<u>Практикум по курсам «Основы информатики», «Алгоритмы и структуры данных»</u> Лабораторная работа № 14 на 2011/12 учебный год: 8 факультет, 1 курс, 1 семестр

Вложенные циклы с параметрами. Обход и линеаризация матриц

Составить программу ввода квадратной матрицы и печати в строку всех ее элементов в заданном ниже порядке следования (обхода). Тип элементов матрицы – целый. Решить задачу на языке Си с применением вложенных циклов с переменными границами. Тестировать программу на нескольких примерах матриц небольшого порядка (до 7×7) в одном пакете тестов. Работа может выполняться с автоматическим тестированием в системе TEST999. В этом случае к формату ввода и вывода данных предъявляются дополнительные требования.

Пример результатов работы для варианта 21:

Введенная матрица:				Напечатанная строка:															
11 1	2 13 1	4																	
	22 23 2			11 12 13 14 24 34 44 43 42 41 31 21 22 23 33 32															
	32 33 3																		
41 4	2 43 4	4																	
	2 13 1																		
	22 23 2				11	12 13	14 15 2	25 3	35 45 :	55 54 :	53 52 :	51 41 3	1 2	21 22 2	3 24 3	4 44 4	3 42 32	! 33	
	2 33 3																		
	12 43 4 52 53 5																		
313	,2 33 3	7 33			Варианты заданий														
1				2					3					4					
7	11	14	16	$\frac{7}{7}$	4	2	1		16	15	13	10	1	1	3	6	10		
4	8	12	15	11	8	5	3		14	12	9	6		2	5	9	13		
2	5	9	13	14	12	9	6		11	8	5	3		4	8	12	15		
1_	3	6	10	16	15	13	10		7	4	2	1		7	11	14	16		
5				6				1	7					8					
10	11 9	15 12	16	10	4	3	1		16	14 12	13	7		1	2 5	6	7		
4 3	5	12 8	14 13	11 15	9 12	5 8	2 6		15 11	12 9	8 5	6 2		$\begin{vmatrix} 3 \\ 4 \end{vmatrix}$	9	8 12	13 14		
$\begin{vmatrix} 3 \\ 1 \end{vmatrix}$	2	6	7	16	14	13	7		10	4	3	1		10	11	15	16		
9				10				_	11					12					
7	13	14	16	7	6	2	1	1	16	15	11	10	1	1	3	4	10		
6	8	12	15	13	8	5	3		14	12	9	4		2	5	9	11		
2	5	9	11	14	12	9	4		13	8	5	3		6	8	12	15		
1	3	4	10	16	15	11	10		7	6	2	1		7	13	14	16		
13	12		4	14		12	4.5	1	15	- 1.1	40		1	16			4.5		
15 14	13 6	7 2	1 10	$\begin{array}{ c c c }\hline 1\\10 \end{array}$	7 2	13 6	15 14		16 13	14 9	10 3	4 5		4 10	5 3	11 6	15 12		
5	3	9	11	111	9	3	5		8	2	6	11		14	9	2	7		
4	8	12	16	16	12	8	4		1	7	12	15		16	13	8	1		
17				18					19					20					
7	6	5	16	7	8	9	10		1	2	9	10		13	12	11	10		
8	1	4	15	6	1	2	11		4	3	8	11		14	7	8	9		
9 10	2 11	3 12	14 13	5 16	4 15	3 14	12 13		5 16	6 15	7 14	12 13		15 16	6 5	3 4	2 1		
21	11	14	13	22	13	14	13	1	23	13	14	13	ı	24		+	1		
1	2	3	4	1	12	11	10	1	4	3	2	1	1	4	5	6	7		
12	13	14	5		13	16	9		5	14	13	12		3	14	15	8		
11	16	15	6	3	14	15	8		6	15	16	11		2	13	16	9		
10	9	8	7	4	5	6	7		7	8	9	10		1	12	11	10		
25				26				-	27					28					
7	6	5	4	7	8	9	10		10	11	12	1		10	9	8	7		
8	15 16	14	3	6	15 14	16	11		9	16 15	13	2		11 12	16	15 14	6		
9 10	16 11	13 12	2 1	5 4	14 3	13 2	12 1		8 7	15 6	14 5	3 4		12	13 2	14 3	5 4		
10	11	14	1	, <u>, </u>	<i>J</i>	4	1	1		υ	J	7	ı		<u> </u>	<u> </u>	7		

Примеры.

```
/* Одномерная интерпретация матрицы.
   При нумерации от 0 номер (і, ј)-ого элемента в линеаризованном массиве очень
   просто выражается через і и ј:
       k = i * n + j
   Обратная операция тоже очень проста:
       i = k / n
       j = k % n. */
#include <stdio.h>
const int maxn = 100;
int main()
  int i, j, n, k;
  int m[maxn * maxn];
  scanf("%i", &n);
  for (k = 0; k != n * n; ++k)
    scanf("%d", m + k); /* вводится k-ое значение, считая от начала
                            массива, т.е. массив в Си отождествляется
                            с адресом его начала */
  for (k = 0; k != n * n; ++k) {
    printf("%d ", m[k]); /* выводится k-ый элемент массива */
    if (k % n == n - 1) putchar('\n'); /* переход на новую строку
                                           листинга по окончании вывода
                                           элементов строки матрицы */
  }
}
Загадка: какому из предложенных вариантов соответствует эта программа?
#include <stdio.h>
enum { maxn = 100 };
int main()
  int m[maxn] [maxn];
  int dir[4] = \{ 1, 0, -1, 0 \};
  int n;
  scanf("%d", &n);
  for (int i = 0; i != n; ++i)
    for (int j = 0; j != n; ++j)
      scanf("%d", &m[i][j]);
  int 1 = n;
  int p = 0;
  int i = 0, j = -1;
  while (l != 0) {
    for (int k = 0; k != 1; ++k) {
      j += dir[p % 4];
      i += dir[(p+3) % 4];
      printf("%d ", m[i][j]);
    }
    ++p;
    l -= p % 2;
  putchar('\n');
```