**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Дисциплина: Теория цифровых автоматов**

**Тема: Синтез и анализ комбинационных схем с одним**

**выходом с учетом неопределенностей**

Выполнил: ст. группы ВТ-31

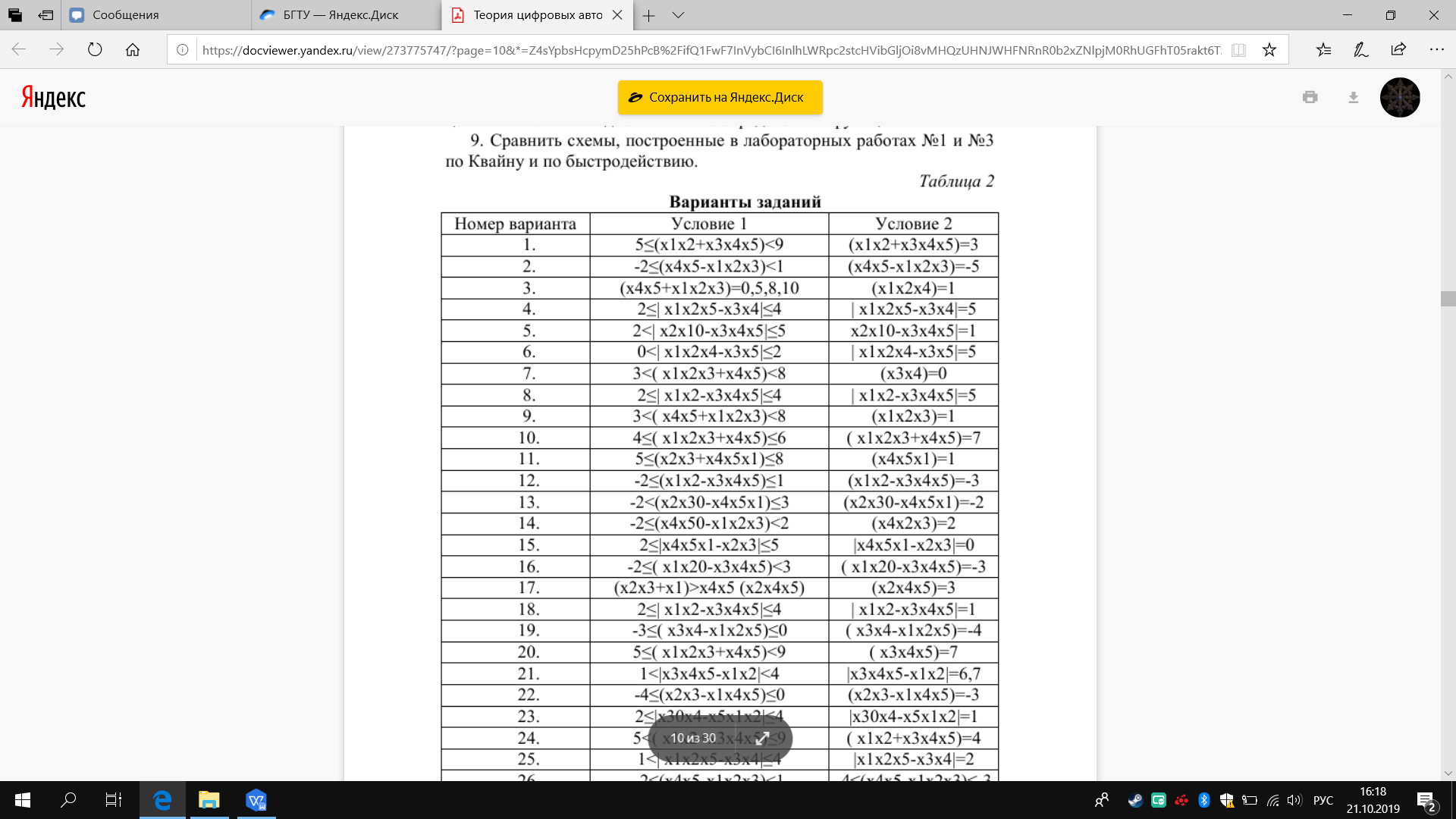
Подкопаев Антон Валерьевич

Проверил: Рязанов Юрий Дмитриевич

**Белгород 2019**

**Цель работы:** научиться строить эффективные по быстродействию и затратам оборудования комбинационные схемы с учетом неопределенностей.

**Вариант 13**

****

**Ход выполнения работы**

1. Составить таблицу истинности заданной частично определенной булевой функции. Булева функция здесь задана двумя условиями, зависящими от значений аргументов. Если на наборе аргументов условие 2 истинно, то значение функции на этом наборе не определено. Если же на наборе аргументов условие 2 ложно, то значение функции на этом наборе равно значению условия 1 на этом наборе аргументов. В условии значение аргумента отождествляется с двоичной цифрой, а последовательность аргументов — с двоичным числом.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 00(0)-000(0)=0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 00(0)-010(2)=-2 | - |
| 0 | 0 | 0 | 1 | 0 | 00(0)-100(4)=-4 | 0 |
| 0 | 0 | 0 | 1 | 1 | 00(0)-110(6)=-6 | 0 |
| 0 | 0 | 1 | 0 | 0 | 01(1)-000(0)=1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 01(1)-010(2)=-1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 01(1)-100(4)=-3 | 0 |
| 0 | 0 | 1 | 1 | 1 | 01(1)-110(6)=-5 | 0 |
| 0 | 1 | 0 | 0 | 0 | 10(2)-000(0)=2 | 1 |
| 0 | 1 | 0 | 0 | 1 | 10(2)-010(2)=0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 10(2)-100(4)=-2 | - |
| 0 | 1 | 0 | 1 | 1 | 10(2)-110(6)=-4 | 0 |
| 0 | 1 | 1 | 0 | 0 | 11(3)-000(0)=3 | 1 |
| 0 | 1 | 1 | 0 | 1 | 11(3)-010(2)=1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 11(3)-100(4)=-1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 11(3)-110(6)=-3 | 0 |
| 1 | 0 | 0 | 0 | 0 | 00(0)-001(1)=-1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 00(0)-011(3)=-3 | 0 |
| 1 | 0 | 0 | 1 | 0 | 00(0)-101(5)=-5 | 0 |
| 1 | 0 | 0 | 1 | 1 | 00(0)-111(7)=-7 | 0 |
| 1 | 0 | 1 | 0 | 0 | 01(1)-001(1)=0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 01(1)-011(3)=-2 | - |
| 1 | 0 | 1 | 1 | 0 | 01(1)-101(5)=-4 | 0 |
| 1 | 0 | 1 | 1 | 1 | 01(1)-111(7)=-6 | 0 |
|  |  |  |  |  |  |  |
| 1 | 1 | 0 | 0 | 0 | 10(2)-001(1)=1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 10(2)-011(3)=-1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 10(2)-101(5)=-3 | 0 |
| 1 | 1 | 0 | 1 | 1 | 10(2)-111(7)=-5 | 0 |
| 1 | 1 | 1 | 0 | 0 | 11(3)-001(1)=2 | 1 |
| 1 | 1 | 1 | 0 | 1 | 11(3)-011(3)=0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 11(3)-101(5)=-2 | - |
| 1 | 1 | 1 | 1 | 1 | 11(3)-111(7)=-4 | 0 |

1. Решить задачу минимизации частично определенной булевой функции в классе дизъюнктивных нормальных форм.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 00000 + | 00100 +  01000 +  10000 +  00001 + | 00101 +  01001 +  01100 +  11000 +  10100 +  01010 + | 01101 +  01110 +  11001 +  11100 +  10101 + | 11101 +  11110 + |  |
| 00-00 +  0-000 +  -0000 +  0000- + | 0010- +  00-01 +  0100- +  0-001 +  0-100 +  01-00 +  -1000 +  1-000 +  -0100 +  10-00 +  010-0 + | 0-101 +  01-01 +  0110- +  011-0 +  01-10 +  -1001 +   1. +   -1100 +  11-00 +  1-100 +  -0101 +   1. + | -1101 +  11-01 +  1110- +  1-101 +  -1110 +  111-0 + |  |  |
| 00-0- +   1. -00 +   0-00- +  --000 +  -0-00 + | 1. -01 +   0-10- +  01-0- +  **01--0 <**  -100- +  -1-00 +   1. -00 +   --100 +  -010- + | -1-01 +  -110- +  --101 +  11-0- +  1-10- +  **-11-0 <** |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 1. **-0- <**   **---00 <** | **-1-0- <**  **--10- <** |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-11-0** |  |  |  |  |  |  | + |  |
| **01--0** |  |  | + |  |  |  | + |  |
| **-1-0-** |  |  | + |  |  | + | + | + |
| **--10-** |  | + |  |  | + |  | + |  |
| **0--0-** | + | + | + |  | + | + | + |  |
| **---00** | + | + | + | + |  |  | + | + |
|  | 00000 | 00100 | 01000 | 10000 | 00101 | 01001 | 01100 | 11000 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | + |  | + |  | **-11-0** |
|  |  | + |  |  |  | **01--0** |
|  | + |  | + | + | + | **-1-0-** |
| + | + |  |  | + | + | **--10-** |
|  | + |  |  |  |  | **0--0-** |
| + |  |  |  | + |  | **---00** |
| 10100 | 01101 | 01110 | 11001 | 11100 | 11101 |  |

В ядро Квайна входят все простые импликанты. Получаем минимальную ДНФ:



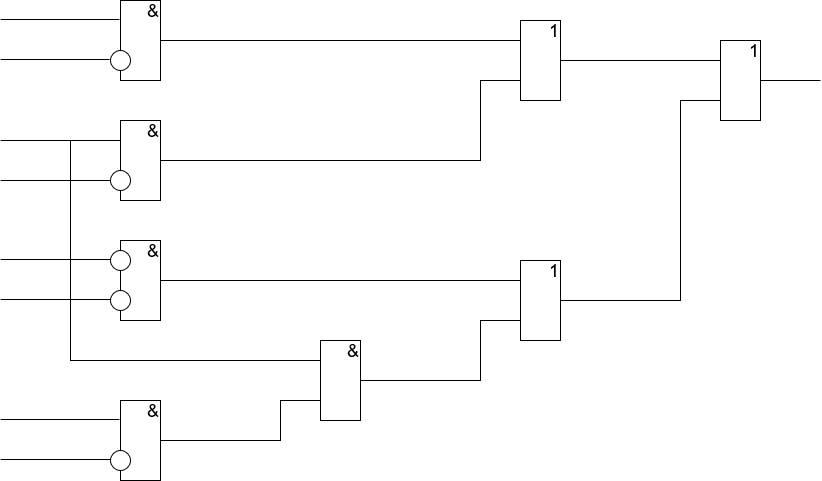
1. Написать программу, строящую таблицу истинности булевой функции, полученной при выполнении второго пункта. Сравнить полученную таблицу с таблицей истинности исходной частично определенной булевой функции.

см. *Приложение*

1. Применить факторизационный метод синтеза многоярусной комбинационной схемы в базисе И-ИЛИ-НЕ с двухвходовыми элементами И и ИЛИ по полученной при выполнении второго пункта минимальной дизъюнктивной нормальной форме булевой функции

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | **1** |
|  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
|  | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | + | + | + | + | **1** | **1** |
|  | 0 | **1** | **1** | 0 | 0 | 0 |
|  | **1** | 0 | 0 | **1** | 0 | 0 |



1. задачу минимизации частично определенной булевой функции в классе конъюнктивных нормальных форм.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
|  | 00010+ | 00011 +  00110 +  10001 +  10010 + | 00111 +  01011 +  10011 +  10110 +  11010 + | 01111 +  10111 +  11011 + | 11111 + |
|  | 0001- +  00-10 +  -0010 + | 00-11 +  0011- +  0-011 +  -0011 +  **100-1 <**  1001- +  -0110 +  10-10 +  1-010 + | 0-111 +  01-11 +  -0111 +  10-11 +  1011- +  -1011 +  1-011 +  1101- + | -1111 +  1-111 +  11-11 + |  |
|  | 00-1- +  -001- +  -0-10 + | 0--11 +  -0-11 +  -011- +  10-1- +  --011 +  **1-01- <** | --111 +  -1-11 +   1. -11 + |  |  |
|  | **-0-1- <** | **---11 <** |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1-01-** |  |  |  |  |  |  |
| **-0-1-** | + | + | + | + |  |  |
| **---11** |  | + |  | + | + | + |
| **100-1** |  |  |  |  |  |  |
|  | 00010 | 00011 | 00110 | 00111 | 01011 | 01111 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | + | + |  |  | + | + |  | **1-01-** |
|  | + | + | + |  |  |  |  | **-0-1-** |
|  |  | + |  | + |  | + | + | **---11** |
| + |  | + |  |  |  |  |  | **100-1** |
| 10001 | 10010 | 10011 | 10110 | 10111 | 11010 | 11011 | 11111 |  |

В ядро Квайна входят все простые импликанты. Получаем минимальную KНФ:



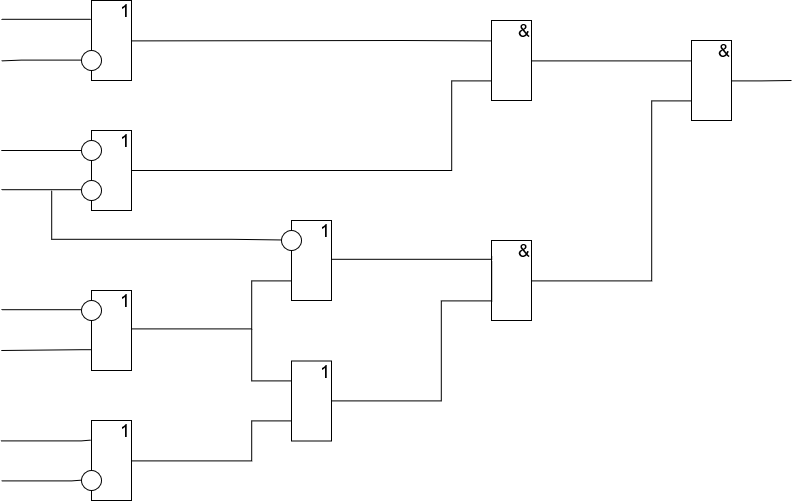
1. Написать программу, строящую таблицу истинности булевой функции, полученной при выполнении пятого пункта. Сравнить полученную таблицу с таблицей истинности исходной частично определенной булевой функции.

см. *Приложение*

1. Применить факторизационный метод синтеза многоярусной комбинационной схемы в базисе И-ИЛИ-НЕ с двухвходовыми элементами И и ИЛИ по полученной при выполнении пятого пункта минимальной конъюнктивной нормальной форме булевой функции.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | **1** | 0 |
|  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|  | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | **1** | **1** |
|  | 0 | **1** | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | + | + | + | + | 1 | 1 |
|  | 0 | 1 | 1 | 0 | 0 | 0 |
|  | 1 | 0 | 0 | 1 | 0 | 0 |



1. Написать программы, моделирующие работу схем, полученных в пунктах 4 и 7, на всех входных наборах и строящие таблицу истинности каждой схемы. Сравнить полученные таблицы истинности с таблицей истинности исходной частично определенной функции.

см. *Приложение*

1. Сравнить схемы, построенные в лабораторных работах №1 и №3 по Квайну и по быстродействию.

*Приложение*

#include <iostream>

#include <vector>

using namespace std;

bool NextSet(int\* a, int n, int m) {

int j = m - 1;

while (j >= 0 && a[j] == n) j--;

if (j < 0) return false;

if (a[j] >= n)

j--;

a[j]++;

if (j == m - 1) return true;

for (int k = j + 1; k < m; k++)

a[k] = 0;

return true;

}

void Print(int\* a, int n) {

static int num = 1;

cout.width(3);

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

cout << a[i] << " ";

}

bool gun(int\* a, int n) {

Print(a, n);

bool x1 = a[0], x2 = a[1], x3 = a[2], x4 = a[3], x5 = a[4];

bool z1, z2, u1, u2, u3, u4, u5, u6, v1, v2, v3, v4, f;

bool not\_x1, not\_x2, not\_x3, not\_x4, not\_x5, not\_z1, not\_z2, not\_u1, not\_u2, not\_v1, not\_v2, not\_v3, not\_v4, not\_f;

// исходная булева функция

f = !x1 and x2 and x3 and !x5 or !x1 and x3 and !x4 or x2 and !x4 or !x4 and !x5;

printf(f ? " 1 " : " 0 ");

// минимальная ДНФ по неполной булевой функции

f = x2 and x3 and !x5 or !x1 and x2 and !x5 or x2 and !x4 or x3 and !x4 or !x1 and !x4 or !x4 and !x5;

printf(f ? " 1 " : " 0 ");

// минимальная КНФ по неполной булевой функции

f = (!x1 or x3 or !x4) and (x2 or !x4) and (!x4 or !x5) and (!x1 or x2 or x3 or !x5);

printf(f ? " 1 " : " 0 ");

// неполная ДНФ

u2 = x2 and !x4;

u3 = x3 and !x4;

u4 = !x4 and !x5;

z1 = x2 and !x5;

u1 = x3 and z1;

v1 = u2 or u3;

v2 = u1 or u4;

f = v1 or v2;

printf(f ? " 1 " : " 0 ");

// неполная КНФ

u2 = x2 or !x4;

u3 = !x4 or !x5;

z1 = !x1 or x3;

z2 = x2 or !x5;

u1 = !x4 or z1;

u4 = z1 or z2;

v1 = u2 and u3;

v2 = u1 and u4;

f = v1 and v2;

printf(f ? " 1 " : " 0 ");

printf("\n");

return (f);

}

int main() {

vector<vector<int>\*>\* list = new vector<vector<int>\*>();

int\* tmp;

int n = 1, m = 5, \* a = new int[m];

for (int i = 0; i < m; i++) a[i] = 0;

int i = 0;

gun(a, m);

while (NextSet(a, n, m)) {

gun(a, m);

}

return 0;

}

