

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №5
з дисципліни
«Алгоритмізація та програмування»

Виконав:
студент групи КН-114
Добрій Назарій

Викладач:
Мочурад Л.І

Львів – 2019 р.

Тема: "Функції і масиви"

Мета: Організувати обробку масивів з використанням функцій, навчитися передавати масиви як параметри функцій

Теоретичні відомості:

Функцію в С можна розглядати:

- як один з похідних типів даних (поряд з масивами й вказівниками);
- як мінімальний виконавчий модуль програми (підпрограму).

Всі функції мають єдиний формат визначення: `()`, де `,` де - або `void`, якщо функція не повертає значення, або тип значення, що повертається функцією;; - або `main` для головної функції, або довільний ідентифікатор, що не збігається зі службовими словами й іменами інших об'єктів програми; - або порожній `()`, або список

Завдання 6

6. Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.

Скріни коду із завдання:

```
4  #include <iostream>
5  #include <ctime>
6
7  using namespace std;
8
9  void CreateMemory(int* arr[], int row, int col)
10 {
11     for (int i = 0; i < row; i++)
12     {
13         arr[i] = new int[col];
14     }
15 }
16
17 void FillArray(int* arr[], int row, int col)
18 {
19     for (int i = 0; i < row; i++)
20     {
21         for (int j = 0; j < col; j++)
22         {
23             arr[i][j] = rand() % 100;
24         }
25     }
26 }
27
28 void PrintArray2d(int* arr[], int row, int col)
29 {
30     for (int i = 0; i < row; i++)
31     {
32         for (int j = 0; j < col; j++)
33         {
34             cout << arr[i][j] << " ";
35         }
36         cout << endl;
37     }
38 }
39
40 void PrintArray(int arr[], int row)
41 {
42     for (int i = 0; i < row; i++)
```

```

40 void PrintArray(int arr[], int row)
41 {
42     for (int i = 0; i < row; i++)
43     {
44         cout << arr[i] << " ";
45     }
46 }
47
48 void DeleteMemory(int* arr[], int row)
49 {
50     for (int i = 0; i < row; i++)
51     {
52         delete[] arr[i];
53     }
54 }
55
56 int main() {
57     int row, col;
58     int maxRow = 0, minRow = 0;
59     int maxCol = 0, minCol = 0;
60     int i = 0, j = 0, n = 0;
61     int arrSecant[100];
62
63     bool a = true;
64
65     srand(time(NULL));
66
67     cout << "Enter a number of rows in massive: ";
68     cin >> row;
69
70     while (row <= 0)
71     {
72         cout << "Row must be bigger than 0!" << endl;
73         cout << "Enter a number of rows in massive: ";
74         cin >> row;
75     }
76
77     cout << "Enter a number of colums in massive: ";
78     cin >> col;
79
80     while (col <= 0)
81     {
82         cout << "Colum must be bigger than 0!" << endl;
83         cout << "Enter a number of colums in massive: ";
84         cin >> col;
85     }
86
87     cout << endl;
88
89     // CREATION OF DYNAMIC LOOPS //
90     int** arr = new int* [row]; // my main matrix
91     int* arrMaxRow = new int[row]; // max elements in ROW
92     int* arrMinRow = new int[row]; // min elements in ROW
93     int* arrMaxCol = new int[col]; // max elements in COL
94     int* arrMinCol = new int[col]; // min elements in COL;
95
96     CreateMemory(arr, row, col);
97
98     FillArray(arr, row, col);
99
100     PrintArray2d(arr, row, col);
101     cout << endl;
102
103     maxRow = arr[i][j]; // max in ROW
104
105     for (int i = 0, n = 0; i < row; i++, n++)
106     {
107         maxRow = arr[i][j];
108         for (int j = 0; j < col; j++)
109         {
110             if (arr[i][j] > maxRow)
111             {
112                 maxRow = arr[i][j];
113             }
114         }

```

```

112         maxRow = arr[i][j];
113     }
114 }
115 arrMaxRow[n] = maxRow;
116 j = 0;
117 }
118
119 cout << "The array of max element in ROW:" << endl;
120 PrintArray(arrMaxRow, row);
121 cout << endl;
122
123 minRow = arr[i][j]; // min in ROW
124
125 for (int i = 0, n = 0; i < row; i++, n++)
126 {
127     minRow = arr[i][j];
128     for (int j = 0; j < col; j++)
129     {
130         if (arr[i][j] < minRow)
131         {
132             minRow = arr[i][j];
133         }
134     }
135     arrMinRow[n] = minRow;
136     j = 0;
137 }
138
139 cout << "The array of min element in ROW:" << endl;
140 PrintArray(arrMinRow, row);
141 cout << endl;
142
143 maxCol = arr[i][j];
144
145 for (int j = 0, n = 0; j < col; j++, n++)
146 {
147     maxCol = arr[i][j];
148     for (int i = 0; i < row; i++)
149     {
150         if (arr[i][j] > maxCol)
151         {
152             maxCol = arr[i][j];
153         }
154     }
155     arrMaxCol[n] = maxCol;
156     i = 0;
157 }
158
159 cout << "The array of max element in COL:" << endl;
160 PrintArray(arrMaxCol, col);
161 cout << endl;
162
163 minCol = arr[i][j];
164
165 for (int j = 0, n = 0; j < col; j++, n++)
166 {
167     minCol = arr[i][j];
168     for (int i = 0; i < row; i++)
169     {
170         if (arr[i][j] < minCol)
171         {
172             minCol = arr[i][j];
173         }
174     }
175     arrMinCol[n] = minCol;
176     i = 0;
177 }
178
179 cout << "The array of min element in COL:" << endl;
180 PrintArray(arrMinCol, col);
181 cout << endl;
182
183 for (int i = 0; i < row; i++)
184 {
185     for (int j = 0; j < col; j++)
186     {

```

```

184     {
185         for (int j = 0; j < col; j++)
186         {
187             if (arrMaxRow[i] == arrMinCol[j])
188             {
189                 arrSecant[n] = arrMaxRow[i];
190                 n++;
191             }
192         }
193     }
194
195     for (int i = 0; i < row; i++)
196     {
197         for (int j = 0; j < col; j++)
198         {
199             if (arrMinRow[i] == arrMaxCol[j])
200             {
201                 arrSecant[n] = arrMinRow[i];
202                 n++;
203             }
204         }
205     }
206
207     if (n == 0)
208     {
209         cout << endl;
210         cout << "Your matrix doesnt have secant element!" << endl;
211     }
212
213     else
214     {
215         cout << endl;
216         cout << "Secant massive:" << endl;
217
218         /*for (int i = 0; i < n; i++)
219         {
220             cout << arrSecant[i] << endl;
221         }*/
222
223         /*for (int i = 0; i < n; i++)
224         {
225             cout << arrSecant[i] << endl;
226         }*/
227
228         int k = 0;
229
230         for (int i = 0; i < row; i++)
231         {
232             for (int j = 0; j < col; j++)
233             {
234                 if (arrSecant[k] == arr[i][j])
235                 {
236                     if (a == true)
237                     {
238                         for (int u = 0; u < col; u++)
239                         {
240                             if (arrSecant[k] == arrMaxCol[u])
241                             {
242                                 cout << arrSecant[k] << endl;
243                                 cout << "row is - " << i << endl;
244                                 cout << "col is - " << j << endl;
245                                 k++;
246                                 a = false;
247                             }
248                         }
249                     }
250
251                     if (a == true)
252                     {
253                         for (int d = 0; d < col; d++)
254                         {
255                             if (arrSecant[k] == arrMinCol[d])
256                             {
257                                 cout << arrSecant[k] << endl;
258                                 cout << "row is - " << i << endl;
259                                 cout << "col is - " << j << endl;
260                                 k++;

```

```

253         cout << "row is - " << i << endl;
254         cout << "col is - " << j << endl;
255         k++;
256         a = false;
257     }
258 }
259 }
260
261     if (a == true)
262     {
263         for (int t = 0; t < col; t++)
264         {
265             if (arrSecant[k] == arrMaxRow[t])
266             {
267                 cout << arrSecant[k] << endl;
268                 cout << "row is - " << i << endl;
269                 cout << "col is - " << j << endl;
270                 k++;
271                 a = false;
272             }
273         }
274     }
275
276     if (a == true)
277     {
278         for (int m = 0; m < col; m++)
279         {
280             if (arrSecant[k] == arrMinRow[m])
281             {
282                 cout << arrSecant[k] << endl;
283                 cout << "row is - " << i << endl;
284                 cout << "col is - " << j << endl;
285                 k++;
286                 a = false;
287             }
288         }
289     }
290 }
291 a = true;
292
293     }
294     cout << endl;
295 }
296
297 cout << endl;
298
299 /// DELETE A MEMORY ///
300 DeleteMemory(arr, row);
301
302 delete[] arr;
303 delete[] arrMaxRow;
304 delete[] arrMinRow;
305 delete[] arrMaxCol;
306 delete[] arrMinCol;
307 ///////////////////////////////////////////////////
308
309 system("pause");
310 return 0;
311 }
312

```

Скріни реалізацій завдання:

```
C:\Users\User\source\repos\198\Debug\198.exe
Enter a number of rows in massive: 3
Enter a number of colums in massive: 3

95  47  69
57  16  92
34  44  74

The array of max element in ROW:
95 92 74
The array of min element in ROW:
47 16 34
The array of max element in COL:
95 47 92
The array of min element in COL:
34 16 69

Secant massive:
47
row is - 0
col is - 1

Press any key to continue . . .
```

Висновок: я навчився передавати масив у функцію як параметр.