

**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:
Мочурad 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 сідовою 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 {
58     int row, col;
59     int maxRow = 0, minRow = 0;
60     int maxCol = 0, minCol = 0;
61     int i = 0, j = 0, n = 0;
62     int arrSecant[100];
63
64     bool a = true;
65
66     srand(time(NULL));
67
68     cout << "Enter a number of rows in massive: ";
69     cin >> row;
70
71     while (row <= 0)
72     {
73         cout << "Row must be bigger than 0!" << endl;
74         cout << "Enter a number of rows in massive: ";
75         cin >> row;
76     }
77
78     cout << "Enter a number of columns in massive: ";
79     cin >> col;
80
81     cout << "Enter a number of columns in massive: ";
82     cin >> col;
83
84     while (col <= 0)
85     {
86         cout << "Column must be bigger than 0!" << endl;
87         cout << "Enter a number of columns in massive: ";
88         cin >> col;
89     }
90
91     cout << endl;
92
93     /// CREATION OF DYNAMIC LOOPS ///
94     int** arr = new int* [row]; // my main matrix
95     int* arrMaxRow = new int[row]; // max elements in ROW
96     int* arrMinRow = new int[row]; // min elements in ROW
97     int* arrMaxCol = new int[col]; // max elements in COL
98     int* arrMinCol = new int[col]; // min elements in COL;
99
100    CreateMemory(arr, row, col);
101
102    FillArray(arr, row, col);
103
104    PrintArray2d(arr, row, col);
105    cout << endl;
106
107    maxRow = arr[i][j]; // max in ROW
108
109    for (int i = 0, n = 0; i < row; i++, n++)
110    {
111        maxRow = arr[i][j];
112        for (int j = 0; j < col; j++)
113        {
114            if (arr[i][j] > maxRow)
115            {
116                maxRow = arr[i][j];
117            }
118        }
119    }

```

```

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             for (int i = 0; i < row; i++)
153             {
154                 if (arr[i][j] > maxCol)
155                 {
156                     maxCol = arr[i][j];
157                 }
158             }
159         }
160     }
161     arrMaxCol[n] = maxCol;
162     i = 0;
163 }
164
165 cout << "The array of max element in COL:" << endl;
166 PrintArray(arrMaxCol, col);
167 cout << endl;
168
169 minCol = arr[i][j];
170
171 for (int j = 0, n = 0; j < col; j++, n++)
172 {
173     minCol = arr[i][j];
174     for (int i = 0; i < row; i++)
175     {
176         if (arr[i][j] < minCol)
177         {
178             minCol = arr[i][j];
179         }
180     }
181     arrMinCol[n] = minCol;
182     i = 0;
183 }
184
185 cout << "The array of min element in COL:" << endl;
186 PrintArray(arrMinCol, col);
187 cout << endl;
188
189 for (int i = 0; i < row; i++)
190 {
191     for (int j = 0; j < col; j++)
192     {

```

```
184
185     {
186         for (int j = 0; j < col; j++)
187         {
188             if (arrMaxRow[i] == arrMinCol[j])
189             {
190                 arrSecant[n] = arrMaxRow[i];
191                 n++;
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 doesn't 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
224         int k = 0;
225
226         for (int i = 0; i < row; i++)
227         {
228             for (int j = 0; j < col; j++)
229             {
230                 if (arrSecant[k] == arr[i][j])
231                 {
232                     if (a == true)
233                     {
234                         for (int u = 0; u < col; u++)
235                         {
236                             if (arrSecant[k] == arrMaxCol[u])
237                             {
238                                 cout << arrSecant[k] << endl;
239                                 cout << "row is - " << i << endl;
240                                 cout << "col is - " << j << endl;
241                                 k++;
242                                 a = false;
243                             }
244                         }
245
246                     if (a == true)
247                     {
248                         for (int d = 0; d < col; d++)
249                         {
250                             if (arrSecant[k] == arrMinCol[d])
251                             {
252                                 cout << arrSecant[k] << endl;
253                                 cout << "row is - " << i << endl;
254                                 cout << "col is - " << j << endl;
255                                 k++;
256                             }
257                         }
258                     }
259                 }
260             }
261         }
262     }
263 }
```

```

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 cout << arrSecant[k] << endl;
293 cout << "row is - " << i << endl;
294 cout << "col is - " << j << endl;
295 k++;
296 a = false;
297 }
298 cout << endl;
299
300 /// DELETE A MEMORY ///
301 DeleteMemory(arr, row);
302
303 delete[] arr;
304 delete[] arrMaxRow;
305 delete[] arrMinRow;
306 delete[] arrMaxCol;
307 delete[] arrMinCol;
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 columns 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.