|  |  |
| --- | --- |
|  | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Сидоров Михаил Михайлович

Группа РК6-56Б

Тип задания Лабораторная работа

Тема лабораторной работы Бинарные сочетания

Вариант 20

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сидоров М.М.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2024 г.*

**Оглавление**

[Задание на лабораторную работу](#_gjdgxs) 2

[Идея решения](#_3znysh7) 2

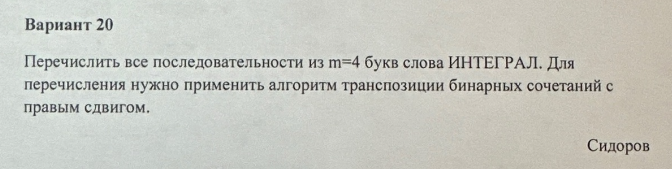
[Алгоритм](#_aplwgwmhwdq3) 2

[Листинг программы](#_5sw567tz43a8) 4

[Результаты работы программы.](#_v48ozo3o72q6) 6

[Источники](#_77pwe7h8fld2) 7

# Задание на лабораторную работу



# Идея решения

Основная идея заключается в том, что каждое сочетание можно представить в виде двоичного числа, где каждый бит соответствует одной из букв слова. Если бит установлен в 1, то буква выбирается в текущее сочетание, если 0 — буква не выбирается. Алгоритм сдвигает биты вправо, начиная с минимального сочетания, и систематически проходит через все возможные варианты.

# Алгоритм

1. **Инициализация начальной комбинации:**

Начальная комбинация comb представляет минимальное сочетание — первые m (в нашем случае 4) бита установлены в 1. Верхний предел limit равен 1 << n, где n — общее количество букв (8).

1. **Генерация следующей комбинации:**

Находим наименьший установленный бит в текущей комбинации, используя выражение u = comb & -comb.

Пример:

Пусть текущая комбинация comb = 15 (00001111), что соответствует выбору букв И, Н, Т, Е.

Вычисляем наименьший установленный бит u = comb & -comb = 00001111 & 11110001 = 00000001. Здесь u — это 1, что соответствует младшему установленному биту.

-comb инвертирует все биты comb и прибавляет 1 (арифметическое отрицание). Побитовое И (&) между comb и -comb выделяет самый младший установленный бит.

1. **Вычисляем новое значение** v = comb + u**.**

Пример:

Теперь comb = 15, а u = 1. Добавим u к comb: v = comb + u = 15 + 1 = 16 (00010000).

1. **Обновляем комбинацию:**

Обновляем значение comb, используя формулу: comb = (((comb ^ v) >> 2) / u) | v.

Пример:

comb ^ v вычисляет побитовое исключающее ИЛИ: comb ^ v = 15 ^ 16 = 31 (00011111).

Сдвигаем результат на 2 бита вправо: (comb ^ v) >> 2 = 31 >> 2 = 7 (00000111).

Делим результат на u: 7 / 1 = 7.

Объединяем результат с v: comb = 7 | 16 = 23 (00010111), что соответствует новому сочетанию букв И, Н, Т, Г.

Таким образом, мы перешли от комбинации И Н Т Е к комбинации И Н Т Г.

1. **Повторение шагов до тех пор, пока не будут перебраны все возможные комбинации.**

# Листинг программы

#include <iostream>

#include <bitset>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

const char letters[] = { 'И', 'Н', 'Т', 'Е', 'Г', 'Р', 'А', 'Л' };

int n = 8;

int m = 4;

int counter = 0;

int comb = (1 << m) - 1;

int limit = 1 << n;

while (comb < limit)

{

cout << "{ ";

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

if (comb & (1 << i)) {

cout << letters[i] << " ";

}

}

bitset<8> x(comb);

cout << "} " << x << " ";

counter++;

cout << counter;

cout << endl;

int u = comb & -comb;

int v = comb + u;

if (u == 0) break;

comb = (((comb ^ v) >> 2) / u) | v;

}

return 0;

}

# Результаты работы программы.

Вывод первых 5 сочетаний из 70:

{ И Н Т Е }

{ И Н Т Г }

{ И Н Е Г }

{ И Т Е Г }

{ Н Т Е Г }

...

# Источники

http://bigor.bmstu.ru/?cnt/?doc=Comby/bin\_comb\_enum.mod/?bck=Comby/comtz.the#T102764529