

Министерство образования и науки Российской Федерации  
Московский Государственный Технический Университет (МГТУ)  
имени Н. Э. Баумана  
Факультет «Робототехника и комплексная автоматизация (РК)»  
Кафедра «Системы автоматизированного проектирования (РК6)»

## **Отчет по лабораторной работе №1**

По курсу «Методы комбинаторных вычислений»

Выполнил:

Студент    Гусаров Аркадий  
                  Андреевич  
Группа        РК6-53Б

Проверил:

Дата            \_\_\_\_\_  
Подпись        \_\_\_\_\_

Москва, 2021 г.

## Вариант 6С

Перечислить все последовательности из  $m = 4$  букв слова ИНТЕГРАЛ. Для перечисления нужно применить алгоритм транспозиции бинарных сочетаний с правым сдвигом.

### Теория

В алгоритме транспозиции с правым сдвигом на каждом шаге очередное бинарное сочетание получается из текущего заменой крайней правой пары разрядов с 01 на 10 (транспозиция), а затем производится сдвиг всех единиц справа от нее в младшие разряды. Сдвиг производится только при наличии единиц, которые могут быть смещены вправо.

$$\begin{array}{ccccccc} \text{Транспозиция} & & \text{Транспозиция} & & \text{Сдвиг} & & \\ \dots (0011\underline{101})_3 - T' \rightarrow (00\underline{1110})_4 - T'' \rightarrow (0101\underline{110}) \rightarrow S'' \rightarrow (010011)_5 \dots & & & & & & \\ & \times & \times & & \times \times \times & & \\ & \text{Итерация 3} & & & \text{Итерация 4} & & \end{array}$$

Следует отметить, что итерации обоих алгоритмов можно записать в аддитивной форме, если интерпретировать бинарные сочетания как целые числа, записанные в системе счисления по основанию 2. В частности, для алгоритма транспозиции с правым сдвигом каждое очередное бинарное сочетание  $B'$  можно всегда получить из текущего сочетания  $B$ , выполнив операции сложения целых чисел по следующей аддитивной формуле:

$$B' = B + 2^N + 2^{E-1} - 1$$

В этой аддитивной формуле показатели степеней двоек  $N$  и  $E$  обозначают, соответственно, число нулевых младших разрядов текущего бинарного сочетания и количество единиц, стоящих подряд слева от них.

### Текст программы

```
#include <iostream>
#include <cmath>
using namespace std;

unsigned char word[] = "INTEGRAL";
// 128 в двоичной форме - 10000000
unsigned char lead = 128;

void printSequence(unsigned char B)
{
    static int inRow = 0;

    if (inRow == 4)
    {
        cout << endl;
        inRow = 0;
    }

    cout << '{';
```

```

for (int i = 0; i < 8; i++)
{
    char digit = (B >> i) & 1;
    if (digit)
    {
        cout << word[i];
    }
}

cout << " ";
inRow++;
}

unsigned char nextSequence(unsigned char B, int n)
{
    int E = 0, N = 0, counter = n;
    unsigned char copyB = B;

    // Считаем E в бинарном числе B, где E - кол-во лидирующих единиц
    while ((copyB & lead) && counter)
    {
        E++;
        counter--;
        copyB = copyB << 1;
    }

    // Считаем N в бинарном числе B, где N - кол-во нулей после лидирующих единиц
    while (!(copyB & lead) && counter)
    {
        N++;
        counter--;
        copyB = copyB << 1;
    }

    int shift_one = pow(2, n - E);
    int shift_two = pow(2, n - N + 1) - pow(2, n - N - E + 1);
    int shift_tre = pow(2, n - N - E - 1);

    return B + shift_one + shift_two + shift_tre;
}

int main()
{
    // Кол-во символов в слове INTEGRAL
    int n = 8;
    // 15 в двоичном виде - 00001111
    unsigned char B = 15, copyB = 15;

    do
    {
        printSequence(copyB);
        copyB = nextSequence(copyB, n);
    } while ((B & copyB));

    printSequence(copyB);

    cout << endl;

    return 0;
}

```

## Результат работы программы

{INTE} {INTG} {INTR} {INTA}  
{INTL} {INEG} {INER} {INEA}  
{INEL} {INGR} {INGA} {INGL}  
{INRA} {INRL} {INAL} {ITEG}  
{ITER} {ITEA} {ITEL} {ITGR}  
{ITGA} {ITGL} {ITRA} {ITRL}  
{ITAL} {IEGR} {IEGA} {IEGL}  
{IERA} {IERL} {IEAL} {IGRA}  
{IGRL} {IGAL} {IRAL} {NTEG}  
{NTER} {NTEA} {NTEL} {NTGR}  
{NTGA} {NTGL} {NTRA} {NTRL}  
{NTAL} {NEGR} {NEGA} {NEGL}  
{NERA} {NERL} {NEAL} {NGRA}  
{NGRL} {NGAL} {NRAL} {TEGR}  
{TEGA} {TEGL} {TERA} {TERL}  
{TEAL} {TGRA} {TGRL} {TGAL}  
{TRAL} {EGRA} {EGRL} {EGAL}  
{ERAL} {GRAL}