Варыянт 3

Заданне 5. В соответствии со своим вариантом написать программы для условий задач из таблицы, представленной ниже, с использованием ***указателей***для доступа к элементам массива. Проанализировать содержимое памяти компьютера.

|  |
| --- |
| 1. Заданы два массива **А** и **В**. Подсчитать в них количество элементов, меньших значения **t**, и первым на печать вывести массив, имеющий наибольшее их количество. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int size = 20;  int A[size], B[size], rmn = 0, rmx = 90, t = 5, count\_A = 0, count\_B = 0;  int\* ptrA = A, \* ptrB = B;  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  \*(ptrA + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  \*(ptrB + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  // Правяраем кожны элемент масіву на тое, ці меньшы ен за 5 і тады павялічваем адпаведны лічыльнік на 1  for (int i = 0; i < size; i++) {  count\_A += (\*(ptrA + i) < t) ? 1 : 0;  count\_B += (\*(ptrB + i) < t) ? 1 : 0;  }  // Спачатку выводзім пачатковые масівы, а потым спачатку той, у якога элементаў меншых за 5 больш. калі аднолькава, то спачатку А, потым Б  cout << "Пачатковы масіў А: "; printArray(size, ptrA);  cout << "Пачатковы масіў B: "; printArray(size, ptrB);  (count\_A >= count\_B) ? (printArray(size, ptrA), printArray(size, ptrB)) : (printArray(size, ptrB), printArray(size, ptrA));  return 0;  } |
|  |
| 1. Даны два массива **С** и **В**, каждый из **m** элементов. Подсчитать количество таких чисел **k**, для которых:   **C[k] > D[k]** и **C[k] < D[k], С[k] = D[k]**. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int size = 20;  int C[size], B[size], rmn = 0, rmx = 90, k = 0;  int\* ptrC = C, \* ptrB = B;  // Ствараем масіў з size выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  \*(ptrC + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  \*(ptrB + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  // Спачатку выводзім пачатковые масівы з дапамогай функцыі  cout << "масіў С: "; printArray(size, ptrC);  cout << "масіў D: "; printArray(size, ptrB);  // павялічваем лічыльнік на 1 калі умова выпаўняецца  for (int i = 0; i < size; i++) {  k += (\*(ptrC + i) > \*(ptrB + i) || \*(ptrC + i) < \*(ptrB + i)) ? 1 : 0;  }  //Вывоздзім вынік  cout << "Лікаў, задавальняючых умове C[k] > D[k] і C[k] < D[k]: " << k << endl;;  k = 0;  for (int i = 0; i < size; i++) {  k += (\*(ptrC + i) == \*(ptrB + i)) ? 1 : 0;  }  cout << "Лікаў, задавальняючых умове С[k] = D[k]: " << k;  return 0;  } |
|  |

Варыянт 4

|  |
| --- |
| 1. Дан массив **A** из **n** элементов и **B** из **m** элементов. Содержится ли наибольший элемент массива **A** в массиве **B**? |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int n = 20;  const int m = 30;  int A[n], B[m], rmn = 0, rmx = 90, k = 0, max\_in\_A;  int\* ptrA = A, \* ptrB = B;  // Ствараем масіў A з n выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++) {  \*(ptrA + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  // Ствараем масіў B з m выпадковых лікаў  for (int i = 0; i < m; i++) {  \*(ptrB + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  //Выводзім масівы  cout << "масіў A: "; printArray(n, ptrA);  cout << "масіў B: "; printArray(m, ptrB);  // Прызначаем першы элемент за максімальны, і параўноўваем яго з наступнымі для знаходжання максімальнага  max\_in\_A = \*(ptrA + 0);  for (int i = 1; i < n; i++) {  if (\*(ptrA + i) > max\_in\_A) {  max\_in\_A = \*(ptrA + i);  }  }  // правяраем ці ёсць у B максімальны элемент з А  for (int i = 0; i < m; i++) {  if (\*(ptrB + i) == max\_in\_A) {  //Калі есць, то выводзім "есць" і спыняем праграму  cout << "Ёсць";  return 0;  }  }  //Калі няма, то праграма дойдзе сюды і выведзе "няма"  cout << "Няма";  return 0;  } |
|  |
| 2. Найти количество различных чисел среди элементов целочисленного массива **Z**, содержащего **n** элементов. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int n = 10;  int A[n], B[n], rmn = 0, rmx = 10, k = 0, counter=0;  int\* ptrA = A, \*ptrB = B;  bool flag = false;  // Ствараем масіў A з n выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++) {  \*(ptrA + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  //Выводзім масіў A  cout << "масіў A: "; printArray(n, ptrA);  //Для кожнага ліку ў масіве А параўноваем яго з лікамі ў В: калі ў B такіх лікаў няма, то павялічваем лічыльнік на 1 і дабаўляем гэты лік у масів  for (int i = 0; i < n; i++) {  for (int j = 0; j <= k; j++) {  if (\*(ptrB + j) == \*(ptrA + i)) {  flag = true;  }  }  if (!flag) {  \*(ptrB+k++) = \*(ptrA + i);  counter++;  }  flag = false;  }  //Выводзім лічыльнік  cout << counter;  return 0;  } |
|  |

Варыянт 7

|  |
| --- |
| 1. Задан одномерный числовой массив **A** из **n** элементов и число **k**. Найти номера всех элементов массива, которые равны, больше и меньше **k**. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int size = 20;  int C[size], rmn = 0, rmx = 90, k;  int\* ptrC = C;  cin >> k; //Уводзім лік k, для якога будзем шукаць  srand((unsigned)time(NULL));  for (int i = 0; i < size; i++) {  \*(ptrC + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  // Спачатку выводзім пачатковые масівы з дапамогай функцыі  cout << "масіў С: "; printArray(size, ptrC);  //Выводзім нумары элементаў, якія менш або больш за k  cout << "Нумары элементаў, якія больш або менш за " << k << ": ";  for (int i = 0; i < size; i++) {  if (\*(ptrC + i) > k || \*(ptrC + i) < k) {  cout << i + 1 << " ";  }  }  //Выводзім нумары элементаў, якія роўныя k  cout << "\nНумары элементаў, якія роўныя " << k << ": ";  for (int i = 0; i < size; i++) {  if (\*(ptrC + i) == k) {  cout << i + 1 << " ";  }  }  return 0;  } |
|  |
| 2. В целом числе **N** выделить цифры, кратные числу **k**, и записать их в одномерный массив. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  int n, counter=0;  const int size = 10;  int C[10], k;  int\* ptr = C;  cin >> n; cin >> k; //Уводзім лік і лік, на кратнасць якому будзем правяраць усе яго лічбы  //Цыкл у 10 праходжанняў, бо int абмежаваны ад −2 147 483 648 да 2 147 483 647  for (int i = 0; i < 10; i++) {  if ((n % 10) % k == 0) { //Бярэм апошнюю лічбу і знаходзім ці есць астача ад дзялення на k і калі няма, то захоўваем гэтую лічбу ў масіў  \*(ptr + counter) = (n % 10);  counter++;  }  //Калі астача ад дзялення на 10 роўна 0, то гэта была самая першая лічба ліку(апошня пры ітэрацыі з канца)  if (n / 10 == 0) {  break;  }  //Адбрасваем апошнюю лічбу  n = n / 10;  }  printArray(counter, ptr);  return 0;  } |
|  |

Варыянт 9

|  |
| --- |
| 1. Даны **N** положительных целых чисел, которые не делятся ни на какие простые числа, кроме 2 и 3. Удалить из массива числа так, чтобы из любых двух оставшихся одно число делилось на другое. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int size = 5;  int C[size];  int\* ptrC = C;  cout << "Увядзіце элементы масіву: " << endl;  for (int i = 0; i < size; i++) {  cin >> \*(ptrC + i);  }  // Спачатку выводзім пачатковы масів з дапамогай функцыі  cout << "масіў: "; printArray(size, ptrC);  //Да кожнага элемента ад пачатку правяраем ці будуць яны дзяліца адзін на аднаго і выводзім гэтыя лікі калі так  for (int i = 0; i < size; i++) {  for (int j = 0; j < i; j++) {  if (\*(ptrC + i) % \*(ptrC + j) == 0 || \*(ptrC + j) % \*(ptrC + i) == 0) {  cout << \*(ptrC + i) << " " << \*(ptrC + j);  return 0;  }  }  }  //Калі няма такіх лікаў, то будзе выканан гэты код  cout << "Такіх лікаў няма";  return 0;  } |
|  |
| 2. Определить, содержится ли наименьший элемент массива **A** в массиве **B**. |
| #include <iostream>  #include <windows.h>  using namespace std;  // Функцыя для вываду масіва ў кансоль  void printArray(int size, int\* ptr) {  for (int i = 0; i < size; i++) {  cout << \*(ptr + i) << " ";  }  cout << '\n';  }  int main()  {  SetConsoleOutputCP(1251);  const int n = 20;  const int m = 30;  int A[n], B[m], rmn = 0, rmx = 90, k = 0, min\_in\_A;  int\* ptrA = A, \* ptrB = B;  // Ствараем масіў A з n выпадковых лікаў  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++) {  \*(ptrA + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  // Ствараем масіў B з m выпадковых лікаў  for (int i = 0; i < m; i++) {  \*(ptrB + i) = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);  }  //Выводзім масівы  cout << "масіў A: "; printArray(n, ptrA);  cout << "масіў B: "; printArray(m, ptrB);  // Прызначаем першы элемент за максімальны, і параўноўваем яго з наступнымі для знаходжання максімальнага  min\_in\_A = \*(ptrA + 0);  for (int i = 1; i < n; i++) {  if (\*(ptrA + i) < min\_in\_A) {  min\_in\_A = \*(ptrA + i);  }  }  // правяраем ці ёсць у B максімальны элемент з А  for (int i = 0; i < m; i++) {  if (\*(ptrB + i) == min\_in\_A) {  //Калі есць, то выводзім "есць" і спыняем праграму  cout << "Ёсць";  return 0;  }  }  //Калі няма, то праграма дойдзе сюды і выведзе "няма"  cout << "Няма";  return 0;  } |
|  |