Варыянт 3

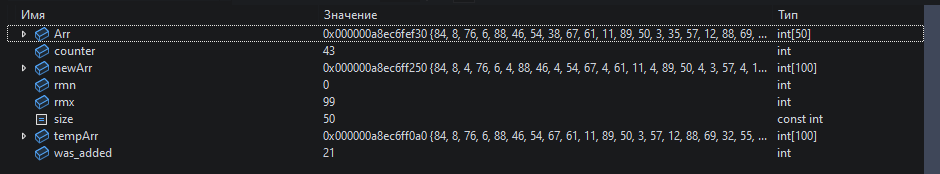
Заданне 5. В соответствии со своим вариантом написать программы по условиям, представленным в таблице ниже. Использо-вать одномерный массив целых случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры.

Представить результаты в окне Отладчика.

Для одной из программ составить блок-схему алгоритма.

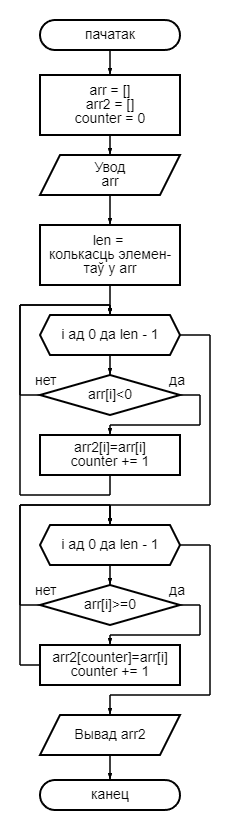
1. Удалить элементы, индексы которых кратны 7. Добавить после каждого нечетного элемента массива элемент со значением 4.

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <Windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int A[], int size) {  for (int i = 0; i < size; i++) {  cout << A[i] << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  using namespace std;  const int size = 50;  int Arr[size], rmn = 0, rmx = 99, counter = 1, was\_added = 0;  int tempArr[size \* 2], newArr[size \* 2];  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  Arr[i] = (int)(((double)rand() /  (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  cout << "Пачатковы масіў:" << '\n';  printArray(Arr, size);  // Выдаляем элементы, індэксы якіх кратныя 7 шляхам стварэння новага масіва і не дадання туды гэтых элементаў  tempArr[0] = Arr[0];  for (int i = 0; i < size; i++) {  if (not (i % 7 == 0)) {  tempArr[counter] = Arr[i];  counter++;  }  }  cout << "Масіў з выдаленымі элементамі, якія маюць кратны 7 індэкс:" << '\n';  printArray(tempArr, counter);  // Устаўляемы 4 пасля кожнага элемента, індэкс якога з'яўляецца няцотным лікам  newArr[0] = tempArr[0];  for (int i = 1; i < counter; i++) {  if (i % 2!= 0) {  newArr[i + was\_added] = tempArr[i];  newArr[i + was\_added + 1] = 4;  was\_added++;  }  else {  newArr[i + was\_added] = tempArr[i];  }  }  cout << "Масіў з устаўленнымі 4 пасля кожнага няцотнага індэкса:" << '\n';  printArray(newArr, counter + was\_added);  return 0;  } |
|  |



2. Задан массив из k чисел. Преобразовать массив следующим образом: все отрицательные элементы массива перенести в начало, а все остальные – в конец.

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <Windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int A[], int size) {  for (int i = 0; i < size; i++) {  cout << A[i] << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  using namespace std;  const int size = 20;  int Arr[size], rmn = -10, rmx = 10, counter = 0;  int newArr[size];  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  Arr[i] = (int)(((double)rand() /  (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  cout << "Пачатковы масіў:\n";  printArray(Arr, size);  //Дабаўляем усе адмоўные лікі па парадку ў новы масіў  for (int i = 0; i < size; i++) {  if (Arr[i] < 0) {  newArr[counter] = Arr[i];  counter++;  }  }  //Дабаўляем у новы масіў усе неадмоўныя лікі пасля адмоўных  for (int i = 0; i < size; i++) {  if (!(Arr[i] < 0)) {  newArr[counter] = Arr[i];  counter++;  }  }  cout << "Адсартаваны па ўмове масіў:\n";  printArray(newArr, size);  return 0;  } |
|  |
|  |



Дап. Заданні:

2.Подсчитать количество пар соседних элементов массива с одинаковыми значениями.

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <Windows.h>  using namespace std;  int main()  {  SetConsoleOutputCP(1251);  using namespace std;  const int size = 20;  int Arr[size], rmn = 0, rmx = 5, counter = 0;  int newArr[size];  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  Arr[i] = (int)(((double)rand() /  (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  cout << "Пачатковы масіў:\n";  printArray(Arr, size);  //Лічым колькасць пар аднолькавых лікаў  for (int i = 0; i < size-1; i++) {  if (Arr[i] == Arr[i+1]) {  counter++;  }  }  // Выводзім вынік падліку  cout << "\n"<<counter;    return 0;  } |
|  |

3. Найти в массиве наибольшее число подряд идущих одинаковых элементов.

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <Windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int A[], int size) {  for (int i = 0; i < size; i++) {  cout << A[i] << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  using namespace std;  const int size = 20;  int Arr[size], rmn = 0, rmx = 5, counter = 0, max\_count = 0;  int newArr[size];  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  Arr[i] = (int)(((double)rand() /  (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  cout << "Пачатковы масіў:\n";  printArray(Arr, size);  //Лічым колькасць аднолькавых лікаў  for (int i = 0; i < size - 1; i++) {  for (int j = i; j < size - 1; j++) {  // Пачынаючы з і элемента перабіраем усе лікі і параўноўваем з наступным  if (Arr[j] == Arr[j + 1]) {  counter++;  }  else { // Калі наступны лік не роўны папярэдняму, то захоўваем колькасць аднолькавых лікаў, што ідуць адзін за аднім. інакш не мяняем значэнне лічыльніка  max\_count = (counter > max\_count) ? counter+1 : max\_count;  counter = 0;  break;  }  }  }  // Выводзім вынік падліку  cout << "Колькасць: " << max\_count;  return 0;  } |
|  |

4. В массиве M, размером k, много совпадающих элементов. Найти количество различных элементов в нем (не упорядочивая массив).

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <Windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int A[], int size) {  for (int i = 0; i < size; i++) {  cout << A[i] << " ";  }  cout << '\n';  }  //Фунцыя для высвятлення ці есть дадзены лік у дадзеным масіве  bool isNumberInArray(int arr[], int size, int num) {  bool flag = false;  // Як толькі такі лік сустракаецца у масіве, то мяняем сцяг на true і спыняем цыкл  for (int i = 0; i < size; i++) {  if (arr[i] == num) {  flag = true;  break;  }  }  return flag;  }  int main()  {  SetConsoleOutputCP(1251);  const int k = 20;  int M[k], rmn = 0, rmx = 9, counter = 0;  int newArr[k];  // Ствараем масіў з k выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < k; i++) {  M[i] = (int)(((double)rand() /  (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  cout << "Пачатковы масіў:\n";  printArray(M, k);  //Напаўняем масів newArr лікамі, якіх яшчэ няма ў гэтым масіве  for (int i = 0; i < k; i++) {  if (!isNumberInArray(newArr, k, M[i])) {  newArr[counter] = M[i];  counter++;  }  }  // Выводзім вынік  cout << "Непаўторныя лікі:\n";  printArray(newArr, counter);  return 0;  } |
|  |