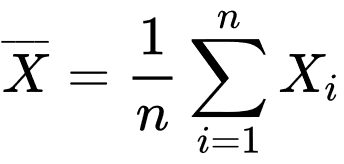
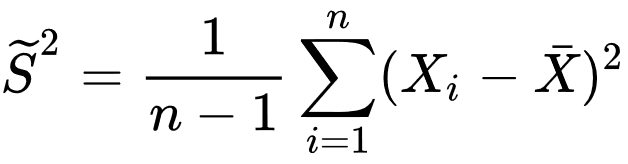
Массивы

Рассмотрим следующую практическую задачу:

Необходимо рассчитать среднее арифметическое и дисперсию выборки из 5 чисел





При использовании всех накопленных знаний за занятия мы решили бы задачу следующим образом:

#include <stdio.h>

#include <math.h>

int main()

{

int x1, x2, x3, x4, x5;

scanf\_s("%d", &x1);

scanf\_s("%d", &x2);

scanf\_s("%d", &x3);

scanf\_s("%d", &x4);

scanf\_s("%d", &x5);

double aver = (x1 + x2 + x3 + x4 + x5) / 5;

double var = (pow(x1 - aver, 2) + pow(x2 - aver, 2) + pow(x3 - aver, 2) + pow(x4 - aver, 2) + pow(x5 - aver, 2))/4;

printf("Aver=%lf\n", aver);

printf("Var=%lf\n", var);

return 0;

}

Недостатком такого подхода является неуниверсальность и немасштабируемость кода. Если будет необходимо рассчитать среднее значение и дисперсию из 100 чисел, то нам необходимо было бы создавать 100 переменных.

Для решения такой проблемы используются массивы.

**Массив** - структура данных, в которой хранится фиксированное число однотипных элементов.

**Элемент массива** – значение, хранящееся в определенной ячейке памяти, расположенной в пределах массива, а также адрес этой ячейки памяти.

В общем виде синтаксис массива выглядит следующим образом:

тип переменная[количество элементов]

Например, объявление массива array из 10 целых элементов выглядит следующим образом:

int array[10];

Количество элементов должно быть установлено до начала компиляции. Другими словами, это должно быть константное значение.[[1]](#footnote-1)

Для определения массива, возможно указать его значения в фигурных скобках. Если указано только **одно** значение, то **весь** массив будет им заполнен. Если указать несколько значений, то они перенесутся в массив, а оставшиеся элементы будут заполнены нулями. При определении массива возможно оставить значение в скобках пустым, если после будет идти перечисление его элементов в скобках. Размер массива будет равняться числу этих элементов.

Например:

int array[5] = {1,2,3,4,5}; //{1,2,3,4,5}

int array[5] = {0}; //{0,0,0,0,0}

int array[5] = {1,2,3}; //{1,2,3,0,0}

char array[] = {'h','e','l','l','o'}; //Число элементов = 5

Для доступа к элементу массива необходимо указать его имя и номер порядкового элемента в скобках. В языке С отсчет элементов идет с **0**.

Рассмотрим пример вывода элементов массива на экран. Обратите внимание, что размер массива задан константным значением. В цикле происходит вывод элементов на экран, начиная с элемента array[0] равного 1.

#include <stdio.h>

const int N = 5;

int main()

{

int array[N] = {1,2,3,4,5};

for (int i = 0; i < N; i++)

{

printf\_s("%d ", array[i]);

}

printf\_s("\n");

return 0;

}

Решение задачи выше будет выглядеть следующим образом:

#include <stdio.h>

#include <math.h>

const int N = 5;

int main()

{

int x[N];

double aver = 0, var = 0;

for (size\_t i = 0; i < N; i++)

scanf\_s("%d", &x[i]);

for (size\_t i = 0; i < N; i++)

aver = aver + x[i];

aver = aver / N;

for (size\_t i = 0; i < N; i++)

var = var + pow(x[i] - aver, 2);

var = var / (N - 1);

printf("Aver=%lf\n", aver);

printf("Var=%lf\n", var);

return 0;

}

Пример:

Ввести массив размером N. Найти в нем максимальный и минимальный элемент. Вывести его на экран.

#include <stdio.h>

const int N = 5;

int main()

{

int x[N];

int Max, Min;

for (int i = 0; i < N; i++)

scanf\_s("%d", &x[i]);

for (int i = 0; i < N; i++)

printf("X[%d]=%d\n", i, x[i]);

Max = x[0];

Min = x[0];

for (int i = 1; i < N; i++)

{

if (x[i] < Min)

Min = x[i];

if (x[i] > Max)

Max = x[i];

}

printf("Min = %d\nMax = %d\n", Min, Max);

return 0;

}

Еще раз об отличии ++i и i++

#include <stdio.h>

int main()

{

int A;

int x = 0;

A = x++;

// A = x;

// x = x + 1;

printf("A = %d\nX = %d\n", A, x);

x = 0;

A = ++x;

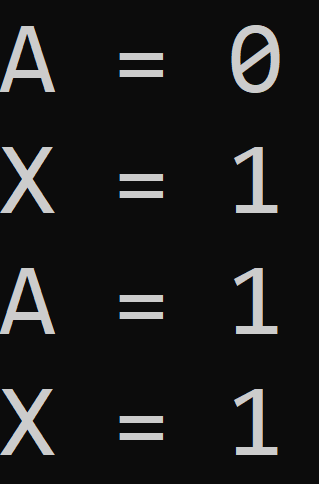
// x = x + 1;

// A = x;

printf("A = %d\nX = %d\n", A, x);

return 0;

}



Язык программирования С допускает использование многомерных массивов. В общем виде синтаксис выглядит следующим образом:

тип переменная[кол-во элементов][кол-во элементов]...[кол-во элементов]

Чаще всего встречаются двумерные массивы или матрицы.

Например, матрицу 3\*4 можно объявить следующим образом:

int array[3][4];

Выглядеть она будет так:

Изображение выглядит как стол

Автоматически созданное описание

Матрица содержит 3 строки и 4 столбца. Первое число в квадратных скобках отвечает за строки, второе – за столбцы.

Инициализировать матрицу также можно разными способами.

Например, так:

int array[N][N] = {

{1,2,3},

{4,5,6},

{7,8,9}

};

Или так:

int array[N][N] = {1,2,3,4,5,6,7,8,9};

Для работы с матрицами необходимо два цикла:

#include <stdio.h>

const int N = 3;

int main()

{

int array[N][N] = {

{1,2,3},

{4,5,6},

{7,8,9}

};

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++)

{

printf\_s("%d ", array[i][j]);

}

printf\_s("\n");

}

return 0;

}

1. На самом деле, это не совсем так. Со стандарта С99 можно создавать массивы с заранее неизвестным числом аргументов, но пока лучше так не делать. [↑](#footnote-ref-1)