Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 12

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условия задач** |
| --- | --- |
| 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.  2. Определить, содержится ли наименьший элемент массива **A** в массиве **B**. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>  bool IS\_POW(int base,int z, int lim, int el) {//с помощью данной функции определяем является ли число el степенью числа base  while (pow(base, z) < lim) {//проверяем степени числа base в диапазоне генерации  if (pow(base, z) == el) {//если число el совпадает с одной из степеней числа base ,то возвращаем значение true для нашей проверки  return true;  }  z++;//проходим каждую степень за один шаг цикла  }  return false;//если число el не совпадает ни с одной из степеней числа, то возвращаем значение true для нашей проверки  }  int NO\_POW(int x) {//с помощью данной функции мы избавляемся от всех степеней двойки из числа  while (x % 2 == 0 and x!=0) {//пока число будет кратно двум, мы каждый шаг убираем одну степень из числа  x /= 2;  }  return x;//возвращаем очищенную версию числа  }  bool DIVISION(int x, int y) {//данная определяет делится хотя бы одно из чисел на второе без остатка  if (x % y == 0 or y % x == 0) { return true; }  else { return false; }  }  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, x,lim\_start,lim\_finale, lim, sum = 0,z=0, \*p;  bool check = false;  const int max\_n = 2000;//ограничиваем максимальный размер  cout << "Введите размер массива: ";  cin >> n;  if (max\_n < n) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  cout <<endl<< "Введите диапазон генерации элементов массива:";  cin >> lim\_finale;  cout << endl;  int array[max\_n];//объявляем массив  srand((unsigned)time(NULL));//с помощью функции srand(seed of random) генерируем новые числа при каждом запуске, функция time(NULL),возвращает текущее время в секундах  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы массива, наш произвольный размер достигается за счёт того, что мы ведем наш цикл ввода и остальные лишь до указанного размера, тем самым игнорируя остальные элементы. Не задавая им значения, в функции int main они примут значение 0  {  \*(array + i) = rand() % lim\_finale;//с помощью указателя \*(array+i) обращаемся к памяти, хранящей значение array[i]  if (\*(array + i) % 2 == 0) {//eсли число кратно 2,то оно может быть степенью 2 или произведением степеней 2 и 3  if(IS\_POW(2,z,lim\_finale, \*(array + i))==true){//делаем проверку на то, является ли наш элемент степенью 2  cout << \*(array + i) << " ";//выводим подходящий символ на экран  check = true;  z = 0;  }  else {  x = \*(array + i);//используем дополнительную переменную, чтобы изначально сгенерированный элемент не изменялся в ходе проверки  if (IS\_POW(3, z, lim\_finale, NO\_POW(x) )== true) {  cout << \*(array + i) << " ";//выводим подходящий символ на экран  check = true;  z = 0;//обновляем счетчик степеней  }  }  }    if (\*(array + i) % 3 == 0) {//eсли число кратно 3,то оно может быть степенью 3  if (IS\_POW(3, z, lim\_finale, \*(array + i)) == true) {//делаем проверку на то, является ли наш элемент степенью 2  cout << \*(array + i) << " ";//выводим подходящий символ на экран  check = true;  }  z = 0;  }  if (check == false) {//при отсутствии генерации символа, подходящего по условиям, делаем шаг в цикле назад, чтобы сгенерировать новое число  i--;  }  check = false;//обновляем показатель вывода подходящих элементов  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (!DIVISION(\*(array + i), \*(array + j))) {//проверяем каждый элемент массива на делимость между собой, при не прохождении проверки удаляем  n--;//при удалении лишнего элемента уменьшаем размер самого массива на 1  for (int k = i; k < n; k++) { \*(array + k) = \*(array + k + 1); }//перемещаем каждый элемент начиная с индекса удаленного на 1 ячейку влево  }  }  }  cout << endl << endl;  for (int i = 0; i < n; i++)  cout << \*(array + i) << " ";//выводим массив при помози указателей  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, k, min\_el = 100, \* min\_point=&min\_el;//создаем указатель \*min\_pointer и с помощью & получаем адрес переменной min\_el  const int max\_n = 2000;  cout << "Введите размер первого массива: ";  cin >> n;  cout << "Введите размер второго массива: ";  cin >> k;  srand((unsigned)time(NULL));  if (max\_n < n or max\_n < k) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int arr\_a[max\_n], arr\_b[max\_n];//объявляем массивы  cout << endl << "Массив A" << endl << n << endl;  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы первого массива  {  \*(arr\_a + i) = rand() % 25;//генерируем по одному элементу первого массива за шаг в диапозоне до 25 чтобы увеличить шансы совпадения  cout << \*(arr\_a + i) << " ";//выводим элементы массива для возможности их анализа  \*min\_point = min(\*(arr\_a + i), \*min\_point);//находим минимальный элемент первого массива  //при использовании в названии переменной указателя мы связываемся напрямую с значеним в ячейке памяти,на которую указывает этот указатель  }  cout << endl << endl << "Минимальный элемент массива A: " << \*min\_point << endl << endl;  cout << "Массив B" << endl << k << endl;  for (int i = 0; i < k; i++)//с помощью цикла for вводим элементы второго массива  {  \*(arr\_b + i) = rand() % 25;//генерируем по одному элементу второго массива за шаг в диапозоне до 25  cout << \*(arr\_b + i) << " ";//выводим элементы массива для возможности их анализа  }  for (int i = 0; i < k; i++)  {  if (\*min\_point == \*(arr\_b + i)) { cout <<endl<< "Есть"; return 0; }//если находим совпадение минимального первого элемента с одним из элементов второго, выводим ответ и завершаем программу  }  cout<<endl << "Нет";//если программа не завершилась к этому моменту, то совпадений нет, что мы и выводим  } |
| **Результат программы** |
| **Память, хранящая значение переменной min\_el** |

Дополнительные варианты

| **№ варианта** | **Условия задач** |
| --- | --- |
| 1 | 1. Ввести целое число N. Выделить из этого числа цифры, кратные m, и записать их в одномерный массив.  2. Заданы два массива по 10 целых чисел в каждом. Найти наибольшее среди чисел первого массива, которое не входит во второй массив. |

|  |
| --- |
| **Код программы** |
| #include <iostream>/  using namespace std;  void main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int c=0, k, m, array[5];//размер 5,т.к. максимальный int содержит 5 цифр  cout << "Введите K:";  cin >> k;  cout << endl << "Введите m:";  cin >> m;  while (k != 0) {//читаем каждый разряд числа K с конца  if ((k % 10) % m == 0 and k%10!=0) { \*(array + c) =k%10; c++; }//если текущий разряд кратен, то записываем его в массив и увеличиваем индекс для следующей записи игнорируем ноль, т.к. он не кратен никакому числу  k /= 10;//переходим к следующему разряду  }  for (int i = c-1; i >= 0; i--)//т.к. читали число и записывали цифры с конца, то и массив будем выводить с конца  cout << \*(array + i) << " ";//выводим массив при помощи указателей  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n=10, k=10, max\_el = 0, \* max\_point = &max\_el;//создаем указатель \*min\_pointer и с помощью & получаем адрес переменной min\_el  bool check=true;  srand((unsigned)time(NULL));  int arr\_a[10], arr\_b[10];//объявляем массивы  cout << "Массив A" << endl << n << endl;  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы первого массива  {  \*(arr\_a + i) = rand() % 25;//генерируем по одному элементу первого массива за шаг в диапазоне до 25 чтобы увеличить шансы совпадения  cout << \*(arr\_a + i) << " ";//выводим элементы массива для возможности их анализа  }  cout << endl << "Массив B" << endl << k << endl;  for (int i = 0; i < k; i++)//с помощью цикла for вводим элементы второго массива  {  \*(arr\_b + i) = rand() % 25;//генерируем по одному элементу второго массива за шаг в диапозоне до 25  cout << \*(arr\_b + i) << " ";//выводим элементы массива для возможности их анализа  }  for (int i = 0; i < n; i++)  {  for (int j = 0; j < k; j++) {  if (\*(arr\_a + i) == \*(arr\_b + j)) {//проверяем есть ли данный элемент первого массива во втором  check = false;  }  }  if(check==true){//если его там нет, то мы смотрим является ли он минимальным  \*max\_point = max(\*(arr\_a + i), \*max\_point);  }  check = true;  }  cout << endl << endl << "Максимальный элемент массива A: " << \*max\_point << endl << endl;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| 4 | 1. Дан массив **A** из **n** элементов и **B** из **m** элементов. Содержится ли наибольший элемент массива **A** в массиве **B**?  2. Найти количество различных чисел среди элементов целочисленного массива **Z**, содержащего **n** элементов. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, k, max\_el = 0, \* max\_point = &max\_el;//создаем указатель \*max\_point и с помощью & получаем адрес переменной max\_e  const int max\_n = 2000;  cout << "Введите размер первого массива: ";  cin >> n;  cout << "Введите размер второго массива: ";  cin >> k;  srand((unsigned)time(NULL));  if (max\_n < n or max\_n < k) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int arr\_a[max\_n], arr\_b[max\_n];//объявляем массивы  cout << endl << "Массив A" << endl << n << endl;  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы первого массива  {  \*(arr\_a + i) = rand() % 25;//генерируем по одному элементу первого массива за шаг в диапазоне до 25 чтобы увеличить шансы совпадения  cout << \*(arr\_a + i) << " ";//выводим элементы массива для возможности их анализа  \*max\_point = max(\*(arr\_a + i), \*max\_point);//находим максимальный элемент первого массива  //при использовании в названии переменной указателя мы связываемся напрямую с значением в ячейке памяти, на которую указывает этот указатель  }  cout << endl << endl << "Максимальный элемент массива A: " << \*max\_point << endl << endl;  cout << "Массив B" << endl << k << endl;  for (int i = 0; i < k; i++)//с помощью цикла for вводим элементы второго массива  {  \*(arr\_b + i) = rand() % 25;//генерируем по одному элементу второго массива за шаг в диапазоне до 25  cout << \*(arr\_b + i) << " ";//выводим элементы массива для возможности их анализа  }  for (int i = 0; i < k; i++)  {  if (\*max\_point == \*(arr\_b + i)) { cout << endl << "Есть"; return 0; }//если находим совпадение максимального первого элемента с одним из элементов второго, выводим ответ и завершаем программу  }  cout << endl << "Нет";//если программа не завершилась к этому моменту, то совпадений нет, что мы и выводим  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, x, counter = 0, counter\_min = 0;  bool check = true;//объявляем переменную буллевого типа для проверки на уникальность  const int max\_n = 2000;//ограничиваем максимальный размер  cout << "Введите размер массива: ";  cin >> n;  if (max\_n < n) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int array[max\_n], uniq[max\_n];//т.к. в условиях нет ограничения на использования доп. массивов,то воспользуемся им для хранения уникальных элементов  srand((unsigned)time(NULL));  for (int i = 0; i < n; i++)  {  \*(array + i) = rand() % 100;  cout << array[i] << " ";//выводим элементы массива для возможности их анализа  for (int j = 0; j < i; j++) {//сравниваем текущий элемент массива с известными уникальными элементами  if ( \*(array+i) == \*(uniq + j)) { check = false; }//если текущий элемент уже встречался,то наш элемент не прошел проверку на уникальность  }  if (check == true) {//если элемент проходит проверку, то мы записываем его в наш банк уникальных элементов  \*(uniq + counter) = \*(array + i);  counter++;//чтобы уникальные элементы не записывались в одну ячейку, мы используем переменную counter для продвижения по уникальному массиву  }  check = true;//обновляем переменную проверки для следующего элемента  }  cout << endl << "Уникальный массив:";  for (int i = 0; i < counter; i++)//выводим уникальный массив для возможности анализа этого массива и работоспособности программы  {  cout << \*(uniq + i) << " ";  }  cout << endl << "Кол-во уникальных элеметов:" << counter; //выводим ответ  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **15** | 1. Дан массив **А** размера **n**, не содержащий нулевых элементов. Преобразовать массив **А** так, чтобы вначале шли положительные элементы, а затем отрицательные. Дополнительные массивы не использовать.  2. Определить, содержится ли наибольший элемент массива **F** в массиве **D**? |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, x,lim\_finale, lim\_start , count\_pos=0,z=0;  bool check = false;  const int max\_n = 2000;//ограничиваем максимальный размер  cout << "Введите размер массива: ";  cin >> n;  if (max\_n < n) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  cout << endl << "Введите начало диапазона генерации элементов массива:";  cin >> lim\_start;  cout <<endl<< "Введите предел диапазона генерации элементов массива:";  cin >> lim\_finale;  cout << endl;  int array[max\_n];//объявляем массив  srand((unsigned)time(NULL));//с помощью функции srand(seed of random) генерируем новые числа при каждом запуске, функция time(NULL),возвращает текущее время в секундах  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы массива, наш произвольный размер достигается за счёт того, что мы ведем наш цикл ввода и остальные лишь до указанного размера, тем самым игнорируя остальные элементы. Не задавая им значения, в функции int main они примут значение 0  {  \*(array + i) = (rand() % (lim\_finale + abs(lim\_start))) - abs(lim\_start);//с помощью указателя \*(array+i) обращаемся к памяти, хранящей значение array[i]  if (\*(array + i) == 0) { i--; }//eсли число равно 0,то генерируем заново  else{ cout << \*(array + i) << " "; }  if (\*(array + i) > 0) { count\_pos++; }//считаем количество положительных чисел  }  for (int i = 0; i < count\_pos; i++) {//смотрим все числа пока не закончатся положительные числа,ведь дальше будут все перемещенные отрицательные  if (\*(array + i ) < 0) {//если число отрицательное, то мы переносим его за предел массива, чтобы не терять последний элемент  \*(array + n) = \*(array + i);  for (int j = i; j < n; j++) {//используем цикл для перемещения всех элементов влево на 1 начиная с индекса отрицательного элемента  \*(array + j) = \*(array + j + 1);  }  i--;//делаем шаг назад, чтобы при постановке двух отрицательных подряд, мы перенесли и следующее отрицательное число  }  }  cout << endl << endl;  for (int i = 0; i < n; i++)  cout << \*(array + i) << " ";//выводим массив при помощи указателей  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  #include <locale>//директива, необходимая для функции time  using namespace std;  int main()  {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  int n, k, max\_el = 0, \* max\_point = &max\_el;//создаем указатель \*max\_pointer и с помощью & получаем адрес переменной max\_el  const int max\_n = 2000;  cout << "Введите размер первого массива: ";  cin >> n;  cout << "Введите размер второго массива: ";  cin >> k;  srand((unsigned)time(NULL));  if (max\_n < n or max\_n < k) {//проверка на введённый пользователем размер  cout << "Превышение максимального размера массива!";  return 0;  }  int arr\_a[max\_n], arr\_b[max\_n];//объявляем массивы  cout << endl << "Массив F" << endl << n << endl;  for (int i = 0; i < n; i++)//с помощью цикла for вводим элементы первого массива  {  \*(arr\_a + i) = rand() % 25;//генерируем по одному элементу первого массива за шаг в диапазоне до 25 чтобы увеличить шансы совпадения  cout << \*(arr\_a + i) << " ";//выводим элементы массива для возможности их анализа  \*max\_point = max(\*(arr\_a + i), \*max\_point);//находим максимальный элемент первого массива  //при использовании в названии переменной указателя мы связываемся напрямую с значением в ячейке памяти, на которую указывает этот указатель  }  cout << endl << endl << "Максимальный элемент массива F: " << \*max\_point << endl << endl;  cout << "Массив D" << endl << k << endl;  for (int i = 0; i < k; i++)//с помощью цикла for вводим элементы второго массива  {  \*(arr\_b + i) = rand() % 25;//генерируем по одному элементу второго массива за шаг в диапазоне до 25  cout << \*(arr\_b + i) << " ";//выводим элементы массива для возможности их анализа  }  for (int i = 0; i < k; i++)  {  if (\*max\_point == \*(arr\_b + i)) { cout << endl << "Есть"; return 0; }//если находим совпадение максимального первого элемента с одним из элементов второго, выводим ответ и завершаем программу  }  cout << endl << "Нет";//если программа не завершилась к этому моменту, то совпадений нет, что мы и выводим  } |
| **Результат программы** |
|  |