**Семинар 1. Статические одномерные массивы. Поэлементная и выборочная обработка массивов.**

**Цель работы:** Освоение приемов обработки одномерных массивов.

Массивы в процедурном С/С++ представляют собой последовательность ***элементов одного типа***, доступ к которым может осуществляться либо с указанием индексов (целого или символьного типов), либо с использованием возможностей адресной арифметики.

Согласно правилам С/С++ все массивы состоят из смежных ячеек памяти. Младший адрес соответствует первому элементу массива, а старший – последнему. Индексы массива всегда начинаются с 0.

Массивы и указатели тесно связаны между собой. Трудно описывать и обрабатывать массивы, не используя указателей.

Объявление массива:

**<Тип элемента> <Имя> [<Размер1>] [= {<Список значений** **>}];**

Количество индексов задает *размерность* массива. Каждому элементу массива соответствует один или несколько *индексов*, определяющих положение элемента в массиве.

*Тип индекса* – порядковый – определяет доступ к элементу.

Нумерация индексов **ВСЕГДА** начинается с 0.

*Размер* – определяет количество элементов по данному индексу.

*Тип элемента* – любой кроме файла, в том числе, другой массив.

Массив в памяти не может занимать более 2 Гб.

Объявление одномерных массивов:

1. **Статически**

int a[10]; - массив на 10 целых чисел**;//** индекс меняется 0 - 9

float mas[20] – массив на 20 вещественных чисел**;**

char sim[8] – массив на 8 символов**;**

double massiv[30] – массив на 30 вещественных чисел двойной точности**;**

unsigned int koord[10] – массив целых беззнаковых чисел .

1. Динамически

int \*dinmas; указатель на целое;

dinmas=new int [100]; массив на 100 элементов целого типа.

В последних стандартах языка появились, так называемые – *автоматические массивы.*

3) **Автоматический массив:**

под массив, локально объявленный внутри подпрограммы, память выделяется в стеке *во время выполнения* программы.

*Примечание*. Допускается в среде Qt Creator (компилятор Clang) указывать размер локального массива переменными:

**int main() {**

**int n;**

**cout << "Enter n:";**

**cin >> n ;**

**float a[n];** //Определять массив можно только после ввода размера!!

Однако, не все компиляторы поддерживают описанный синтаксис, поэтому, в лабораторных, ДЗ и РК их лучше не использовать!!

**Индекс массива меняется от 0 до величины, на 1 меньшей указанной в размере**

Внешние и статические массивы можно инициализировать при объявлении.

int a[5]={0,-36,78,3789,50};

float b[10]={0,-3.6,7.8,3.789,5.0,6.1,0,-6.5,8.9,3.0};

long double c[4]={7.89L,6.98L,0.5L,56.8L};

short \*m={2,3,5,8,12,0,56};

По правилам С\С++ **имя массива** является его адресом.

Поэтому, для адресации элементов массива независимо от способа описания можно использовать адресную арифметику.

Однако, в этом семестре мы будем работать только со статическими массивами.

Так как в настоящее время пользователь ***не имеет*** доступа к исходным текстам программ, описание массива вида

**const int N=20;**

или **#define N 20;**

int masiv[N];

*использовать не рекомендуется!!!.*

***Основные приемы обработки массивов***

1. Однотипная обработка массивов.

* Поэлементная.
* Выборочная.

1. Переформирование массивов.

* Без изменения размеров массива.
* С изменением размеров массива.

1. Одновременная обработка нескольких массивов и/или подмассивов.

* С синхронным изменением индексов.
* С асинхронным изменением индексов.

1. Поисковые задачи.

**Задача 1.**

Сформировать массив целых чисел с применение датчикам случайных чисел. Написать программу нахождения максимального элемента этого массива. Определить все элементы, равные максимальному, этого массива. Вывести исходный массив и всю необходимую информацию

**Анализ задания.**

Для решения задачи необходимо:

- сформировать массив или ввести с клавиатуры;

-вывести исходный массив;

- найти максимальный элемент;

-найти и вывести ВСЕ элементы, равные максимальному, и их индексы.

Как видно из анализа, для решения всеx подзадач можно использовать **поэлементную** обработку массива.

**Текст программы**

**#include <stdio.h>**

**#include <conio.h>**

**#include <time.h>**

**#include <stdlib.h>**

**int main()**

**{ int a[100];**

**int i,n,k;**

**srand((unsigned)time(NULL));// установка датчика сл. чисел**

**puts("Input n<=100");**

**scanf("%d",&n);**

**//printf("input %3d elementov massiva\n",n);**

**for(i=0;i<n;i++) //scanf("%d",&a[i]);**

**a[i]=rand()%11;//-rand()%11;**

**printf("\n");**

**puts("Input k for output"); // для вывода в несколько строк**

**scanf("%d",&k);**

**puts("Inputed Massiv");**

**for(i=0;i<n;i++)**

**printf("%5d%c",a[i],((i+1)%k==0)?'\n':' ');**

**printf("\n");**

**int amax=a[0],imax=0;// установка начального значения максимума**

**for(i=1;i<n;i++)**

**if(a[i]>amax)**

**{ imax=i;**

**amax=a[i];**

**}**

**printf("amax=%5d imax=%5d\n",amax,imax+1);**

**puts("ALL MAX A");**

**for(i=1;i<n;i++)**

**if(a[i]==amax)**

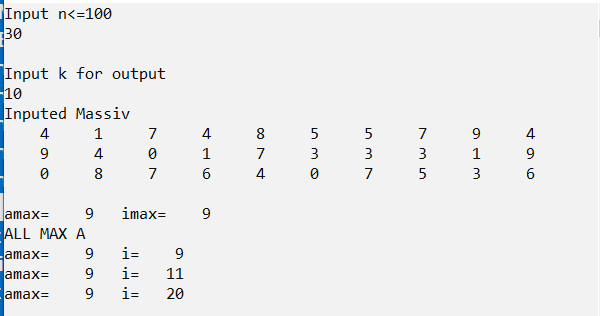
**printf("amax=%5d i=%5d\n",amax,i+1);**

**getch();**

**return 0;**

**}**

**Примеры работы программы**

****

**Задача 2.**

Сформировать массив не более 100 элементов. В нем найти количество отрицательных элементов, кратных 3 поместить его на место элемента с номер 5. Вывести индексы отрицательных элементов и массив до и после преобразования.

**Анализ задания.**

Для решения задачи необходимо:

- сформировать массив или ввести с клавиатуры;

-вывести исходный массив;

- найти количество отрицательных элементов, кратных 3;

-заменить 5 элемент массива найденным количеством;

- вывести полученный массив.

Как видно из анализа, для формирования, вывода массивов и поиска количества отрицательных элементов, кратных 3, можно использовать поэлементную обработку массива. А для замещения пятого элемента – переформирование массива без изменения его размеров.

**Текст программы**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<conio.h>**

**#include<time.h>**

**int a[100];**

**int i,j,n,kol3=0;**

**void main(int argc, char\* argv[])**

**{ srand((unsigned)time(NULL));**

**puts("Input n <=100");**

**scanf("%d",&n);**

**for(i=0;i<n;i++)**

**a[i]=rand()%51-rand()%50;**

**puts("Inputed Massiv");**

**for(i=0;i<n;i++)**

**printf("%4d”, a[i]);**

**printf("\n");**

**// поиск элементов**

**for(i=0;i<n;i++)**

**if((a[i]<0)&&( a[i]%3==0))**

**{ kol3++;**

**printf("%2d -- %4d\n",i+1, a[i]);**

**}**

**if(n>=5) // проверка, что 5 элемент есть**

**{**

**a[4]=kol3;**

**printf("kol3= %4d \n",kol3);**

**puts("New Massiv");**

**for(i=0;i<n;i++)**

**printf("%4d", a[i]);**

**printf("\n");**

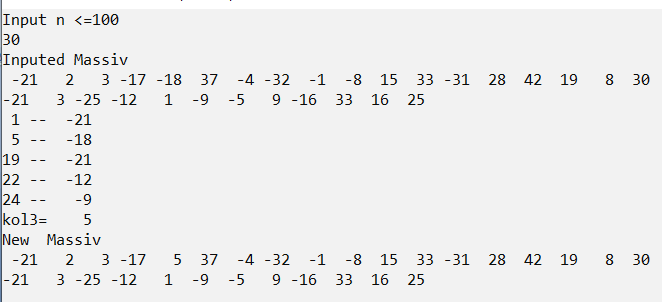
**}**

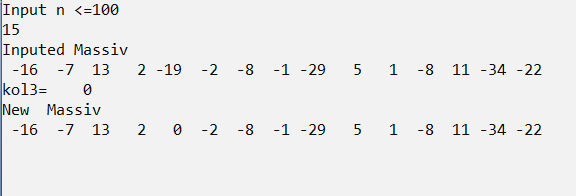
**else puts("Not Elem. N 5 ");**

**getch();**

**}**

**Примеры работы**

****

****

**Задача 4.**

Создать массив целых чисел. В нем найти максимальную сумму элементов из сумм 2 элементов и их индексы по схеме a[1]+a[n], a[2]+a[n-1],.... Вывести индексы элементов и сами элементы, составивщие максимальную сумму

**Анализ задания.**

Для решения задачи необходимо:

- сформировать массив или ввести с клавиатуры;

-вывести исходный массив;

- найти сумму элементов и их максимум по заданной схеме;

Как видно из анализа, для формирования и вывода массива можно использовать поэлементную обработку массива. А для поиска заданных сумм - одновременную обработку массивов или подмассивов с синхронным изменением индексов.

**Текст программы**

**#include<stdlib.h>**

**#include<conio.h>**

**#include<time.h>**

**#include <stdio.h>**

**int a[100];**

**int i,j,n,msum,imax;**

**int main()**

**{ srand((unsigned)time(NULL));**

**puts("Input n kratnoe 2<=100");**

**scanf("%d",&n);**

**for(i=0;i<n;i++)**

**a[i]=rand()%50;**

**puts("Inputed Massiv");**

**for(i=0;i<n;i++)**

**printf("%4d",a[i]);**

**printf("\n");**

**// поиск нужных элементов**

**msum=-32000;imax=-1;// первое приближение**

**for(i=0;i<(n/2);i++)**

**if(a[i]+a[n-i-1]>msum)**

**{ imax=i;**

**msum=(a[i]+a[n-i-1]);**

**}**

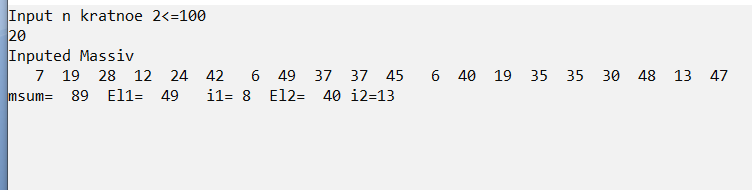
**printf("msum=%4d El1=%4d i1=%2d El2=%4d i2=%2d\n",msum,a[imax],imax+1,a[n-imax-1],n-imax);**

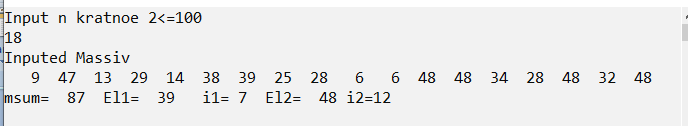
**printf("\n");**

**getch();**

**}**

**Примеры работы**

****

****

**Задачи для домашней работы**

**Задача 1.**

Создать масив вещественных чисел. В нем найти среднее арифметического элементов, стоящих на четных местах. Сформировать новый массив и переписать в него все элементы, заменив все отрицательные элементы найденным средним арифметическим. Вывести элементы, стоящие на четных мечтах и найденное среднее арифметическое, а также исходный и полученный массивы.

**Задача 2.**

Создать массив целых чисел не более 100 элементов. Определить сумму отрицательных элементов массива и количество положительных элементов. Вывести на печать исходный массив, найдженные отрицательные элементы, их сумму, положительные элементы, их крличество.

**Задача 2.**

Создать массив целых чисел, размером не более 100 элементов. Сформировать его вводом с клавиатуры или случайными числами. Заменить в массиве все отрицательные элементы их модулем. Вывести на экран исходный массив, все найденные отрицательные элементы, их индексы и полученный массив.