ПРАКТИЧЕСКАЯ работа №7. **Динамическое распределение памяти**

***Вариант №1***

Разработать программу перемножения двух матриц *A, B* размерности *n*х*n*. Все матрицы размещаются в оперативной памяти динамически, а значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

***Вариант №2***

Разработать программу нормирования матрицы размерности *m*х*n*, которое заключается в том, что каждый элемент в этой матрице вычисляется на основании исходной матрицы, как отношение суммы всех других элементов в его строке к сумме всех других элементов в его столбце. Матрица должна размещаться в оперативной памяти динамически, значения *m* и *n* вводятся с клавиатуры по запросу. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №3***

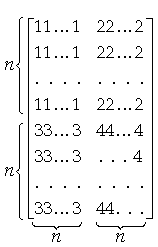
Разработать программу, которая в матрице размерности *n*х*n* меняет местами строку, содержащую элемент с минимальным значением со столбцом, содержащим элемент с максимальным значением. Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №4***

Дана действительная квадратная матрица порядка *n*, все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей. Матрица размещается в памяти, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №5***

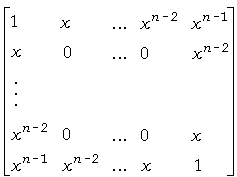
Построить квадратную матрицу порядка 2*n*:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №6***

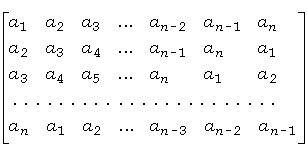
Дано действительное число *x*. Получить квадратную матрицу порядка *n*<10:



Матрица размещается в памяти динамически, значения *x* и *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №7***

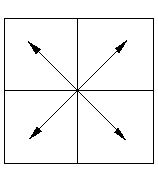
Даны действительные числа *a*1, …, *an*. Получить квадратную матрицу порядка *n:*



Матрица размещается в памяти динамически, значение *n* и числа *a*1, …, *an* вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №8***

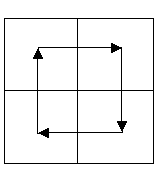
Дана действительная квадратная матрица порядка 2*n*. Получить новую матрицу, переставляя ее блоки размера *n*x*n*:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №9***

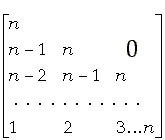
Дана действительная квадратная матрица порядка 2n. Получить новую матрицу, переставляя ее блоки размера *n*x*n*:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №10***

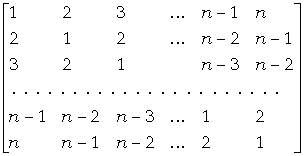
Получить квадратную матрицу порядка *n*:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №11***

Получить квадратную матрицу порядка *n*:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №12***

Дана действительная квадратная матрица порядка *n*. Преобразовать матрицу по правилу: строку с номером *n* сделать столбцом с номером *n*, а столбец с номером *n* сделать строкой с номером *n*. Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №13***

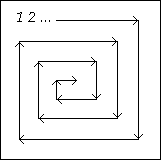
Даны две действительные квадратные матрицы порядка *n*. Получить новую матрицу умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы. Матрицы размещаются в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

***Вариант №14***

Даны две действительные квадратные матрицы порядка *n*. Получить новую матрицу прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы. Матрицы размещаются в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

***Вариант №15***

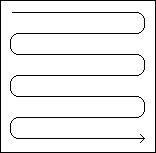
Получить целочисленную квадратную матрицу порядка *n*<8, элементами которой являются числа 1, 2, ..., *n*2, расположенные в ней по спирали:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №16***

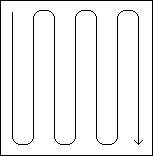
Даны действительные числа . Получить действительную квадратную матрицу порядка *n*<8, элементами которой являются числа , расположенные в ней по схеме:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №17***

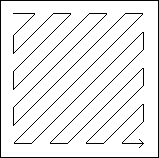
Даны действительные числа . Получить действительную квадратную матрицу порядка *n*<8, элементами которой являются числа , расположенные в ней по схеме:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №18***

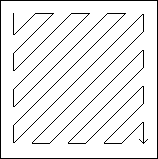
Даны действительные числа . Получить действительную квадратную матрицу порядка *n*<8, элементами которой являются числа , расположенные в ней по схеме:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №19***

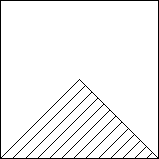
Даны действительные числа . Получить действительную квадратную матрицу порядка *n*<8, элементами которой являются числа , расположенные в ней по схеме:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

***Вариант №20***

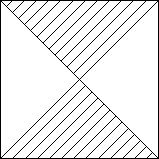
Дана действительная квадратная матрица порядка *n*. Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

***Вариант №21***

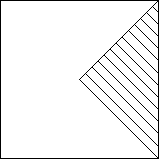
Дана действительная квадратная матрица порядка *n*. Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

***Вариант №22***

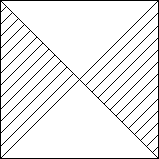
Дана действительная квадратная матрица порядка *n*. Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

***Вариант №23***

Дана действительная квадратная матрица порядка *n*. Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

***Вариант №24***

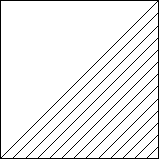
Дана целочисленная матрица размера *m*x*n*. Найти матрицу, получающуюся из данной перестановкой столбцов – первого с последним, второго с предпоследним и т.д. Матрица размещается в памяти динамически, значения *m* и *n* вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №25***

Дана целочисленная матрица размера размера *m*x*n*. Найти матрицу, получающуюся из данной перестановкой строк – первой с последней, второй с предпоследней и т.д. Матрица размещается в памяти динамически, значения *m* и *n* вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

***Вариант №26***

Дана действительная квадратная матрица порядка *n*. Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

***Вариант №27***

Дана действительная квадратная матрица порядка *n*, все элементы которой различны. В этой матрице в каждой строке элементы, стоящие на нечетных местах, заменить суммой, на четных – произведением соответствующей пары. Матрица размещается в памяти динамически, значение *n* вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.