**Практическая работа №14**

 **Тема:** **Линейный поиск в массиве.**

 **Цель работы**: Приобретение практических навыков в работе с одномерными массивами в языке С++. Организация линейного поиска в массиве программными средствами.

**Краткие теоретические сведения**

**Линейный поиск в массиве**

**Задания 1,2,3 (на 3)**

**Задача №1**

|  |  |
| --- | --- |
| #include <conio.h> | // подключаем библиотеку |
| #include<iostream.h> | // консольного и |
|  using namespace std; | // потокового ввода-вывода |
| void main () | // точка начала программы |
| { |   |
| int a[5]={0,5,3,7,9}; | // объявляем и инициализируем |
|   | // целочисленный массив с именем а |
| clrscr(); | // очищаем экран |
| for (int i=1; i<5; i++) | // организуем n-1проход |
| { |   |
| for (int j=0; j<5; j++) |   |
| if (a[j]<a[j+1]) | // если порядок неправильный |
| { | // запоминаем a[j] в дополнительной |
| int temp=a[j]; | // переменной temp |
| a[j]=a[j+1]; | // меняем местами a[j] и a[j+1] |
| a[j+1]=temp; |   |
| } |   |
| } |   |
| for ( int i=0; i<5;i++) | // выводим на экран отсортированный |
| cout<<a[i]<<" "; | // массив |
| getch (); |   |
| } |   |

  **Как создать линейный поиск**

В примере ниже мы создали массив на 20 элементов и заполнили его ячейки случайными числами с помощью функции **rand.**

В строке 26 мы предлагаем пользователю с клавиатуры ввести ключ, а дальше мы проверим массив на наличие этого ключа. Если ключ совпал с какой-то ячейкой в массиве, то мы выведем индекс этой ячейки. Это типичный пример, чтобы понять как работает алгоритм.

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445 | #include <iostream>#include <cstdlib>#include <ctime>using namespace std; int main() {  setlocale(LC\_ALL, "rus");    int ans[20]; // создали массив для записи всех индексов  int h = 0;  int arr[20]; // создали массив на 20 элементов  int key; // создали переменную в которой будет находиться ключ   srand ( time(NULL) );   for (int i = 0; i < 20; i++) {    arr[i] = 1 + rand() % 20; // заполняем случайными числами ячейки     cout << arr[i] << " "; // выводим все ячейки массива     if (i == 9) {      cout << endl;    }  }   cout << endl << endl << "Введите ключ : "; cin >> key; // считываем ключ   for (int i = 0; i < 20; i++) {    if (arr[i] == key) { // проверяем равен ли arr[i] ключу      ans[h++] = i;    }  }   if (h != 0) { // проверяем были ли совпадения    for (int i = 0; i < h; i++) {      cout << "Ключ " << key << " находится в ячейке " << ans[i] << endl; //выводим все индексы    }  }  else {    cout << "Мы не нашли ключ " << key << " в массиве";  }    system("pause");  return 0;} |

Линейный поиск находится **в строках 28 — 32**.

**Задания для самостоятельной работы**

**3**.Создайте массив из 10 ячеек заполнив его случайными  числами  от 1 до 10 (включительно). Предложите пользователю с клавиатуры ввести ключ и выведите сколько раз ключ совпадал со значением ячеек в массиве, используя при подсчете линейный поиск, бинарный поиск.

**Упражнения 4и5 выполняются в тетради (на 4,5)**

**Упражнение 4**

Проанализируйте представленные фрагменты кода программы. Сформулируйте, какое действие выполняет каждый фрагмент. Предполагается, что в каждом из примеров массив инициализирован некоторым набором значений (на месте многоточия «…»).

a) **int a[10]={…}, s=0;**

б) int M[25]={…};

**for (int k=0; k<25; k++) if (M[k]==5)**

в) int i[3]={3,2,1};

for (int j=0; j<3; j+=2) cout<<i[j];

г) int S[3]={1}, f=1; for (int i=0; i<3; i++)

f+=S[i];

cout<<f;

cout<<c[0];

c[0]+=c[i];

for (int i=0; i<3 i++);

**Упражнение 5**

Определите, что будет выведено на экран в результате выполнения приведенных фрагментов кода.

Объясните работу этих конструкций пошагово.

a) int c[3]={1,2,3};

б) int c[3]={1,2,3};

for (int i=0; i<3; i++) c[i]+=c[2-i];

cout<< c[0]+c[1]+c[2];

в)float x[50]={…};

for (int n=0; n<50; n++)

if ((0<x[n] && (x[n]<1)) cout<<x[n];

г) float y[50]={…};

for (int m=0; m<49; m++) y[m]=y[m+1];

for (int i=0; i<10; i+=2) s+=a[i]

 **Вопросы и задания для отчета (на 5)**

1.Общий алгоритм работы линейного поиска по условию и его работа на примере.

2.Общий алгоритм поиска максимального и минимального элементов и его работа на примере.

3.Общий алгоритм сортировки методом «прямого выбора» и пошаговая реализация на примере.

4.Объясните, почему при поиске максимального и минимального элементов в примере переменным min и max первоначально присвоены значения А[0]? Можно ли для этой цели использовать другие элементы массива? Чем должен определяться выбор начальных значений min и max?