Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

|  |
| --- |
| Институт информационных технологий и анализа данных |

наименование института

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

по дисциплине:

|  |
| --- |
| Программирование |
| «Модульное программирование» |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил | АСУб-20-2 |  |  |  | Арбакова А.В. |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил |  |  |  |  | Столбов А.Б. |
|  | должность |  | подпись |  | Фамилия И.О. |

Иркутск 2021 г.

**1.Условие задачи № 5:**

Даны векторы A[10], B[10], C[10], каждый элемент которых имеет значение 1 или 0. Значение каждой компоненты вектора, имеющего максимальное число компонент со значением 0 (в предположении, что такой вектор только 1), заменить логической суммой значений соответствующих компонент двух других векторов.

**2.Математическая модель**

1. Введём в массивы числа (0 или 1) вручную или рандомно.
2. Подсчет количества нулей в каждом массиве.
3. Выявление максимального количества среди нулей всех массивов.
4. Применим замену (логическую сумму от двух других массивов) к массиву с максимальным количеством нулей.
5. Вывод результатов.

**3.Таблица 1 – Внешние спецификации**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | ОДЗ |
| 1 | menu | Пункт меню | int | [0,1;2] |
| 2 | kola, kolb, kolc | Количество нулей в каждом из векторов. | int | [1;10] |
| 3 | i | Вспомогательная | int | [1;10] |
| 4 | A[], B[], C[] | Массивы. | int | [10] |
| 5 | koll | Вспомогательная переменная, для передачи данных между функциями. | int | [1;10] |
| 6 | M[], N[], P[] | Вспомогательные массивы, для передачи данных между функциями. | int | [10] |
| 7 | n | Вспомогательная переменная для ввода массивов. | int | [0; 1] |
| 8 | C1 | Сообщение | Текст | 1 вид |
| 9 | C2 | Сообщение | Текст | 1 вид |
| 10 | C3 | Сообщение | Текст | 1 вид |
| 11 | C4 | Сообщение | Текст | 3 вида |
| 12 | C5 | Сообщение | Текст | 3 вида |
| 13 | C6 | Сообщение | Текст | 3 вида |
| 14 | C7 | Сообщение | Текст | 1 вид |
| 15 | С8 | Сообщение | Текст | 1 вид |
| 16 | С9 | Сообщение | Текст | 1 вид |
| 17 | С10 | Сообщение | Текст | 3 вида |
| 18 | С11 | Сообщение | Текст | 3 вида |

Таблица сообщений

C1: «Даны векторы A[10], B[10], C[10], каждый элемент которых имеет значение 1 или 0. Значение каждой компоненты вектора, имеющего максимальное число компонент со значением 0 (в предположении, что такой вектор только 1), заменить логической суммой значений соответствующих компонент двух других векторов.»

C2: «Выберите способ:

1. Ручной ввод.

2. Рандомный ввод.

0. Выход из программы.»

C3: «Максимальное количество нулей должно быть единственным, иначе программа выдаст ввод значений заново.»

C4.1: «Введите вектор A:»

C4.2: «Введите вектор B:»

C4.3: «Введите вектор C:»

C5.1: «Вектор A:»

C5.2: «Вектор B:»

C5.3: «Вектор C:»

C6.1: «Количество нулей в векторе А:»

C6.2: «Количество нулей в векторе B:»

C6.3: «Количество нулей в векторе C:»

C7: «Конечные векторы:»

С8: «Ошибка! Введите число.»

С9: «Введите число еще раз (1 или 0)»

С10.1: «Произведена замена вектора A на логическую сумму векторов B и C:»

С10.2: «Произведена замена вектора B на логическую сумму векторов A и C:»

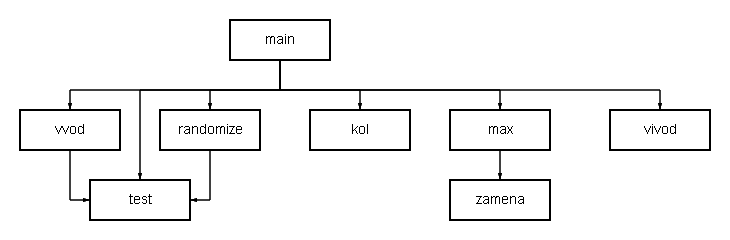
С10.3: «Произведена замена вектора C на логическую сумму векторов A и B:»

С11.1: «A:»

С11.2: «B:»

С11.3: «C:»

**4.Иерархия модулей**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип результата | Параметры |
| 1 | test | Проверяет состояние входного потока. | false/true | - |
| 2 | main | Главная функция. | - | int menu – переменная для меню. |
| 3 | vvod | Ввод массивов в ручную. | int | int n – переменная для ввода.  int M[] – массив для ввода. |
| 4 | randomize | Ввод массивов через рандом. | int | int M[] – массив для ввода. |
| 5 | kol | Подсчет нулей в переданном массиве. | int | int M[] – заменяющий массив.  int koll – заменяющая переменная для счета количества нулей. |
| 6 | max | Выявление максимального количества нулей в каком-либо массиве. | int | int kola – количество нулей в массиве.  int kolb – количество нулей в массиве.  int kolc – количество нулей в массиве. |
| 7 | zamena | Применение замены (логической суммы) для массива. | int | int M[] – заменяющий массив.  int N[] – заменяющий массив.  int P[] – заменяющий массив. |
| 8 | vivod | Вывод результатов. | int | int A[] – массив.  int B[] – массив.  int C[] – массив. |

Глобальные переменные:

int i - счетчик

int kola, int kolb, int kolc, int koll – счетчик количества нулей.

int A[10], int B[10], int C[10] – основные массивы.

int M[10] int N[10], int P[10] – заменяющие массивы.

**5.Алгоритмизация**

Алгоритм главной программы main():

1. объявление переменной menu
2. подключение русского языка на вывод
3. цикл menu
   1. вывод С1
   2. вывод С2
   3. с проверкой на тест от букв и знаков вводится переменная menu, равная 1, 2 или 0
   4. сравнить menu
      1. если menu = 1
         1. вывод С3
         2. вывод С4.1
         3. ввод массива А через функцию vvod(A)
         4. вывод С5.1
         5. цикл для вывод на экран массива А
         6. вывод С4.2
         7. ввод массива B через функцию vvod(B)
         8. вывод С5.2
         9. цикл для вывод на экран массива B
         10. вывод С4.3
         11. ввод массива C через функцию vvod(C)
         12. вывод С5.3
         13. цикл для вывод на экран массива C
         14. вызов функции kol(A, kola)
         15. вызов функции kol(B, kolb)
         16. вызов функции kol(C, kolc)
         17. вывод C6.1, kola, C6.2, kolb, C6.3, kolc на экран
         18. цикл, исключающий максимального равного количества нулей в массивах
             1. вызов функции max(kola, kolb, kolc)
             2. вывод С7
         19. вызов функции вывода на экран vivod(A, B, C)
         20. выход в меню
      2. если menu = 1
         1. вывод С3
         2. вывод С5.1
         3. рандомный ввод randomize(A)
         4. цикл для вывод на экран массива А
         5. вывод С5.2
         6. рандомный ввод randomize(B)
         7. цикл для вывод на экран массива B
         8. вывод С5.3
         9. рандомный ввод randomize(C)
         10. цикл для вывод на экран массива C
         11. вызов функции kol(A, kola)
         12. вызов функции kol(B, kolb)
         13. вызов функции kol(C, kolc)
         14. вывод C6.1, kola, C6.2, kolb, C6.3, kolc на экран
         15. цикл, исключающий максимального равного количества нулей в массивах
             1. вызов функции max(kola, kolb, kolc)
             2. вывод С7
         16. вызов функции вывода на экран vivod(A, B, C)
         17. выход в меню
      3. если menu = 0
         1. возврат 0
         2. выход в меню
   5. возврат 0

Алгоритм функции test():

1. Если состояние потока – неудача
   1. очистка ввода
   2. игнорирование новой строки
   3. возврат false
2. иначе true

Алгоритм функции vvod(int M[]):

1. n=0 и i=0
2. цикл пока i<10
   1. ввод элементов массива с проверкой на тест и на условие - только значения равные 0 или 1, иначе C9

Алгоритм функции randomize(int M[])

1. цикл, заполняющий рандомными 0 и 1 массив

Алгоритм kol(int M[], int& koll)

1. koll=0
2. цикл, проверяющий элементы массива на равность 0, при нахождении такого элемента счетчик k увеличивается на 1

Алгоритм функции max(int kola, int kolb, int kolc)

1. если kola – максимальное среди всех, т.е. больше kolb и kolc
   1. вызов функции zamena(A,B,C)
   2. вывод С10.1
   3. вывод замененного массива на экран
2. если kolb – максимальное среди всех, т.е. больше kola и kolc
   1. вызов функции zamena(B,A,C)
   2. вывод С10.2
   3. вывод замененного массива на экран
3. если kolc – максимальное среди всех, т.е. больше kola и kolb
   1. вызов функции zamena(C,A,B)
   2. вывод С10.3
   3. вывод замененного массива на экран

Алгоритм функции zamena(int M[], int N[], int P[])

1. цикл, выполняющий замену элементов данного массива на логическую сумму элементов двух других массивов.

Алгоритм функции vivod(int A[], int B[], int C[])

1. вывод С11.1
2. цикл, выводящий массив A
3. вывод С11.2
4. цикл, выводящий массив B
5. вывод С11.3
6. цикл, выводящий массив C

**6.Кодирование алгоритма или запись алгоритма на языке C++**

#include <iostream>

using namespace std;

int i;

int kola, kolb, kolc, koll;

int A[10], B[10], C[10], M[10];

int N[10], P[10];

int test()

{

if (cin.fail() != 0)

{

cout << "Ошибка! Введите число.\n";

cin.clear();

cin.ignore(332767, '\n');

return false;

}

return true;

}

void vvod(int M[])

{

int n = 0, i = 0;

while (i < 10)

{

cout << "[" << i + 1 << "]= ";

do

cin >> n;

while (test() != true);

if ((n == 0) || (n == 1))

{

M[i] = n;

i++;

}

else

cout << endl << "Введите число еще раз (1 или 0)" << endl << endl;

}

}

void kol(int M[], int& koll)

{

koll = 0;

for (i = 0; i < 10; i++)

{

if (M[i] == 0)

koll++;

}

}

void zamena(int M[], int N[], int P[])

{

for (i = 0; i < 10; i++)

M[i] = N[i] | P[i];

}

void max(int kola, int kolb, int kolc)

{

if ((kola > kolb) && (kola > kolc))

{

zamena(A, B, C);

cout << endl << "Произведена замена вектора A на логическую сумму векторов B и C: ";

for (i = 0; i < 10; i++)

cout << A[i] << " ";

}

if ((kolb > kola) && (kolb > kolc))

{

zamena(B, A, C);

cout << endl << "Произведена замена вектора B на логическую сумму векторов A и C: ";

for (i = 0; i < 10; i++)

cout << B[i] << " ";

}

if ((kolc > kolb) && (kolc > kola))

{

zamena(C, A, B);

cout << endl << "Произведена замена вектора C на логическую сумму векторов A и B: ";

for (i = 0; i < 10; i++)

cout << C[i] << " ";

}

}

void vivod(int A[], int B[], int C[])

{

cout << endl;

cout << "A: ";

for (i = 0; i < 10; i++)

cout << A[i] << " ";

cout << endl;

cout << "B: ";

for (i = 0; i < 10; i++)

cout << B[i] << " ";

cout << endl;

cout << "C: ";

for (i = 0; i < 10; i++)

cout << C[i] << " ";

cout << endl << endl;

}

void randomize(int M[])

{

for (i = 0; i < 10; i++)

M[i] = rand() % 2;

}

int main()

{

int menu = 1;

setlocale(LC\_ALL, "rus");

while (menu)

{

cout << "Даны векторы A[10], B[10], C[10], каждый элемент которых имеет значение 1 или 0. Значение каждой компоненты вектора, имеющего максимальное число компонент со значением 0 (в предположении, что такой вектор только 1), заменить логической суммой значений соответствующих компонент двух других векторов." << endl;

cout << "\nВыберите способ:\n1. Ручной ввод. \n2. Рандомный ввод. \n0. Выход из программы." << endl;

do

cin >> menu;

while (test() != true);

switch (menu)

{

case 1:

do

{

cout << endl << "Максимальное количество нулей должно быть единственнным, иначе программа выдаст ввод значений заново." << endl;

cout << endl << "Введите вектор A:" << endl;

vvod(A);

cout << endl << "Вектор A: ";

for (i = 0; i < 10; i++)

cout << A[i] << " ";

cout << endl << endl << "Введите вектор B:" << endl;

vvod(B);

cout << endl << "Вектор B: ";

for (i = 0; i < 10; i++)

cout << B[i] << " ";

cout << endl << endl << "Введите вектор C:" << endl;

vvod(C);

cout << endl << "Вектор C: ";

for (i = 0; i < 10; i++)

cout << C[i] << " ";

kol(A, kola);

kol(B, kolb);

kol(C, kolc);

cout << endl << endl;

cout << endl << "Количество нулей в векторе А: " << kola;

cout << endl << "Количество нулей в векторе B: " << kolb;

cout << endl << "Количество нулей в векторе C: " << kolc;

cout << endl;

} while (((kola == kolb) && (kola > kolc)) || ((kola == kolc) && (kola > kolb)) || ((kolb == kolc) && (kolb > kola)) || ((kola == kolb) && (kola == kolc)));

max(kola, kolb, kolc);

cout << endl << endl << "Конечные векторы: ";

vivod(A, B, C);

break;

case 2:

do

{

cout << endl << "Максимальное количество нулей должно быть единственным, иначе программа задаст значения заново." << endl;

cout << endl << "Вектор A: ";

randomize(A);

for (i = 0; i < 10; i++)

cout << A[i] << " ";

cout << endl << "Вектор B: ";

randomize(B);

for (i = 0; i < 10; i++)

cout << B[i] << " ";

cout << endl << "Вектор C: ";

randomize(C);

for (i = 0; i < 10; i++)

cout << C[i] << " ";

kol(A, kola);

kol(B, kolb);

kol(C, kolc);

cout << endl << endl;

cout << endl << "Количество нулей в векторе А: " << kola;

cout << endl << "Количество нулей в векторе B: " << kolb;

cout << endl << "Количество нулей в векторе C: " << kolc;

cout << endl;

} while (((kola == kolb) && (kola > kolc)) || ((kola == kolc) && (kola > kolb)) || ((kolb == kolc) && (kolb > kola)) || ((kola == kolb) && (kola == kolc)));

max(kola, kolb, kolc);

cout << endl << endl << "Конечные векторы: ";

vivod(A, B, C);

break;

case 0:

return 0;

break;

}

} return 0;

}

**7.Таблица тестов**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № теста | Исходные данные | | | | | | Результат | Примечания |
| Ручной ввод | | | Рандомный ввод | | |
| A[] | B[] | C[] | A[] | B[] | C[] |
| 1 | 1  0  0  0  0  0  0  0  1  1 | 0  0  0  0  1  1  1  1  1  0 | 0  1  1  1  1  0  0  1  0  1 |  |  |  | Вектор A: 1 0 0 0 0 0 0 0 1 1  Вектор B: 0 0 0 0 1 1 1 1 1 0  Вектор C: 0 1 1 1 1 0 0 1 0 1  Количество нулей в векторе А: 7  Количество нулей в векторе B: 5  Количество нулей в векторе C: 4  Произведена замена вектора A на логическую сумму векторов B и C: 0 1 1 1 1 1 1 1 1 1  Конечные векторы:  A: 0 1 1 1 1 1 1 1 1 1  B: 0 0 0 0 1 1 1 1 1 0  C: 0 1 1 1 1 0 0 1 0 1 | - |
| 2 | 1  0  0  0  1  1  1  0  0  1  1  0  0  0  1  1  1  0  0  1 | 1  0  0  0  0  0  0  1  1  0  1  1  0  0  0  0  0  1  1  0 | 0  0  0  1  0  0  1  1  0  0  0  0  0  1  0  0  1  1  0  0 |  |  |  | Вектор A: 1 0 0 0 1 1 1 0 0 1  Вектор B: 1 0 0 0 0 0 0 1 1 0  Вектор C: 0 0 0 1 0 0 1 1 0 0  Количество нулей в векторе А: 5  Количество нулей в векторе B: 7  Количество нулей в векторе C: 7  Максимальное количество нулей должно быть единственным, иначе программа выдаст ввод значений заново.  Вектор A: 1 0 0 0 1 1 1 0 0 1 Вектор B: 1 1 0 0 0 0 0 1 1 0  Вектор C: 0 0 0 1 0 0 1 1 0 0  Произведена замена вектора C на логическую сумму векторов A и B: 1 1 0 0 1 1 1 1 1 1  Конечные векторы:  A: 1 0 0 0 1 1 1 0 0 1  B: 1 1 0 0 0 0 0 1 1 0  C: 1 1 0 0 1 1 1 1 1 1 | Одинаковое максимальное количество нулей в двух массивах. |
| 3 |  |  |  | 1  1  0  0  1  0  0  0  0  0 | 1  1  0  0  1  0  0  0  0  0 | 1  1  0  0  1  0  0  0  0  0 | Вектор A: 1 1 0 0 1 0 0 0 0 0  Вектор B: 1 1 0 0 1 0 0 0 0 0  Вектор C: 1 1 0 0 1 0 0 0 0 0  Количество нулей в векторе А: 7  Количество нулей в векторе B: 2  Количество нулей в векторе C: 6  Произведена замена вектора A на логическую сумму векторов B и C: 1 1 1 1 1 1 1 0 1 1  Конечные векторы:  A: 1 1 1 1 1 1 1 0 1 1  B: 1 1 1 1 1 1 1 0 1 0  C: 1 0 0 1 0 0 1 0 0 1 | - |
| 4 |  |  |  | 0  1  0  1  1  0  0  0  1  1  0  0  1  0  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  1  0  0  0  0  1  0  0  1  0  1  1  0  1  1  0  0  0  0  1  1 | 1  1  1  0  0  0  1  0  1  0  0  0  1  0  0  0  1  1  1  1  0  1  0  1  0  1  1  1  0  0  0  1  0  1  0  1  1  1  0  0  0  1  0  1  1  1  0  1  1  0 | 1  1  0  0  0  1  1  1  1  0  1  1  1  1  1  1  1  0  1  0  1  0  0  0  0  1  0  1  0  0  1  0  0  0  0  1  0  1  0  0  1  0  1  1  0  1  1  0  0  1 | Вектор A: 0 1 0 1 1 0 0 0 1 1  Вектор B: 1 1 1 0 0 0 1 0 1 0  Вектор C: 1 1 0 0 0 1 1 1 1 0  Количество нулей в векторе А: 5  Количество нулей в векторе B: 5  Количество нулей в векторе C: 4  Максимальное количество нулей должно быть единственным, иначе программа задаст значения заново.  Вектор A: 0 0 1 0 1 1 1 0 1 0  Вектор B: 0 0 1 0 0 0 1 1 1 1  Вектор C: 1 1 1 1 1 1 1 0 1 0  Количество нулей в векторе А: 5  Количество нулей в векторе B: 5  Количество нулей в векторе C: 2  Максимальное количество нулей должно быть единственным, иначе программа задаст значения заново.  Вектор A: 0 0 0 0 1 0 0 1 0 1  Вектор B: 0 1 0 1 0 1 1 1 0 0  Вектор C: 1 0 0 0 0 1 0 1 0 0  Количество нулей в векторе А: 7  Количество нулей в векторе B: 5  Количество нулей в векторе C: 7  Максимальное количество нулей должно быть единственным, иначе программа задаст значения заново.  Вектор A: 1 0 1 1 0 0 0 0 1 1  Вектор B: 0 1 0 1 1 1 0 1 1 0  Вектор C: 1 0 1 1 0 1 1 0 0 1  Количество нулей в векторе А: 5  Количество нулей в векторе B: 4  Количество нулей в векторе C: 4  Произведена замена вектора A на логическую сумму векторов B и C: 1 1 1 1 1 1 1 1 1 1  Конечные векторы:  A: 1 1 1 1 1 1 1 1 1 1  B: 0 1 0 1 1 1 0 1 1 0  C: 1 0 1 1 0 1 1 0 0 1 | Сгенерировано 3 раза с выпадением одинакового максимального количества нулей в двух массивах. |
| 5 | 1  0  0  0  0  0  0  0  0  0  1  0  0  0  1  1  1  0  1  0 | 0  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0  1  0  1  0 | 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0 |  |  |  | Вектор A: 1 0 0 0 0 0 0 0 0 0  Вектор B: 0 0 0 1 0 0 0 0 0 0  Вектор C: 0 0 0 0 0 0 0 1 0 0  Количество нулей в векторе А: 9  Количество нулей в векторе B: 9  Количество нулей в векторе C: 9  Максимальное количество нулей должно быть единственным, иначе программа выдаст ввод значений заново.  Вектор A: 1 0 0 0 1 1 1 0 1 0  Вектор B: 0 0 1 1 1 0 1 0 1 0  Вектор C: 0 0 0 0 0 0 1 1 1 0  Количество нулей в векторе А: 5  Количество нулей в векторе B: 5  Количество нулей в векторе C: 7  Произведена замена вектора C на логическую сумму векторов A и B: 1 0 1 1 1 1 1 0 1 0  Конечные векторы:  A: 1 0 0 0 1 1 1 0 1 0  B: 0 0 1 1 1 0 1 0 1 0  C: 1 0 1 1 1 1 1 0 1 0 | Одинаковое максимальное количество нулей в трёх массивах. |