**Содержание**

1 Постановка задачи

1.1 Математическая модель задачи

1.2 Входные данные

1.3 Выходные данные

1.4 Обработка ошибок

**1 Постановка задачи**

**1.1 Математическая модель задачи**

Массив – это коллекция переменных, которые имеют общее имя и базовый тип. Массив является удобным способом хранения нескольких связанных элементов данных в едином контейнере для большего удобства и эффективности программирования. Все элементы данных, сохраняемых в массиве, должны иметь один и тот же тип; например при создании массива для хранения типов Integer(Double, String, Currency и т. д.) все элементы данных, сохраненных в этом массиве, должны быть численными Integer(Double, String, Currency и т. д.)

Массив позволяет сохранять и манипулировать многими элементами данных посредством единственной переменной. Кроме уменьшения общего числа различных имен переменных, которые необходимо отслеживать, другим основным преимуществом использования массивов является то, что можно использовать циклы для легкой обработки различных элементов массивов. Объединяя массивы и структуры цикла (обычно For...Next или For...Each), можно записать небольшое число операторов, которые обрабатывают большой объем данных. Выполнения тех же задач с использованием отдельных переменных может потребовать написания сотен операторов.

Массивы бывают статические и динамические. Статическими называют массивы, количество элементов которых заранее известно и не изменяется в ходе выполнения программы. Динамические массивы – массивы, в которых либо не известно начальное количество элементов, либо размерность массива(количество элементов) изменяется при выполнении программы.

Описание массивов:

1. одномерный статический массив:

Dim<имя массива> (<начальное значение индекса> To <конечное значение индекса>) [As <тип элементов массива>]

Или

Dim <имя массива> (<количество элементов массива >) [As<тип элементов массива>];

1. двумерный статический массив

Dim <имя массива> (<начальное значение индекса по строкам> To <конечное значение индекса по строкам >, <начальное значение индекса по столбцам> To <конечное значение индекса по столбцам >) [As<тип элементов массива>]

Или

Dim <имя массива> (<количество строк>, <количество столбцов>) [As <тип элементов массива>].

Первый способ отличается от второго тем, что в первом случае указывается индекс первого и последнего элементов, во втором же – только количество элементов, нумерация которых может начинаться как с 0, так и с 1. Это зависит от опции Base (задаёт базовый индекс). Если опция не указана. То нумерация элементов массива начинается с нуля. Для изменения базового индекса в начале листа модуля необходимо написать Option Base 1.

Пример:

а)Dim A(1To 10) As Integer –массив A состоит из 10 элементов целого типа, индексы которых 1, 2, …, 10;

б)Dim A(10) As Integer – массив состоит из 10 значений целого типа. Индексация зависит от опции Base. Если опция не указана, то номера элементов – от 0 до 9, если же указана (т.е. вначале модуля записано Option Base 1), то номера элементов изменяются от 1 до 10;

3) динамический массив:

Dim <имя массива> ( ) [As <тип элементов массива>].

После определения количества элементов массива выполняется его переопределение:

ReDim <имя массива> (<задаётся размерность массива (одномерного/двумерного)>).

Пример:

Dim A ( ) As Single-динамический массив А вещественных элементов:

n = 7

ReDim A (1 To n) – переопределение одномерного массива из n значений

ReDim A (5, n) – переопределение двумерного динамического массива, состоящего из 5 строк и n столбцов (начало индексации элементов определяется по опции Base).

Обращение к элементу массива осуществляется следующим образом: указывается имя массива, а затем в круглых скобках указывается номер элемента в массиве. Если массив двумерный – указывается вначале номер строки, затем через запятую номер столбца.

1. Для нахождения суммы элементов массива с нечетными номерами мы будем суммировать все элементы исходного массива индексы (i и j), которых являются нечетными.
2. Для нахождения суммы элементов массива, расположенных между первым и последним отрицательными элементами необходимо найти первые и последние отрицательные элементы для строки и столбца, а затем их просуммировать.
3. Для нахождения суммы элементов А[i,j], для которых i+j=k (k – целое число, вводимое пользователем) проверяем верность равенства, и если оно истинно, то вычисляем искомую сумму.

Матрицей называется система m·n чисел, расположенных в прямоугольной таблицы из m строк и n столбцов. Числа этой таблицы называются элементами матрицы. Обозначение матрицы:

, , .

Элементы ai1, ai2, ..., ain составляют i-ю строку (i=1,2,... m), элементы a1k ,a2k , ..., amk - k-й столбец (k=1, 2 , ..., n), aik – элемент, принадлежащий i – й строке и k – му столбцу матрицы, числа i, k называют индексами элемента. Строки и столбцы матрицы называют её рядами; под двумя параллельными рядами понимают две строки или два столбца матрицы. Матрицу, имеющую m строк и n столбцов, называют матрицей размеров m×n. Употребляются и более краткие обозначения матрицы размеров m×n: , , . Матрицу обозначают так же одной заглавной буквой, например,

A=, B=.

Если необходимо отметить, что матрица А имеет mстрок и nстолбцов, т.е. необходимо указать её размеры, то пишут Аm×n, или Аmn.

**1.2 Входные данные**

Входные данные:

-значения размерности массива А

- значения элементов массива

Требования к входным данным:

-значения размерности массива и число k должны быть целыми положительными числами

-ввод размерности массива осуществляется в соответствующие поле на диалоговой форме

-значения элементов массива должны быть числовыми

**1.3 Выходные данные**

Выходные данные:

- сумма элементов c нечетными номерами;

- сумма элементов, расположенных между первыми и последними отрицательными элементами;

- сумма элементов А[i,j], для которых i+j=k (k – целое число, вводимое пользователем).

Требования к выходным данным:

- все выходные данные выводятся на лист Excel и на форму.

**1.4 Обработка ошибок**

При составлении приложений важно предусмотреть, чтобы программа анализировала возможные ошибки, возникающие при её выполнении по вине пользователя, и информировало его о причине возникновения ошибки:

-неправильность ввода значений элементов массива;

- неправильность ввода размерности массива.

Для перехвата все возможных ошибок используется оператор IF THEN.