<endl;
for (i = 0; i < 7; i++)
    for (j = 0; j < 5; j++) cin >> a[i][j];
cout << "\n\nVector" << endl;
int kol, k = 0;
for (i = 1; i < 7; i += 2)
{
    kol = 0; v[k] = 0;
    for (j = 0; j < 5; j++)
        if (a[i][j] % 2 == 0 & a[i][j])
            { v[k] += a[i][j];
              kol++;
            }
    if (kol>0) v[k] /= kol;
cout << v[k] << "\n\t";
k++;
}
system("pause");
return 0;
}
Lab 7. Two-dimensional arrays

Results:
Input matrix 7x5:
5 0 0 -2 3
8 7 6 -2 0
1 2 0 0 9
1 0 2 0 3
7 -3 -4 -5 0
1 -4 -8 0 2
5 4 3 1 2
Vector
4 2 -3.33333

Example 7.5. Input matrix 4×3 of real numbers and calculate with the function the minimum element and its indexes.

```
#include <iostream>
using namespace std;
double MIN(double a[4][3], int& imin, int& jmin)
{
    double min = a[0][0]; imin = jmin = 0;
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 3; j++)
            if (a[i][j] < min)
            {
                min = a[i][j]; imin = i; jmin = j;
            }
    return min;
}

int main()
{
    double a[4][3], m;
    int i, j, ind_i, ind_j;
cout << " Input matrix 4x3:" << endl;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 3; j++) cin >> a[i][j];
m = MIN(a, ind_i, ind_j);
```
Lab 7. Two-dimensional arrays

cout << "\n Min = " << m << " in " << ind_i+1 << "-th row and " << ind_j+1 << "-th column" << endl;
system("pause");
return 0;
}

Results:
Input matrix 4x3:
5 -8 0
-1 2 2
-5 9 2
1 2 3
Min = -8
in 1-th row and 2-th column

Example 7.6. Input 5×5 matrix of real numbers. Use the function to change the elements of the main diagonal with the arithmetic mean of the corresponding row.

```cpp
#include <iostream>
using namespace std;
void CHANGE(double a[5][5])
{
    for (int i = 0; i < 5; i++)
    {
        double s = 0;
        for (int j = 0; j < 5; j++)
            s += a[i][j];
        a[i][i] = s / 5;
    }
}

int main()
{
    double a[5][5];
    int i, j;
```
cout << " Input matrix 5x5:" << endl;
for (i = 0; i < 5; i++)
    for (j = 0; j < 5; j++)
        cin >> a[i][j];
CHANGE(a);
cout << " Matrix with changes" << endl;
for (i = 0; i < 5; i++)
{
    for (j = 0; j < 5; j++)
        cout << a[i][j] << "\t";
    cout << endl;
}
system("pause");
return 0;
}

Results:
Input matrix 5x5:
1 2 3 4 5
1 0 2 0 3
1 1 2 2 3
0 0 1 2 3
1 1 1 1 4

Matrix with changes
3 2 3 4 5
1 1.2 2 0 3
1 1 1.8 2 3
0 0 1 1.2 3
1 1 1 1 1.6

Example 7.7. Fill matrix of 5×10 with random integers and create vector of maximal elements of the columns.

#include <iostream>
using namespace std;
void vec_f(int a[5][10], int x[10])
{
    int max;
    for (int j = 0; j < 10; j++)
    {
        max = a[0][j];
        for (int i = 0; i < 5; i++)
            if (max < a[i][j]) max = a[i][j];
        x[j] = max;
    }
}
int main()
{
    int a[5][10], x[10];
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 10; j++)
            a[i][j] = rand() % 100 - 50;
cout << "Matrix: " << endl;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 10; j++)
            cout << a[i][j] << "\t";
        cout << endl;
    }
}
vec_f(a, x);
cout << "Vector: " << endl;
for (int j = 0; j < 10; j++)
    cout << x[j] << " ";
system("pause");
return 0;
}

Results:
Matrix:
-9 17 -16 -50 19 -26 28 8 12 14
-45 -5 31 -23 11 41 45 -8 -23 -14
41 -46 -48 3 42 32 -29 -34 -32 45
-3 -24 21 -12 19 -38 17 49 -15 44
-47 -39 -28 -17 23 14 -9 -39 3 18
Vector:
41 17 31 3 42 41 45 49 12 45

Control questions

1) Which of the following declarations of two-dimensional arrays are incorrect? Why?
   a) int C[1..5, 1..5];  b) float C[1..5][1..5];
   c) float C[5][5];  d) int C[5][5];

2) How are elements of two-dimensional array allocated in memory?
3) Declare matrix S of integer numbers with size 7x3.
4) Which of the following fragments calculates the sum of elements of the
   main diagonal of integer matrix 5x5?
   a) for(i=0, s=0; i<5; i++) s++;
   b) for(i=0, s=0; i<5; i++) s+=A[i][i];
   c) for(i=0, s=0; i<5; i++) for(j=0; j<5; j++) s+=A[i][j];
   d) for(i=0, s=0; i<5; i++) A[i][i]=0;

5) How matrices are passed to functions as parameters?
6) What type has a function which for matrix of integer numbers:
   a) finds the biggest element?
   b) calculates the average?
   c) creates vector?
   d) replaces some elements in this matrix?
7) Which declarations are incorrect and why?
   void fun(int a[][4], int m);
   void fun(int a[4][]), int m);
   void fun(int a[3][4]);
   void fun(int a[][4], int m, int n);

Individual task

Create projects and flowcharts for problems in Tables 7.1-7.5. Process matrices
in functions (Tables 7.3-7.5).
<table>
<thead>
<tr>
<th>№ var</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input real matrix with 5 rows and 4 columns. Calculate the number of positive, negative and zero elements</td>
</tr>
<tr>
<td>2</td>
<td>Input integer matrix with 4 rows and 5 columns. Find the smallest element and its indexes</td>
</tr>
<tr>
<td>3</td>
<td>Input real matrix with 5 rows and 5 columns. Find the smallest element of the main diagonal and the index of its row</td>
</tr>
<tr>
<td>4</td>
<td>Input real matrix with 4 rows and 4 columns. Swap elements of the first row and elements of the secondary diagonal</td>
</tr>
<tr>
<td>5</td>
<td>Input integer matrix with 3 rows and 5 columns. Change all negative elements to zero</td>
</tr>
<tr>
<td>6</td>
<td>Input real matrix with 6 rows and 6 columns. Find the smallest and the biggest elements</td>
</tr>
<tr>
<td>7</td>
<td>Input real matrix with 5 rows and 5 columns. Calculate the sum of elements of the secondary diagonal</td>
</tr>
<tr>
<td>8</td>
<td>Input integer matrix with 6 rows and 3 columns. Output indices of positive elements</td>
</tr>
<tr>
<td>9</td>
<td>Input double matrix with 3 rows and 6 columns. Find the biggest element of the second row</td>
</tr>
<tr>
<td>10</td>
<td>Input real matrix with 4 rows and 4 columns. Find the biggest element of the main diagonal and the index of its column</td>
</tr>
<tr>
<td>11</td>
<td>Input integer matrix with 5 rows and 5 columns. Swap elements of the first column and elements of the secondary diagonal</td>
</tr>
<tr>
<td>12</td>
<td>Input integer matrix with 3 rows and 4 columns. Change all positive elements to the average of all elements</td>
</tr>
<tr>
<td>13</td>
<td>Input real matrix with 6 rows and 6 columns. Find indexes of the smallest and the biggest elements in the last column</td>
</tr>
<tr>
<td>14</td>
<td>Change the zero elements of integer matrix 5×5 to its maximum element</td>
</tr>
<tr>
<td>15</td>
<td>Swap the first and the last negative elements in the real matrix 7×3</td>
</tr>
<tr>
<td>16</td>
<td>Input integer matrix 6×5. Swap the first and the last even elements</td>
</tr>
<tr>
<td>17</td>
<td>Input double matrix 5×4. Define the index of the row with minimal element</td>
</tr>
<tr>
<td>18</td>
<td>Input double matrix 5×4. Define the index of the column with maximal element</td>
</tr>
<tr>
<td>№ var</td>
<td>Individual problem</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>1</td>
<td>Input integer matrix with 5 rows and 6 columns. Calculate vector's element as the sums of odd columns</td>
</tr>
<tr>
<td>2</td>
<td>Input real matrix with 4 rows and 4 columns. Calculate vector's element as the scalar products of matrix rows and the last column</td>
</tr>
<tr>
<td>3</td>
<td>Input integer matrix with 6 rows and 4 columns. Calculate vector's element as the products of odd elements of even rows</td>
</tr>
<tr>
<td>4</td>
<td>Input integer matrix with 4 rows and 4 columns. Calculate vector's element as the scalar products of elements of the first row and columns of the matrix</td>
</tr>
<tr>
<td>5</td>
<td>Input real matrix with 6 rows and 4 columns. Calculate vector's element as the matrix column with the smallest sum of elements</td>
</tr>
<tr>
<td>6</td>
<td>Input integer matrix with 4 rows and 5 columns. Calculate vector's element as the products of even elements of odd columns</td>
</tr>
<tr>
<td>7</td>
<td>Input integer matrix with 4 rows and 5 columns. Calculate vector's element as the matrix row with the biggest sum of elements</td>
</tr>
<tr>
<td>8</td>
<td>Input integer matrix with 5 rows and 4 columns. Calculate vector's element as the averages of even columns</td>
</tr>
<tr>
<td>9</td>
<td>Input real matrix with 4 rows and 4 columns. Calculate vector's element as the scalar products of matrix rows and the last row</td>
</tr>
<tr>
<td>10</td>
<td>Input integer matrix with 5 rows and 3 columns. Calculate vector's element as the products of negative elements of even rows</td>
</tr>
<tr>
<td>11</td>
<td>Input integer matrix with 4 rows and 4 columns. Calculate vector's element as the scalar products of elements of the first column and rows of the matrix</td>
</tr>
<tr>
<td>12</td>
<td>Input real matrix with 6 rows and 5 columns. Calculate vector's element as the matrix column with the biggest sum of elements</td>
</tr>
<tr>
<td>13</td>
<td>Input integer matrix with 4 rows and 5 columns. Calculate vector's element as the averages of even elements of odd columns</td>
</tr>
<tr>
<td>14</td>
<td>Input integer matrix 3×5 with elements from 0 to 9. Calculate vector's element as the percentage of each of these numbers in the matrix</td>
</tr>
<tr>
<td>15</td>
<td>Input double matrix 6×7. Define the row with the minimum sum and swap this row with the first row</td>
</tr>
<tr>
<td>16</td>
<td>Input two matrices of real numbers 4×5. Swap the rows of the matrices containing maximum elements</td>
</tr>
<tr>
<td>17</td>
<td>Input double matrix 4×5. Calculate vector's element as the average of the columns containing the maximum and minimum elements.</td>
</tr>
<tr>
<td>18</td>
<td>Input two matrices of real numbers 4×5. Swap the columns of the matrices containing maximum elements</td>
</tr>
<tr>
<td>№ var.</td>
<td>Size of array</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>1</td>
<td>5×5</td>
</tr>
<tr>
<td>2</td>
<td>4×4</td>
</tr>
<tr>
<td>3</td>
<td>6×4</td>
</tr>
<tr>
<td>4</td>
<td>3×3</td>
</tr>
<tr>
<td>5</td>
<td>4×5</td>
</tr>
<tr>
<td>6</td>
<td>3×5</td>
</tr>
<tr>
<td>7</td>
<td>5×3</td>
</tr>
<tr>
<td>8</td>
<td>5×5</td>
</tr>
<tr>
<td>9</td>
<td>4×4</td>
</tr>
<tr>
<td>10</td>
<td>6×4</td>
</tr>
<tr>
<td>11</td>
<td>5×5</td>
</tr>
<tr>
<td>12</td>
<td>3×5</td>
</tr>
<tr>
<td>13</td>
<td>4×3</td>
</tr>
<tr>
<td>14</td>
<td>6×4</td>
</tr>
<tr>
<td>15</td>
<td>5×5</td>
</tr>
<tr>
<td>16</td>
<td>4×5</td>
</tr>
<tr>
<td>17</td>
<td>3×5</td>
</tr>
<tr>
<td>18</td>
<td>4×6</td>
</tr>
</tbody>
</table>
### Functions for changing of matrix elements

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Size of array</th>
<th>Data type</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4×3</td>
<td>Integer</td>
<td>Change even elements with 0</td>
</tr>
<tr>
<td>2</td>
<td>6×4</td>
<td>Double</td>
<td>Swap the smallest and the biggest matrix elements</td>
</tr>
<tr>
<td>3</td>
<td>4×4</td>
<td>Integer</td>
<td>Swap elements of the main and secondary diagonals</td>
</tr>
<tr>
<td>4</td>
<td>4×5</td>
<td>Double</td>
<td>Change all negative elements to the value of the smallest element</td>
</tr>
<tr>
<td>5</td>
<td>3×5</td>
<td>Integer</td>
<td>Calculate the sum of positive odd elements and change the elements in the corners of matrix to this sum</td>
</tr>
<tr>
<td>6</td>
<td>5×3</td>
<td>Integer</td>
<td>Change all non-zero elements to the value of the smallest element</td>
</tr>
<tr>
<td>7</td>
<td>5×3</td>
<td>Integer</td>
<td>Transpose matrix</td>
</tr>
<tr>
<td>8</td>
<td>4×4</td>
<td>Integer</td>
<td>Change odd elements to the sum of elements above the main diagonal</td>
</tr>
<tr>
<td>9</td>
<td>6×4</td>
<td>Double</td>
<td>Swap the first and the biggest matrix elements</td>
</tr>
<tr>
<td>10</td>
<td>4×4</td>
<td>Integer</td>
<td>Swap elements of the main diagonal and the last column</td>
</tr>
<tr>
<td>11</td>
<td>5×3</td>
<td>Integer</td>
<td>Change all odd elements to the value of the smallest element</td>
</tr>
<tr>
<td>12</td>
<td>5×5</td>
<td>Double</td>
<td>Arrange the elements of the main diagonal in reverse order</td>
</tr>
<tr>
<td>13</td>
<td>3×4</td>
<td>Integer</td>
<td>Change elements that are multiples of 5 to the maximal matrix element</td>
</tr>
<tr>
<td>14</td>
<td>3×6</td>
<td>Double</td>
<td>Swap elements of the first and last columns</td>
</tr>
<tr>
<td>15</td>
<td>5×5</td>
<td>Integer</td>
<td>Change all zero matrix elements to the maximum</td>
</tr>
<tr>
<td>16</td>
<td>4×6</td>
<td>Integer</td>
<td>Change all even elements of the matrix to the value of the last element</td>
</tr>
<tr>
<td>17</td>
<td>5×4</td>
<td>Double</td>
<td>Swap elements of the first and last rows</td>
</tr>
<tr>
<td>18</td>
<td>5×5</td>
<td>Integer</td>
<td>Change all negative elements under the main diagonal to 0</td>
</tr>
</tbody>
</table>
### Function for creation of vector

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Size of array</th>
<th>Data type</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3×5</td>
<td>Double</td>
<td>Calculate vector's element as the squares of the smallest elements in columns</td>
</tr>
<tr>
<td>2</td>
<td>5×3</td>
<td>Integer</td>
<td>Calculate vector's element as the averages of elements in matrix rows</td>
</tr>
<tr>
<td>3</td>
<td>3×4</td>
<td>Double</td>
<td>Calculate vector's element as the sums of elements with values not bigger than 10, in columns</td>
</tr>
<tr>
<td>4</td>
<td>6×4</td>
<td>Integer</td>
<td>Calculate vector's element as the averages of positive elements in the matrix rows</td>
</tr>
<tr>
<td>5</td>
<td>4×3</td>
<td>Double</td>
<td>Calculate vector's element as the sums of positive elements in the matrix rows</td>
</tr>
<tr>
<td>6</td>
<td>5×4</td>
<td>Integer</td>
<td>Calculate vector's element as the averages of two-digit numbers in matrix columns</td>
</tr>
<tr>
<td>7</td>
<td>4×6</td>
<td>Double</td>
<td>Calculate vector's element as the sums of elements in the odd matrix columns</td>
</tr>
<tr>
<td>8</td>
<td>3×5</td>
<td>Double</td>
<td>Calculate vector's element as the maximums of matrix columns</td>
</tr>
<tr>
<td>9</td>
<td>5×3</td>
<td>Integer</td>
<td>Calculate vector's element as the sums of positive elements in the matrix rows</td>
</tr>
<tr>
<td>10</td>
<td>3×4</td>
<td>Double</td>
<td>Calculate vector's element as the products of columns elements with values from (-3, 4)</td>
</tr>
<tr>
<td>11</td>
<td>4×4</td>
<td>Double</td>
<td>Calculate vector's element as the squares of the main diagonal elements</td>
</tr>
<tr>
<td>12</td>
<td>5×5</td>
<td>Double</td>
<td>Calculate vector's element as the elements of the secondary diagonal</td>
</tr>
<tr>
<td>13</td>
<td>3×6</td>
<td>Integer</td>
<td>Calculate vector’s element as the products of nonzero elements in matrix columns</td>
</tr>
<tr>
<td>14</td>
<td>5×5</td>
<td>Double</td>
<td>Calculate vector's element as the sums of the absolute values of the negative elements in the matrix rows</td>
</tr>
<tr>
<td>15</td>
<td>4×6</td>
<td>Integer</td>
<td>Calculate vector's element as the average of the first and the last elements in the matrix rows</td>
</tr>
<tr>
<td>16</td>
<td>3×4</td>
<td>Double</td>
<td>Calculate vector's element as the smallest elements in the matrix columns</td>
</tr>
<tr>
<td>17</td>
<td>5×5</td>
<td>Integer</td>
<td>Calculate vector's element as the sum of elements of the main and the secondary diagonal</td>
</tr>
<tr>
<td>18</td>
<td>5×4</td>
<td>Integer</td>
<td>Calculate vector's element as the average of the even elements in the matrix rows</td>
</tr>
</tbody>
</table>

**Table 7.5**
Lab № 8

Pointers and dynamic memory

**Goal:** to get practical skills of programming use of pointers and dynamic memory with one-dimensional arrays.

**Examples of programs**

**Example 8.1.** Enter a sequence of integers and create a dynamic array of numbers arranged after the first one-digit negative number (if there are no such number, select all). Use the function to swap elements: 1 and 2, 3 and 4, and so on.

The program code:

```cpp
#include <iostream>
using namespace std;
void fun(int a[], int n)
{
    if (n % 2) n--;
    for (int i = 0; i < n; i += 2)
    {
        int tmp = a[i];
        a[i] = a[i + 1];
        a[i + 1] = tmp;
    }
}
int main()
{
    int n = 0, i, j, kol, k = -1;
    cout << "Input n="; cin >> n;
    int * a = new int[n];
    cout << "Input " << n << " integers\n";
    for (i = 0; i < n; i++) cin >> a[i];
    for (i = 0; i < n; i++)
    if (a[i]>-10 && a[i]<0)
    {
        k = i; break;
    }
    kol = n - k - 1;
    int * b = new int[kol];
    for (j = 0, i = k + 1; i < n; i++, j++)
    b[j] = a[i];
    cout << "Dynamic array: \n";
    for (i = 0; i < kol; i++) cout << b[i] << " " ;
    fun(b, kol);
    cout << "\n Dynamic array with swapped elements: \n";
    for (i = 0; i < kol; i++) cout << b[i] << " " ;
    delete[]a;
    delete[]b;
    system("pause");
    return 0;
}
```
Lab 8. Pointers and dynamic memory

Results:
Input n=10
Input 10 integers
5
-13
7
3
2
10
-9
6
-10
8
Dynamic array:
6 -10 8
Dynamic array with swapped elements:
-10 6 8

Example 8.2. Enter some sequence of real numbers and create a dynamic array only from the numbers with values from the interval [60, 100]. Use the function to define the minimum and maximum elements and calculate the arithmetic mean of all elements.

The program code:

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

double fun(double a[], int n, double &min, double &max)
{
    double s = 0;
    min = max = a[0];
    for (int i = 0; i < n; i++)
    {
        s += a[i] / n;
        if (min > a[i]) min = a[i];
        if (max < a[i]) max = a[i];
    }
    return s;
}

int main()
{
    int n = 0, i, j, kol = 0;
    double min, max, avg;
    cout << "Input n="; cin >> n;
    double * a = new double[n];
    cout << "Input " << n << " real numbers\n";
    for (i = 0; i < n; i++)
        cin >> a[i];
    for (i = 0; i < n; i++)
        if (a[i] >= 60 && a[i] <= 100)
            kol++;
    double * b = new double[kol];
```

Flowchart of function `fun()`
for (j = 0, i = 0; i < n; i++)
    if (a[i] >= 60 && a[i] <= 100)
    {
        b[j] = a[i];
        j++;
    }
    cout << "Dynamic array: \n";
for (i = 0; i < kol; i++)
    cout << b[i] << " ";
    cout << endl;
    avg = fun(b, kol, min, max);
    cout << "Average= " << avg << " \nMIN= " << min << " MAX= " << max << endl;
    delete[]a;
    delete[]b;
    system("pause");
    return 0;
}

Results:
Input n=9
Input 9 real numbers
90
-123
5
3
216
88
4
56
99
Dynamic array:
90 88 99
Average=92.3333, min=88, max=99

Control questions
1) What is pointer? What is the syntax of its declaration?
2) What does NULL pointer mean?
3) How can we get an address of variable? What is & operand?
4) Explain the difference between variables a and b:
   a) int a; double b;  
   b) int *a; double *b;
5) Explain the difference between usual and dynamic arrays.
6) Which are the correct declarations of dynamic array of 5 integer numbers:
   a) int a[5];  
   b) int *a[5];  
   c) int *a=malloc(5);  
   d) int *a=(int*) malloc(20);  
   e) int *a=(int*) malloc(5*sizeof(int));
   f) int *a=new int [5];  
   g) int *a=new [5];  
   h) int *a=new int (5);
7) How can we free memory which was allocated with `new`?
8) Select correct commands to free memory from dynamic array `a` of 5 elements:
   a) `delete a[5]`;  d) `free (a)`;
   b) `delete a[]`;  e) `free a[5]`;
   c) `delete [a];
9) What is the difference between `malloc()` and `calloc()`?
10) What is the `realloc()` function used for?

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Individual problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input a sequence of integer numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers with value smaller than 6. Write a function to calculate an average of array <code>b</code></td>
</tr>
<tr>
<td>2</td>
<td>Input a sequence of real numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of positive numbers. Write a function to find the smallest element of array <code>b</code></td>
</tr>
<tr>
<td>3</td>
<td>Input a sequence of real numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers before the first negative number (or all numbers in case if there is no negative number). Write a function to sort array <code>b</code> on ascend</td>
</tr>
<tr>
<td>4</td>
<td>Input a sequence of integer numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of even non-zero numbers. Write a function to swap the biggest and the smallest elements of array <code>b</code></td>
</tr>
<tr>
<td>5</td>
<td>Input a sequence of numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers before the first number with zero value or all numbers (in case if there is no zero). Write a function to calculate the product of elements with absolute value smaller than 10</td>
</tr>
<tr>
<td>6</td>
<td>Input a sequence of real numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers with absolute value not bigger than 8. Write a function to calculate an average of the smallest and the biggest elements</td>
</tr>
<tr>
<td>7</td>
<td>Input a sequence of integer numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers before the first three-digit number or all numbers (in case if there is no three-digit number). Write a function to calculate an average of odd elements</td>
</tr>
<tr>
<td>8</td>
<td>Input a sequence of real numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers from <code>[-4, 7]</code>. Write a function to find the sum of array <code>b</code></td>
</tr>
<tr>
<td>9</td>
<td>Input a sequence of real numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers before the first number=100 (or all numbers in case if there is no negative number). Write a function to find maximum of array <code>b</code></td>
</tr>
<tr>
<td>10</td>
<td>Input a sequence of integer numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers which are multiple to 3 and 4. Write a function to find the sum of negative elements</td>
</tr>
<tr>
<td>11</td>
<td>Input a sequence of numbers (array <code>a</code>) and create dynamic array (array <code>b</code>) of numbers after the first two-digit number or all numbers (in case if there is no zero). Write a function to calculate the quantity of negative elements</td>
</tr>
<tr>
<td>№ var.</td>
<td>Individual problems</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td>12</td>
<td>Input a sequence of real numbers (array a) and create a dynamic array (array b) of numbers with values from the interval [10, 25]. Write a function to find the number of elements with values bigger than the value of the first element of array b</td>
</tr>
<tr>
<td>13</td>
<td>Input a sequence of numbers (array a) and create a dynamic array (array b) of odd numbers. Using the function change all negative elements to zeros</td>
</tr>
<tr>
<td>14</td>
<td>Input a sequence of real numbers (array a) and create a dynamic array (array b) of numbers with absolute values outside the interval (20, 40]. Using the function, calculate the number of elements less than the average of all elements</td>
</tr>
<tr>
<td>15</td>
<td>Input a sequence of numbers (array a) and create a dynamic array (array b) of nonzero numbers. Use the function to find the largest of the even elements</td>
</tr>
<tr>
<td>16</td>
<td>Input a sequence of real numbers (array a) and create a dynamic array (array b) of numbers with absolute values from the interval [5, 50]. Use a function to find the minimal positive element</td>
</tr>
<tr>
<td>17</td>
<td>Input a sequence of numbers (array a) and create a dynamic array (array b) of numbers whose values do not exceed 100. Using a function, calculate the sum of two-digit elements</td>
</tr>
<tr>
<td>18</td>
<td>Input a sequence of numbers (array a) and create a dynamic array (array b) of the numbers after the first three-digit number. Using the function, calculate the quantity of elements that are multiples of 5</td>
</tr>
</tbody>
</table>
Lab № 9

Characters (char) and C-strings (char*)

Goal: to get practical skills of writing programs with characters and c-strings.

Examples of programs

**Example 9.1.** Enter a string and calculate a quantity if * in it.

```c
#include <iostream>
#include <string>

using namespace std;

int main()
{
    char s[50];
    int i, n, kol = 0;
    puts("Enter a string: ");
    gets_s(s);
    n = strlen(s);
    for (i = 0; i < n; i++)
    {
        if (s[i] == '*') kol++;
    }
    printf("Quantity of symbols * in the string = %d\n", kol);
    system("pause");
    return 0;
}
```

Result:
Enter a string:
q***H****f**
Quantity of symbols * in the string = 9

**Example 9.2.** Enter two strings and define whether they are equal or different.

Flowchart of function **check()**

Flowchart of the **main** function
#include <iostream>
using namespace std;

int check(char* s1, char* s2)
{
    int i, j, n1, n2;
    n1 = strlen(s1); n2 = strlen(s2);
    if (n1 != n2) return 0;
    for (i = 0; i < n1; i++)
        if (s1[i] != s2[i]) return 0;
    return 1;
}

int main()
{
    char s1[50], s2[50];
    puts("Enter the first string");
    gets_s(s1);
    puts("Enter the second string");
    gets_s(s2);
    int res = check(s1, s2);
    if (res)
        cout << "The strings are equal" << endl;
    else
        cout << "The strings are different" << endl;
    system("pause");
    return 0;
}

Results:
Enter the first string
Hello world
Enter the second string
Hello world!
The strings are different

Example 8.3. Enter a string and define whether it contains punctuation marks.

#include <iostream>
using namespace std;

bool punct(char* s)
{
    int i;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i] == '.' || s[i] == ',' || s[i] == ';' || s[i] == ':' || s[i] == '?' || s[i] == '-')
            return 1;
    return 0;
}

int main()
{
    char s[100];
    puts("Enter a string");
    gets_s(s);
    int res = punct(s);
    if (res)
        cout << "The string contains punctuation marks" << endl;
    else
        cout << "Punctuation marks are absent" << endl;
    system("pause");
Lab 9. Characters (char) and C-strings (char*)

```cpp
return 0;
```

Results:
Enter a string
For example, the length of a string can be found with the length() function:
The string contains punctuation marks

Enter a string
You can access the characters in a string by referring to its index number inside square brackets
Punctuation marks are absent

Example 8.4. Enter a string. Create a new string with the vowels.

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

int kol_vo(char *s)
{
    int k = 0, i;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
            k++;
    return k;
}

void new_string(char * s, char *s1)
{
    int i, j = 0;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
            {s1[j] = s[i]; j++;}
    s1[j] = '\0';
}
```

Flowchart of function
kol_vo()

Flowchart of function
new_string()

Flowchart of the
main function

#include <iostream>
using namespace std;
int kol_vo(char *s)  
{  
    int k = 0, i;
    for (i = 0; s[i] != '\0'; i++)  
        if (s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')  
            k++;
    return k;
}   

void new_string(char * s, char *s1)  
{  
    int i, j = 0;
    for (i = 0; s[i] != '\0'; i++)  
        if (s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')  
            {s1[j] = s[i]; j++;}
    s1[j] = '\0';
}
int main()
{
    int i, kol = 0; char * s = new char[100];
    puts(" Enter a string: "); gets_s(s, 100);
    kol = kol_vo(s);
    char * s1 = new char[kol + 1];
    new_string(s, s1);
    puts("\n New string from vowels "); puts(s1);
    delete[] s; delete[] s1;
    system("pause");
    return 0;
}

Results:
Enter a string:
You can access the characters in a string by referring to its index number inside square brackets

New string from vowels
Ouaaeaelaeioiieuilieuaeae

Example 8.5. Enter a string and output all words in a column.

#include <iostream>
using namespace std;
int main()
{
    char s[100], *t;
    puts(" Enter a string:"); gets_s(s);
    cout << "\n Words: 
";
    t = strtok(s, " .,;?!
-");
    while (t != NULL)
    {
        puts(t);
        t = strtok(NULL, " .,;?!
-");
    }
    system("pause");
    return 0;
}

Results:
Enter a string:
You can access the characters in a string by referring to its index number inside square brackets

Words:
You
can
access
the
characters
in
a
string
by
referring
to
its
index
number
inside
square
brackets
Example 8.8. Enter a string. Delete all short words (with length < 4) in this string.

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

int main()
{
    char * s = new char[100], *t;
    char * s1 = new char[100];
    strcpy(s1, "\0");
    puts(" Input a string:"); gets_s(s, 100);
    t = strtok(s, " .,;?!- ");
    while (t != NULL)
    {
        if (strlen(t) >= 4)
        {
            strcat(s1, t);
            strcat(s1, " \0");
        }
        t = strtok(NULL, " .,;?!- ");
    }
    puts("\n String without short words");
    strcpy(s, s1);
    puts(s);
    delete[] s; delete[] s1;
    system(" pause");
    return 0;
}
```

Results:
Input a string:
You can access the characters in a string by referring to its index number inside square brackets

String without short words
access characters string referring index number inside square brackets

Control questions
1) What is ANSI-table? How many characters are in this table?
2) What is the size of char variable?
3) What operations with symbols do you know?
4) Which expressions are true?
   a) 'θ' < 'y'
   б) '5' > 'f'
   в) 'F' > 'f'
   г) 'w' > 'W'
   д) ' ' > 'θ'
   е) 'Я' > 'ю'
5) How can we change the case of Latin and Cyrillic symbols in programs?
6) Write all ways of declaration of c-string.
7) Is it possible to assign a part of c-string to another c-string? Write an example.
8) Which function allows to add one string to another? Write an example.
9) Write all ways of input of c-strings in console mode.
Individual task

1. Create program and flowchart for problems from table 9.1. Do not use functions from string.h, except of strlen.
2. Create program and flowchart for problems from table 9.2. Use functions from string.h.

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Individual problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input a string and define the number of low case letters in this string</td>
</tr>
<tr>
<td>2</td>
<td>Input a string and change the capital letters to spaces</td>
</tr>
<tr>
<td>3</td>
<td>Input a string and define the number of digits in this string</td>
</tr>
<tr>
<td>4</td>
<td>Input a string and change the Latin letters to exclamation mark</td>
</tr>
<tr>
<td>5</td>
<td>Input a string and change punctuation marks to symbol ‘#’</td>
</tr>
<tr>
<td>6</td>
<td>Input a string and define the number of vowels in this string</td>
</tr>
<tr>
<td>7</td>
<td>Input a string and output ANSI-codes instead of characters of this string</td>
</tr>
<tr>
<td>8</td>
<td>Input a string and define the number of punctuation marks in this string</td>
</tr>
<tr>
<td>9</td>
<td>Input a string and change the capital letters to spaces</td>
</tr>
<tr>
<td>10</td>
<td>Input a string and define the number of commas in this string</td>
</tr>
<tr>
<td>11</td>
<td>Input a string and change the Latin letters to ‘*’</td>
</tr>
<tr>
<td>12</td>
<td>Input a string and change punctuation marks to symbol ‘-’</td>
</tr>
<tr>
<td>13</td>
<td>Input a string and define the number of capital letters in this string</td>
</tr>
<tr>
<td>14</td>
<td>Input a string and output ANSI-codes of digits in this string</td>
</tr>
<tr>
<td>15</td>
<td>Input a string and calculate the quantity of hyphen symbol in the string</td>
</tr>
<tr>
<td>16</td>
<td>Input a string and create a new string of Latin letters</td>
</tr>
<tr>
<td>17</td>
<td>Input a string and calculate the quantity of “th” combination</td>
</tr>
<tr>
<td>18</td>
<td>Input a string and output all vowels</td>
</tr>
</tbody>
</table>

Table 9.1
Table 9.2

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input a string and delete words with digits from the string</td>
</tr>
<tr>
<td>2</td>
<td>Input a string and output the longest word of the string in the reverse order</td>
</tr>
<tr>
<td>3</td>
<td>Input a string and create a new string of words with length bigger than 6</td>
</tr>
<tr>
<td>4</td>
<td>Input a string and output all words, which start and end with the same symbol</td>
</tr>
<tr>
<td>5</td>
<td>Input a string and delete words with odd length</td>
</tr>
<tr>
<td>6</td>
<td>Input a string and output the shortest word of the string</td>
</tr>
<tr>
<td>7</td>
<td>Input a string and a word. Delete this word from the string</td>
</tr>
<tr>
<td>8</td>
<td>Input a string and output all words, which length is bigger than 7</td>
</tr>
<tr>
<td>9</td>
<td>Input a string and output all words, which start from letter K</td>
</tr>
<tr>
<td>10</td>
<td>Input a string and find the longest sequence of letters A and the length of this sequence</td>
</tr>
<tr>
<td>11</td>
<td>Input a string with parenthesis and output all symbols inside the parenthesis</td>
</tr>
<tr>
<td>12</td>
<td>Input a string and define, how many combinations &quot;Microsoft&quot; are in the string</td>
</tr>
<tr>
<td>13</td>
<td>Input a string and count the quantity of words, which start and end with the same letter (ignoring the case)</td>
</tr>
<tr>
<td>14</td>
<td>Input a string and define whether the combination &quot;C++&quot; presents in the string, and if yes, output the position of the first combination</td>
</tr>
<tr>
<td>15</td>
<td>Input a string and insert length of word after each word. Use function <code>itoa</code> to convert a number to C-string</td>
</tr>
<tr>
<td>16</td>
<td>Input a string and change symbols of arithmetic operations to their names (+ “plus” etc.)</td>
</tr>
<tr>
<td>17</td>
<td>Input a string and create a new string of words with hyphen</td>
</tr>
<tr>
<td>18</td>
<td>Input a string and insert symbol * before and after each word</td>
</tr>
</tbody>
</table>
Lab 10
Text files

**Goal:** to get practical skills of creating and editing of text files in Visual C++.

**Examples of programs**

**Example 1.** Create a file with strings (the last string is an empty string). Output the file content. Output strings longer than 20 characters. Calculate average length of the strings.

```cpp
Function Description view_file()

Function view_file() displays the contents of the file.
Function parameter – string variable name – the name of the physical file on the disk.

The function is type of void, that is, it does not return a value of main().
Local variables:
– s – the string up to 100 characters in length for reading lines from a file;
– f – the file variable with which has got the access to the file.

The algorithm of the function:
1. Open file:
   ```cpp
   f = fopen(name, "rt");
   ```
   The fopen command opens the file. The first parameter (name) is the name of the file that is being opened, the second parameter ("rt") is the opening mode: the file is opened as a text file for reading data.
2. Make sure the file is open. If not, display a message and interrupt the function:
   ```cpp
   if (f == NULL) { cout << "Cannot open file to view\n"; return; }
   ```
3. Read lines from a file until they are finished:
while (fgets(s, 100, f) > 0)

The next line is read from the file using the command:
fgets(s, 100, f)

It has three arguments:
- `s` is a string variable that writes a string read from a file;
- `100` – maximum number of characters that can be read at a time;
- `f` – the file from which the data is read.

The function `fgets()` returns the actual length (number of characters) of the read string. That is, the loop will exit when `fgets()` returns a value of 0, which means that there are no more lines in the file.

4. To delete \n from the end of the string:
   s[strlen(s) - 1] = '\0';

5. Display each scanned line `s`.
   puts(s);

6. At the end of the cycle, close the file:
   fclose(f);

The code:
#include <iostream>
using namespace std;

// Create file
void create_file(char* name)
{
    char s[100]; FILE* f;
    f = fopen(name, "wt"); // to open the file as text for creation
    // to check if file is opened
    if (f == NULL) { cout << "Cannot create file\n"; return; }
    cout << "Input strings" << endl;
    do {
        gets_s(s, 100); // to enter a string from the keyboard
        fputs(s, f);    // to write the string to file
        fputs("\n", f);  // to move a cursor to the beginning of the next line
    } while (strcmp(s, ""));
    fclose(f);      // to close (and save) the file
}

// View file
void view_file(char* name)
{
    char s[100]; FILE* f;
    f = fopen(name, "rt"); // to open file as text for reading
    if (f == NULL) { cout << "Cannot open file to view\n"; return; }
    cout << "\nView file" << endl;
    while (fgets(s, 100, f))
    { // to read strings from file while not reached the end of file
        s[strlen(s) - 1] = '\0'; // to delete \n from the end of the string
        puts(s);                // output a string
    }
    fclose(f);
}
// Output strings longer than 20 characters
void strings20(char* name)
{
    char s[100]; FILE* f;
f = fopen(name, "rt");
    if (f == NULL)
        { cout << "Cannot open file\n"; return; }
cout << "\nStrings with length bigger than 20 characters:" << endl;
while (fgets(s, 100, f)>0)
{
    s[strlen(s) - 1] = '\0';
    if (strlen(s) > 20) puts(s);
}
fclose(f);
}

// Calculate average of the strings
double avg_str(char* name)
{
    char s[100]; FILE* f;
    int sum = 0, kol = 0; // kol is for total sum of strings' length
    f = fopen(name, "rt");
    if (f == NULL)
        { cout << "Cannot open file\n"; return 0; }
while (fgets(s, 100, f))
{
    sum += strlen(s)-1; // -1 is to delete \n from the end of the string
    kol++;
}
fclose(f);
    if (kol - 1)
        return (double)sum / (kol - 1); // average
    else
        return 0;
}

// Main function
int main()
{
    FILE * f = NULL; //declaration of the file variable
    char name [] = "myfile.txt"; //name of the file on the HDD
    create_file(name);
    view_file(name);
    strings20(name);
    double sr = avg_str(name);
cout << "\nAverage strings length=" << sr << endl;
system("pause");
    return 0;
}
Results:

C-strings
Therefore, this array has a capacity to store sequences of up to 20 characters. But this capacity does not need to be fully exhausted:
the array can also accommodate shorter sequences.
For example,
at some point in a program, either the sequence "Hello"
or the sequence "Merry Christmas" can be stored in foo,
since both would fit in a sequence with a capacity for 20 characters.

View file
C-strings
Therefore, this array has a capacity to store sequences of up to 20 characters. But this capacity does not need to be fully exhausted:
the array can also accommodate shorter sequences.
For example,
at some point in a program, either the sequence "Hello"
or the sequence "Merry Christmas" can be stored in foo,
since both would fit in a sequence with a capacity for 20 characters.

Strings with length bigger than 20 characters:
Therefore, this array has a capacity to store sequences of up to 20 characters. But this capacity does not need to be fully exhausted:
the array can also accommodate shorter sequences.
For example,
at some point in a program, either the sequence "Hello"
or the sequence "Merry Christmas" can be stored in foo,
since both would fit in a sequence with a capacity for 20 characters.

Average strings length=48.5

Control questions

1) What file is called text file?
2) What is the difference between text and binary files?
3) Write methods of reading and writing data from/to text file.
4) What is the difference between methods Write and WriteLine?
5) Write command to create a text file.
6) Write command to remove a text file.
**Individual task**

Create files, which name is your surname and .rtf extension.

<table>
<thead>
<tr>
<th>№ var.</th>
<th><strong>Individual problem</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Calculate a quantity of strings with length bigger than 20</td>
</tr>
<tr>
<td>2</td>
<td>Calculate a quantity of strings, which end with punctuation marks</td>
</tr>
<tr>
<td>3</td>
<td>Output all strings, which do not contain parenthesis</td>
</tr>
<tr>
<td>4</td>
<td>Output all words with length bigger than 5, and calculate their quantity</td>
</tr>
<tr>
<td>5</td>
<td>Calculate the quantity of strings, which begin and end with the same letter</td>
</tr>
<tr>
<td>6</td>
<td>Output the longest word of the file and the length of this word</td>
</tr>
<tr>
<td>7</td>
<td>Output all strings, which do not contain punctuation marks</td>
</tr>
<tr>
<td>8</td>
<td>Output the shortest word of the file and the order number of its row</td>
</tr>
<tr>
<td>9</td>
<td>Calculate a quantity of strings, which contain digits</td>
</tr>
<tr>
<td>10</td>
<td>Output all strings, which start from the low-case letter</td>
</tr>
<tr>
<td>11</td>
<td>Calculate the quantity of punctuation marks in the file</td>
</tr>
<tr>
<td>12</td>
<td>Output all strings, which do not contain capital letters</td>
</tr>
<tr>
<td>13</td>
<td>Output all words, which contain letter ‘z’, and calculate their quantity</td>
</tr>
<tr>
<td>14</td>
<td>Calculate the quantity of strings which contain punctuation marks</td>
</tr>
<tr>
<td>15</td>
<td>Output the shortest words from each string of the file</td>
</tr>
<tr>
<td>16</td>
<td>Output all strings which do not contain digits</td>
</tr>
<tr>
<td>17</td>
<td>Calculate the quantity of strings, which start from the capital letter</td>
</tr>
<tr>
<td>18</td>
<td>Output all words with length smaller than 6 and calculate their quantity</td>
</tr>
</tbody>
</table>
Lab 11

Structures

Goal: to get practical skills of operation with structures and their fields.

Examples of programs

Example 11.1. Create a program for processing data with student session results: student name, group, and the results of the two exams. The program provides the possibility of inputting and displaying data on some number of students and the selection of data on students who have successfully passed the session and got the scholarship (have no unsatisfactory marks and an average score is more than 75 points), and also determine the name of the student with the highest average score. Use array of structures to store data.

The program code:

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

struct student
{
    char surname[22], gr[8];
    int ex1, ex2;
};

int main()
{
    int kol = 0;  // Total number of students
    cout << "Enter a number of students - "; cin >> kol;
    student *z = new student[kol];  // Array of students
    cout << "Enter rows with information about " << kol 
         << " students and their marks : 
Surname Group Mark1 Mark2" << endl;
    for (int i = 0; i < kol; i++)
        scanf("%s %s %i %i", z[i].surname, z[i].gr, &z[i].ex1, &z[i].ex2);
    cout << "Marks of " << kol << " students: 
Surname Group Mark1 Mark2";
    for (int i = 0; i < kol; i++)
        printf("%s %s %i %i\n", z[i].surname, z[i].gr, z[i].ex1, z[i].ex2);
    double sr, max(0);  // Average score and maximal mark
    char maxname[22];  // Surname of the student with maximal mark
    int n = 0;  // Quantity of students
    for (int i = 0; i < kol; i++)
    {
        if (z[i].ex1 >= 60 && z[i].ex2 >= 60 && (z[i].ex1 + z[i].ex2) / 2 >= 75)
            n++;
        printf("%s %s %i %i\n", z[i].surname, z[i].gr, z[i].ex1, z[i].ex2);
    }
    sr = (z[i].ex1 + z[i].ex2) / 2.0;
    if (sr > max)
    {
        max = sr;
        strcpy(maxname, z[i].surname);
    }
    cout << "Successfully passed the session and got the scholarship: 
Surname Group Mark1 Mark2";
    return 0;
}
```
cout << endl << "Quantity of students: " << n << endl;
cout << maxname << " has maximal average mark " << max << endl;
system("pause");
return 0;
}

Results:
Enter a number of students - 4
Enter rows with information about 4 students and their marks:
Surname Group Mark1 Mark2
Shevchenko IPZ-1.1 65 90
Ivanov IK-1.1 50 60
Chumak IPZ-1.2 75 85
Grib IPZ-1.1 20 65

Marks of 4 students:
Surname Group Mark1 Mark2
Shevchenko PZ-1.1 65 90
Ivanov IK-1.1 50 60
Chumak IPZ-1.2 75 85
Grib IPZ-1.1 20 65

Successfully passed the session and got the scholarship:
Shevchenko IPZ-1.1 65 90
Chumak IPZ-1.2 75 85

Quantity of students: 2
Chumak has maximal average mark 80

Example 11.2. Write information about 3 books to the file. View this file. Find the newest book.

The code:
#include <iostream>
using namespace std;
int main()
{
struct book
{
    char author[51];
    char bname[51];
    int year;
    float price;
};
FILE * f;
book z, maxyear; //z is a book, maxyear is a book with maximal year
//File creation
f = fopen("myfile.txt", "wt"); // To open file for creation
if (f == NULL) // Test if file is opened successfully
    { cout << "Can't open the file\n"; return 0; }
puts("Input information about 3 books");
printf("Author\tBook\tYear\tPrice\n");
for (int n = 0; n < 3; n++)
{
    // To read an information about a book from the keyboard
    scanf("%s %s %d %f", z.author, z.bname, &z.year, &z.price);
// To write inputted information to file
fprintf(f, "%s|%s|%d|%.2f\n", z.author, z.bname, z.year, z.price);
}
fclose(f);

// View file
puts("\nView file");
f = fopen("myfile.txt", "rt"); // To open file for reading
if (f == NULL)
    { cout << "Can't open the file\n"; return 0; }
while (!feof(f)) // While the end of file is not reached
{
    // To read an information about a book from the file to variable z
    fscanf(f, "%s|%s|%d|%.2f\n", z.author, z.bname, &z.year, &z.price);
    // To output the z variable to the screen
    printf("%s|%s|%d|%.2f\n", z.author, z.bname, z.year, z.price);
}
fclose(f);

// Search for a book with maximal year
maxyear.year = 0; // Initial value for maximal year
f = fopen("myfile.txt", "rt");
if (f == NULL)
    { cout << "Can't open the file\n"; return 0; }
while (!feof(f))
{
    fscanf(f, "%s|%s|%d|%.2f\n", z.author, z.bname, &z.year, &z.price);
    // If a year of the current book is bigger than maximal year
    if (z.year > maxyear.year)
        maxyear = z; // Assign the current book to maxbook
}
fclose(f);
if (maxyear.year > 0) // If maximal year is found
    { // Output the information about it
        puts("\nThe newest book");
        printf("%s|%s|%d|%.2f\n", maxyear.author, maxyear.bname, maxyear.year, maxyear.price);
    }
system("pause");
return 0;

Results:
Input information about 3 books
Author Book Year Price
Shevchenko Kobzar 2010 150
Rowling Harry_Potter 2016 340
Straustrup The_C++_Programming_Language 2012 275

View file
Shevchenko | Kobzar | 2010 | 150.00
Rowling | Harry_Potter | 2016 | 340.00
Straustrup | The_C++_Programming_Language | 2012 | 275.00

The newest book
Rowling | Harry_Potter | 2016 | 340.00
Control questions

1) Give a definition of the structure as a data type.
2) How can we access to structure fields?
3) How can we define the size of memory, which is necessary to store the structure?
4) Is it correct to give the same names for variables and structure fields?
5) Write declaration of the structure, which describes for some electric device the following characteristics: device name, power consumption and rated voltage.

Individual task

1. Enter array of five structures and solve individual problem (Table 11.1).
2. Create text file with information about several structures and solve individual problem (Table 11.2).

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Structure fields</th>
<th>Individual problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Information about students’ exams: – surname, – group, – physics – informatics – history</td>
<td>Define the average mark of each student and select students with average marks bigger than 75</td>
</tr>
<tr>
<td>2</td>
<td>– surname, – group, – physics – informatics – history</td>
<td>Select students with at least one unsatisfactory mark and define their quantity</td>
</tr>
<tr>
<td>3</td>
<td>– surname, – group, – physics – informatics – history</td>
<td>Output all students who passed the session and their average marks. Sort them in descending order</td>
</tr>
<tr>
<td>4</td>
<td>Information about employees: – surname, – position, – education – birth year, – salary</td>
<td>Select employees younger than 30 years and calculate their quantity</td>
</tr>
<tr>
<td>5</td>
<td>– surname, – position, – education – birth year, – salary</td>
<td>Select employees, without high education and calculate their percentage in all employees</td>
</tr>
<tr>
<td>6</td>
<td>– surname, – position, – education – birth year, – salary</td>
<td>Define all the oldest and the youngest employees</td>
</tr>
<tr>
<td>7</td>
<td>– surname, – position, – education – birth year, – salary</td>
<td>Calculate the average salary and select all employees with salary higher than average</td>
</tr>
<tr>
<td>8</td>
<td>Information about products: – name, – manufacturer, – price, – quantity</td>
<td>Define the most expensive product, output full information about it and calculate its total cost</td>
</tr>
<tr>
<td>9</td>
<td>– name, – manufacturer, – price, – quantity</td>
<td>Calculate the total quantity and the total cost of all goods</td>
</tr>
<tr>
<td>10</td>
<td>– name, – manufacturer, – price, – quantity</td>
<td>Calculate the average price and select all goods with price lower than average</td>
</tr>
<tr>
<td>11</td>
<td>– name, – manufacturer, – price, – quantity</td>
<td>Input a manufacturer and select all goods of this manufacturer and their quantity and average price</td>
</tr>
<tr>
<td>№ var.</td>
<td>Structure fields</td>
<td>Individual problem</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>12</td>
<td><em>Information about TV-programs:</em> – name, – frequency (times a week), – rating</td>
<td>Select all programs with frequency 1 time a week and define the most popular of them</td>
</tr>
<tr>
<td>13</td>
<td>– name, – frequency (times a week), – rating</td>
<td>Select all programs with frequency more than 3 times a week and sort them by rating</td>
</tr>
<tr>
<td>14</td>
<td>– rating</td>
<td>Calculate the average rating and select all programs with rating higher than average</td>
</tr>
<tr>
<td>15</td>
<td><em>Information about books in a library:</em> – author, – name, – publishing year, – number of pages</td>
<td>Define the most popular TV-program and sort programs by name AZ</td>
</tr>
<tr>
<td>16</td>
<td><em>Information about books in a library:</em> – author, – name, – publishing year, – number of pages</td>
<td>Define the oldest book and output all books of the given author</td>
</tr>
<tr>
<td>17</td>
<td>– publishing year, – number of pages</td>
<td>Calculate an average number of pages and find the most new book</td>
</tr>
<tr>
<td>18</td>
<td>– number of pages</td>
<td>Calculate a quantity of books, which were published after 2000. Sort books by number of pages on descending order.</td>
</tr>
</tbody>
</table>

### Table 11.2

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Content of the file</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List for automobile inspection: information about the stolen cars: state number, brand of car, color, date of application (3 fields: day, month, year)</td>
<td>Output information about stolen cars “BMW”, which were stolen in the current year</td>
</tr>
<tr>
<td>2</td>
<td>Information about the payment for building services: street, house number, surname of habitant, date of payment (3 fields: day, month, year), debt</td>
<td>Output information about debtors from Pushkinskaya street, with a debt over 500 ₴</td>
</tr>
<tr>
<td>3</td>
<td>Summer curriculum of trains: number of train, place of departure, place of destination, departure time (3 fields: hours, minutes, seconds)</td>
<td>Output information about the trains of direction Odessa - Kyiv, that leave from 10:00 to 17:00</td>
</tr>
<tr>
<td>4</td>
<td>Log of events of the operating system: the name of a start program; level of event (error, warning and others like that); date of event (3 fields: day, month, year); time of event (3 fields: hours, minutes, seconds)</td>
<td>Output information about errors and define how many days passed from each of errors to current moment of time</td>
</tr>
<tr>
<td>5</td>
<td>Information about medications in a pharmacy: the name of medications, expiry of their term (3 fields: day, month, year), price</td>
<td>Output data about medications, which expiry term is in a current year</td>
</tr>
<tr>
<td>6</td>
<td>List of employees of an enterprise: table number, surname, sex (m/f), birth year, position</td>
<td>Output information about all employees-women older than 55 years and men older 60 years</td>
</tr>
<tr>
<td>№ var.</td>
<td>Content of the file</td>
<td>Problem</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------</td>
<td>---------</td>
</tr>
<tr>
<td>7</td>
<td>Information about products: name, date of producing (3 fields: day, month, year), term of realization, price</td>
<td>Output information about products which were produced today</td>
</tr>
<tr>
<td>8</td>
<td>The list of employees: the log number, surname, position, year of birth, year of recruitment</td>
<td>Output information about employees who have worked for more than ten years at the enterprise</td>
</tr>
<tr>
<td>9</td>
<td>List of holidays in the calendar: holiday name, date (3 fields: day, month, year)</td>
<td>Display information about winter holidays</td>
</tr>
<tr>
<td>10</td>
<td>Schedule of airplanes: flight number, destination, departure time (3 fields: hours, minutes, seconds)</td>
<td>Output information about all flights before 10 am.</td>
</tr>
<tr>
<td>11</td>
<td>Student group list: surname, name, year of birth, average mark</td>
<td>Find a student with the highest average mark</td>
</tr>
<tr>
<td>12</td>
<td>Repertoire of the opera house: name of the play, genre, date (3 fields: day, month, year), beginning</td>
<td>Display the information about the ballets, which have not yet taken place, as well as children's plays (beginning up to 15 hours)</td>
</tr>
<tr>
<td>13</td>
<td>Student group list: surname, name, date of birth (3 fields: day, month, year), average mark</td>
<td>Output information about students who have birthday in the current month</td>
</tr>
<tr>
<td>14</td>
<td>The list of employees: the log number, surname, position, year of birth, year of recruitment</td>
<td>Output information about engineers</td>
</tr>
<tr>
<td>15</td>
<td>Information about products: name, date of producing (3 fields: day, month, year), term of realization, price</td>
<td>Output information about products with the term of realization today</td>
</tr>
<tr>
<td>16</td>
<td>Summer curriculum of trains: number of train, place of departure, place of destination, departure time (3 fields: hours, minutes, seconds)</td>
<td>Output information about the trains, which are departing now</td>
</tr>
<tr>
<td>17</td>
<td>Information about products: name, date of producing (3 fields: day, month, year), term of realization, price</td>
<td>Output information about products with word “chocolate” in the name</td>
</tr>
<tr>
<td>18</td>
<td>Information about products: name, date of producing (3 fields: day, month, year), term of realization, price</td>
<td>Output information about products which were produced last year</td>
</tr>
</tbody>
</table>
Lab 12

Binary files

**Goal:** to get practical skills of creation and edition of binary files in Visual C++.

**Examples of programs**

**Example 12.1.** To create the binary file with at least 10 records. A structure consists of the fields: *Surname, Name, Patronymic, Position, Year of employment, Salary*. View the file. Select data about engineers who worked more than 5 years. Create a text file with information about employees with salary smaller than average.

The program code:

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

struct employee
{
    char surname[30];
    char name[30];
    char patronymic[30];
    char position[30];
    int start_year;
    float salary;
};

//To add one record to a binary file
void add_record(char* name)
{
    employee z;
    FILE* f;
    f = fopen(name, "ab");
    if (f == NULL)
    {
        cout << "Cannot create file to add record\n"; return; }
    scanf("%s\t%s\t%s\t%s\t%i\t%f", z.surname, z.name, z.patronymic, z.position, &z.start_year, &z.salary);
    fwrite(&z, sizeof(employee), 1, f);
    fclose(f);
}

//View the binary file
void view_file(char* name)
{
    employee z;
    FILE* f;
    f = fopen(name, "rb");
    if (f == NULL) { cout << "Cannot open file to view\n"; return; }
    cout << "View binary file" << endl;
    cout << "Surname\tName\tPatronymic\tPosition\tStart date\tSalary\n" << endl;
    while (fread(&z, sizeof(employee), 1, f))
    {
        printf("%s\t%s\t%s\t%s\t%d\t%.2f\n", z.surname, z.name, z.patronymic, z.position, z.start_year, z.salary);
    }
    fclose(f);
}
```
// To select data about the engineers who worked for more than 5 years.
void select_data(char* name)
{
    employee z;
    FILE* f;
    int curr_year = 2020;
    f=fopen(name, "rb");
    if(f==NULL)
    {
        cout<<"Cannot open file to view\n"; return;
    }
    cout<<"\nInformation about engineers who worked for more than 5 years"<<endl;
    cout<<"Surname\tName\tPatronymic\tPosition\tStart date\tSalary\n"<<endl;
    while (fread(&z, sizeof(employee), 1, f))
    {
        if (strcmp(z.position, "engineer")==0 && curr_year - z.start_year > 5)
            printf("%s\t%s\t%s\t%s\t%i\t%6.2f\n", z.surname, z.name, z.patronymic,
                    z.position, z.start_year, z.salary);
    }
    fclose(f);
}

// Create the text file with information about employees with salary smaller than average
void create_text_file(char* bname, char* tname)
{
    employee z;
    FILE* fb, *ft;
    float sum = 0, avg;
    int k = 0;
    fb = fopen(bname, "rb");
    if (fb == NULL)
    {
        cout << "Cannot open file\n";
        return;
    }
    while (fread(&z, sizeof(employee), 1, fb))
    {
        sum += z.salary;
        k++;
    }
    if (k) avg = sum / k;
    else avg = 0;
    cout << "\nAverage salary = " << avg << endl;
    fseek(fb, 0, 0);//To return to the beginning of the file
    ft = fopen(tname, "wt"); //To open ft file for creation
    if (ft == NULL)
    {
        cout << "Cannot create text file\n"; return;
    }
    // To read data from the file fb to variable z while not end of file
    while (fread(&z, sizeof(employee), 1, fb))
    {
        // If a salary of the current employee is less than average,
        if (z.salary < avg)
            fprintf(ft, "%s\t%s\t%s\t%i\t%g\n", z.surname, z.name, z.patronymic,
                        z.position, z.start_year, z.salary);
    }
    fclose(ft);
    fclose(fb);
//View text file
void view_text_file(char* name)
{
    employee z;
    FILE* f;
    f = fopen(name, "rt");
    if (f == NULL) { cout << "Cannot open text file to veiw\n"; return; }
    cout << "View text file with information about employees with salary smaller than average" << endl;
    cout << "Surname\tName\tPatronymic\tPosition\tStart date\tSalary\n" << endl;
    while (!feof(f))
    {
        fscanf(f, "%s\t%s\t%s\t%i\t%f\n", z.surname, z.name, z.patronymic, z.position, &z.start_year, &z.salary);
        printf("%s\t%s\t%s\t%i\t%6.2f\n", z.surname, z.name, z.patronymic, z.position, z.start_year, z.salary);
    }
    fclose(f);
}

//Main file
int main()
{
    char name[] = "binaryfile.txt", tname[]="textfile.txt";
    cout<<"Input information about employees:"<<endl;
    cout<<"Surname\tName\tPatronymic\tPosition\tStart date\tSalary\n"<<endl;
    for(int i=0; i<6; i++) add_record(name);
    view_file(name);
    select_data(name);
    create_text_file(name, tname);
    view_text_file(tname);
    system("pause");
    return 0;
}

Test data:
Shevchuk Petro Olegovich director 2008 25000
Vasilyev Serhii Vladimirovich bookkeeper 2015 20000
Tarasenko Oksana Ivanivna engineer 2013 12000
Semenov Ivan Petrovich guard 2017 7000
Grib Semen Dmitriyovich engineer 2018 8500
Petrenko Ivan Mykolayovich engineer 2012 11000

Results:
View binary file
Surname  Name  Patronymic  Position  Start date  Salary
Shevchuk  Petro  Olegovich  director  2008  25000.00
Vasilyev  Serhii  Vladimirovich  bookkeeper  2015  20000.00
Tarasenko  Oksana  Ivanivna  engineer  2013  12000.00
Semenov  Ivan  Petrovich  guard  2017  7000.00
Grib  Semen  Dmitriyovich  engineer  2018  8500.00
Petrenko  Ivan  Mykolayovich  engineer  2012  11000.00

Information about engineers who worked for more than 5 years
Surname  Name  Patronymic  Position  Start date  Salary
Tarasenko  Oksana  Ivanivna  engineer  2013  12000.00
Petrenko  Ivan  Mykolayovich  engineer  2012  11000.00

Average salary = 13916.7
View text file with information about employees with salary smaller than average

<table>
<thead>
<tr>
<th>Surname</th>
<th>Name</th>
<th>Patronymic</th>
<th>Position</th>
<th>Start date</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tarasenko</td>
<td>Oksana</td>
<td>Ivanivna</td>
<td>engineer</td>
<td>2013</td>
<td>12000.00</td>
</tr>
<tr>
<td>Semenov</td>
<td>Ivan</td>
<td>Petrovich</td>
<td>guard</td>
<td>2017</td>
<td>7000.00</td>
</tr>
<tr>
<td>Grib</td>
<td>Semen</td>
<td>Dmitriyovich</td>
<td>engineer</td>
<td>2018</td>
<td>8500.00</td>
</tr>
<tr>
<td>Petrenko</td>
<td>Ivan</td>
<td>Mykolayovich</td>
<td>engineer</td>
<td>2012</td>
<td>11000.00</td>
</tr>
</tbody>
</table>

Screenshot of the binary file:

**Control questions**

1) What file is called binary?
2) What is the difference between opening modes `OpenOrCreate` and `Open`?
3) Write instruction to create binary file.
4) Write instruction to move 1) to the beginning of binary file; 2) to the end of binary file. In which situations we use these commands?

**Individual task**

Create binary file, view it and select (output) records according to your individual variant (table 12.1). Write selected records to the text file.

<table>
<thead>
<tr>
<th>№ var.</th>
<th>Content of the binary file</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List of students in a group: log number, last and first names, marks in physics, math and</td>
<td>Select students, who has no “3” marks, and average ball of</td>
</tr>
<tr>
<td></td>
<td>philosophy</td>
<td>each student</td>
</tr>
<tr>
<td>2</td>
<td>Information about cars for sale: type of car, type of engine, mileage of run, year of</td>
<td>Select the Ford cars with mileage of run less than 50000 km,</td>
</tr>
<tr>
<td></td>
<td>producing, starting price</td>
<td>which were produced more than two years ago</td>
</tr>
<tr>
<td>3</td>
<td>List of goods in the shop of electronics: code, name, producer, country-producer, year of</td>
<td>Select all refrigerators and calculate an average price of</td>
</tr>
<tr>
<td></td>
<td>producing, price</td>
<td>China TV</td>
</tr>
<tr>
<td>4</td>
<td>List of workers of enterprise: table number, last name and initials, position, date of</td>
<td>Output information about a programmer with the biggest salary,</td>
</tr>
<tr>
<td></td>
<td>employment, salary</td>
<td>and select all managers</td>
</tr>
<tr>
<td>5</td>
<td>List of books in a library: inventory number, name of book, author, year of publishing,</td>
<td>Select books with names, which begin with a word &quot;Programming&quot;,</td>
</tr>
<tr>
<td></td>
<td>price</td>
<td>and also books that were published more than 10 years ago</td>
</tr>
<tr>
<td>6</td>
<td>List of subscribers of telephone station: the last name of subscriber, telephone number,</td>
<td>Output information about subscribers that have a debt of more</td>
</tr>
<tr>
<td></td>
<td>license fee for a month, debt</td>
<td>200 ₴. Define the biggest debt</td>
</tr>
<tr>
<td>№ var.</td>
<td>Content of the binary file</td>
<td>Problem</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>List of printers for a sale: type of printer, firm-producer, speed of work (an amount of pages is in a minute), cost of printer</td>
<td>Output information about printers of firm “HP”, that print more than 10 pages in a minute</td>
</tr>
<tr>
<td>8</td>
<td>Results of competitions in athletics (100 m): name, gender, country, result (time)</td>
<td>Display information about Ukrainian athletes. Determine the leader in the distance of 100 m.</td>
</tr>
<tr>
<td>9</td>
<td>Data on employees: log number, name, position, gender, year of birth, marital status, number of children</td>
<td>Output information about single men. Calculate the number of employees who have more than 2 children</td>
</tr>
<tr>
<td>10</td>
<td>Results of the football tournament: team, country, city, number of victories, drawbacks and defeats</td>
<td>Output data about teams from Ukraine and calculate the total number of victories, drawbacks and defeats for them.</td>
</tr>
<tr>
<td>11</td>
<td>Results of athletics competitions: country, number of gold, silver and bronze medals</td>
<td>Calculate the total number of medals for each country. Identify the top three countries</td>
</tr>
<tr>
<td>12</td>
<td>List of students of the course: order number, surname and name, group, average point</td>
<td>Identify the best and worst students, and select students of inputted group</td>
</tr>
<tr>
<td>13</td>
<td>Employee data: log number, surname and name, position, gender, year of birth, marital status, number of children</td>
<td>Output information about single men. Calculate the number of employees with more than 3 children</td>
</tr>
<tr>
<td>14</td>
<td>Athletics Results (100m): surname, gender, country, result (time)</td>
<td>Output information about Ukrainian athletes. Define a leader among men and women</td>
</tr>
<tr>
<td>15</td>
<td>Bus ticket price information: destination, price, bus model, departure time</td>
<td>Output information about bus to Kiev, and find the most expensive ticket</td>
</tr>
<tr>
<td>17</td>
<td>Product list: serial number, name, manufacturer, price, quantity</td>
<td>Determine the total cost of each product and the total cost of all goods. Output the goods with the price over 1000 UAH.</td>
</tr>
<tr>
<td>18</td>
<td>Mobile phone store price list: firm, model, year, price</td>
<td>Output information first about all Nokia phones and then information on phones with a price of less than 2000 UAH.</td>
</tr>
</tbody>
</table>
**Recommended literature list**


# Content

Introduction .......................................................................................................................... 3

Lab № 6 One-dimensional array. Operating with arrays in functions .......................... 4
  Examples of programs ........................................................................................................ 4
  Control questions ............................................................................................................. 11
  Individual task .................................................................................................................. 11

Lab № 7 Two-dimensional arrays ......................................................................................... 14
  Examples of programs ........................................................................................................ 14
  Control questions ............................................................................................................. 20
  Individual task .................................................................................................................. 20

Lab № 8 Pointers and dynamic memory .............................................................................. 26
  Examples of programs ........................................................................................................ 26
  Control questions ............................................................................................................. 28

Lab № 9 Characters (char) and C-strings (char*) ............................................................... 31
  Examples of programs ........................................................................................................ 31
  Control questions ............................................................................................................. 35
  Individual task .................................................................................................................. 36

Lab 10 Text files ................................................................................................................... 38
  Examples of programs ........................................................................................................ 38
  Control questions ............................................................................................................. 41
  Individual task .................................................................................................................. 42

Lab 11 Structures .................................................................................................................. 43
  Examples of programs ........................................................................................................ 43
  Control questions ............................................................................................................. 46
  Individual task .................................................................................................................. 46

Lab 12 Binary files ............................................................................................................... 49
  Examples of programs ........................................................................................................ 49
  Control questions ............................................................................................................. 52
  Individual task .................................................................................................................. 52

Recommended literature list ............................................................................................... 54

Content ................................................................................................................................ 56