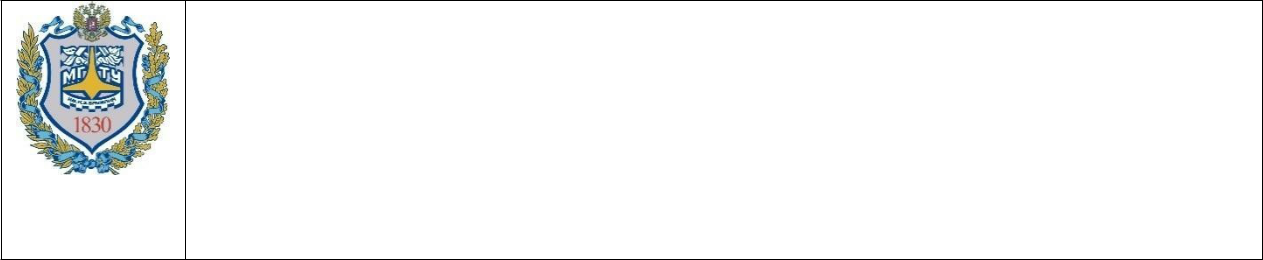
**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана**

**(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 | Егоров Алексей |
| Группа | РК6-56Б |
|  |  |
| Тип задания | Лабораторная работа №1 |
|  |  |

|  |  |
| --- | --- |
| Студент \_\_\_\_\_\_\_\_\_\_ | **Егоров А. С.** |
| *подпись, дата* | *фамилия, и.о.* |
|  |  |
| Преподаватель \_\_\_\_\_\_\_\_\_\_ |  |
| *подпись, дата* | *фамилия, и.о.* |

*Москва, 2021*

1. **Техническое задание**

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

* 1. **Описание алгоритма**

Алгоритм на каждом шаге получает очередное бинарное сочетание из текущего заменой крайней правой пары разрядов 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.

При аппаратной реализации сочетаний или при программировании на языке ассемблера важно иметь возможность обработки записей сочетаний в двоичном формате. В этом случае любое сочетание из **n** элементов по **m** элементов следует задавать в форме **n**-разрядного двоичного числа (Bn,…Bj,…B1), где **m** единичных разрядов обозначают элементы сочетания, а остальные **(n-m)** разрядов имеют нулевые значения. Очевидно, что при таком формате записи различные сочетания должны отличаться расположением единичных разрядов и существует всего **C(n,m)** способов расположить **m** единиц или **(n-m)** нулей в n-разрядном двоичном наборе. Например, в следующей таблице перечислены все 6 таких *бинарных сочетаний*, которые обеспечивают запись **4**-х разрядными двоичными числами всех сочетаний из **n=4** элементов произвольного множества **{E1, E2, E3, E4}** по **m=2** элемента:

**Таблица 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (0011)1 | (0101)2 | (0110)3 | (1001)4 | (1010)5 | (1100)6 |
| {E1, E2} | {E1, E3} | {E2, E3} | {E1, E4} | {E2, E4} | {E3, E4} |

* 1. **Описание входных данных**

На вход программа получает строку с единственным аргументом: названием исполняемого файла.

**IV. Описание выходных данных**

На выходе программа выдаёт строки перечисления и выводит общее количество выведенных комбинаций.

1. **Текст программы**
2. #include <iostream>
3. #include <cmath>
5. int transposition(int &seq, int count) {
6. **int loser = 1;**
8. bool isPrevOne;
9. bool isZero;
11. **for (int i = 0; i < count - 1; ++i) {**
12. isPrevOne = seq & loser;
13. isZero = ~seq & (loser << 1);
15. if (isPrevOne && isZero) {
16. **// swap bits**
17. seq -= pow(2, i) - pow(2, i + 1);
18. return i;
19. }
20. loser = loser << 1;
21. **}**
22. return -1;
23. }
25. int shift(int seq, int count) {
26. **int res = seq;**
27. for (int i = 0; i < count; ++i) {
28. if (~(seq >> i) & 1) {
29. res -= pow(2, count - i - 1) - pow(2, i);
30. } else {
31. **return res;**
32. }
33. }
35. return res;
36. **}**
38. void printSequence(int sequence, int count, std::string word) {
39. std::cout << '{';
40. for(int i = 0; i < count; ++i) {
41. **bool digit = (sequence >> i) & 1;**
42. if (digit) {
43. std::cout << word[i];
44. }
45. }
46. **std::cout << "} ";**
47. }
49. unsigned char nextSequence(int seq, int count) {
50. int index = transposition(seq, count);
51. **seq = shift(seq, index);**
52. return seq;
53. }
55. int main() {
56. **std::string word = "INTEGRAL";**
57. int cells = 4;
59. int count = word.size();
61. **int sequence = pow(2, cells) - 1;**
62. int curr\_sequence = sequence;
63. sequence = sequence << cells;
65. int i = 0;
66. **do {**
67. ++i;
68. printSequence(curr\_sequence, count, word);
69. curr\_sequence = nextSequence(curr\_sequence, count);
70. if (i % 5 == 0) {
71. **std::cout << std::endl;**
72. }
73. } while(sequence != curr\_sequence);
74. printSequence(curr\_sequence, count, word);
76. **std::cout << "\nCombinations displayed:" << ++i << std::endl;**
77. return 0;
78. }
79. **Выходные данные**

{INTE} {INTG} {INEG} {ITEG} {NTEG}

{INTR} {INER} {ITER} {NTER} {INGR}

{ITGR} {NTGR} {IEGR} {NEGR} {TEGR}

{INTA} {INEA} {ITEA} {NTEA} {INGA}

{ITGA} {NTGA} {IEGA} {NEGA} {TEGA}

{INRA} {ITRA} {NTRA} {IERA} {NERA}

{TERA} {IGRA} {NGRA} {TGRA} {EGRA}

{INTL} {INEL} {ITEL} {NTEL} {INGL}

{ITGL} {NTGL} {IEGL} {NEGL} {TEGL}

{INRL} {ITRL} {NTRL} {IERL} {NERL}

{TERL} {IGRL} {NGRL} {TGRL} {EGRL}

{INAL} {ITAL} {NTAL} {IEAL} {NEAL}

{TEAL} {IGAL} {NGAL} {TGAL} {EGAL}

{IRAL} {NRAL} {TRAL} {ERAL} {GRAL}

Combinations displayed:70

1. **Список литературы**
2. Материалы учебного плана по предмету «Программирование графических приложений».
3. Волосатова Т.М., Родионов С. В., Беломойцев Д.Е. «Разработка графических приложений в среде X Window System» .

