Частное учреждение образования

«Колледж бизнеса и права»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Ведущий методист колледжа  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Паскал  « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 года |
| Специальность 2-40 01 01 «Программное обеспечение информационных технологий» | Учебная дисциплина «Основы алгоритмизации и программирование» |

**Лабораторная работа № 34**

**Инструкционно-технологическая карта**

Тема: Решение задач с использованием генераторов множества всех подмножеств, k-элементных подмножеств множества, разбиения множества на подмножества.

Цель: Научиться создавать алгоритмы и программы для решения задач с использованием генераторов множества всех подмножеств, k-элементных подмножеств множества, разбиения множества на подмножества.

Время выполнения: 2 часа.

1. **Порядок выполнения работы**
2. Изучить теоретические сведения к лабораторной работе.
3. Разработать на языке С++ программу вывода на экран решения задачи в соответствии с вариантом индивидуального задания, указанным преподавателем.
4. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.
5. Ответить на контрольные вопросы.
6. **Теоретические сведения**

**Генерация размещений**

Пусть задано некоторое конечное множество из **N** различных элементов. И требуется выбрать из него **M** элементов. Выбранные **M** из предложенных **N** элементов называются **выборкой**. Если важен порядок, в котором произведена выборка элементов, то говорят об **упорядоченной выборке**, если порядок не важен, то говорят о **неупорядоченной выборке**.

В комбинаторике упорядоченная выборка объемом **M** элементов из предложенного множества **N** различных элементов называется **размещением** из **N** элементов по **M**. Все комбинации внутри размещения могут отличаться друг от друга как самими элементами, так и порядком их расположения. Различают также **размещения без повторений** (когда все M элементов внутри выборки различны) и **размещения с повторениями**.

Задача: Найти все возможные размещения из множества элементов {1,2,3} по 2. Существуют следующие размещения:

**1:** 1 2

**2:** 1 3

**3:** 2 1

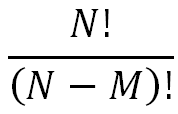
**4:** 2 3

**5:** 3 1

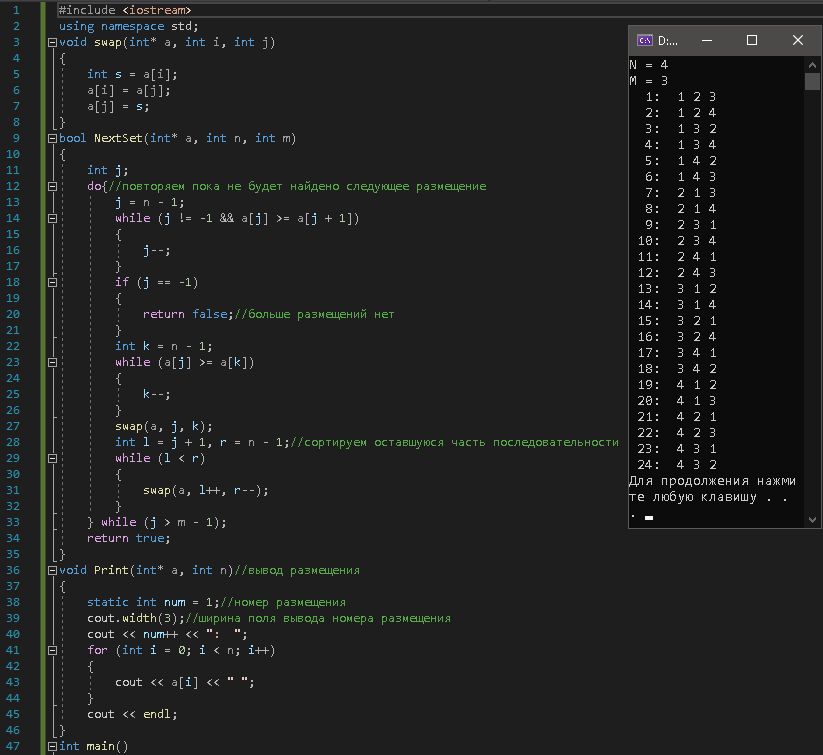
**6:** 3 2

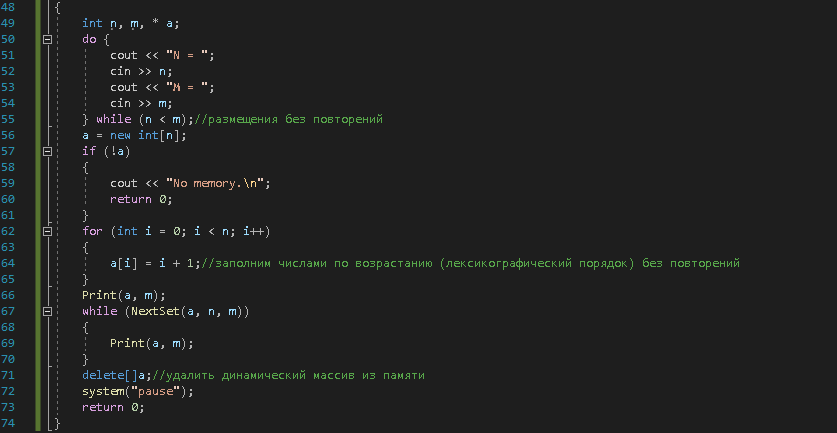
**Размещения без повторений**

В случае размещения без повторений количество элементов множества **N≥M**.  
Количество размещений без повторений для **N** различных элементов по **M** составляет составляет



В случае если **N=M**, количество размещений совпадает с количеством перестановок без повторений и составляет **N!**. Такое же количество размещений получится в случае если **M=N-1**. Рассмотрим задачу получения всех размещений без повторений для чисел **1…N** по **M**. Для генерации всех возможных размещений из N по M в лексикографическом порядке воспользуемся [ранее рассмотренным решением](https://prog-cpp.ru/permutation/) для генерации перестановок без повторений. Реализация на С++:

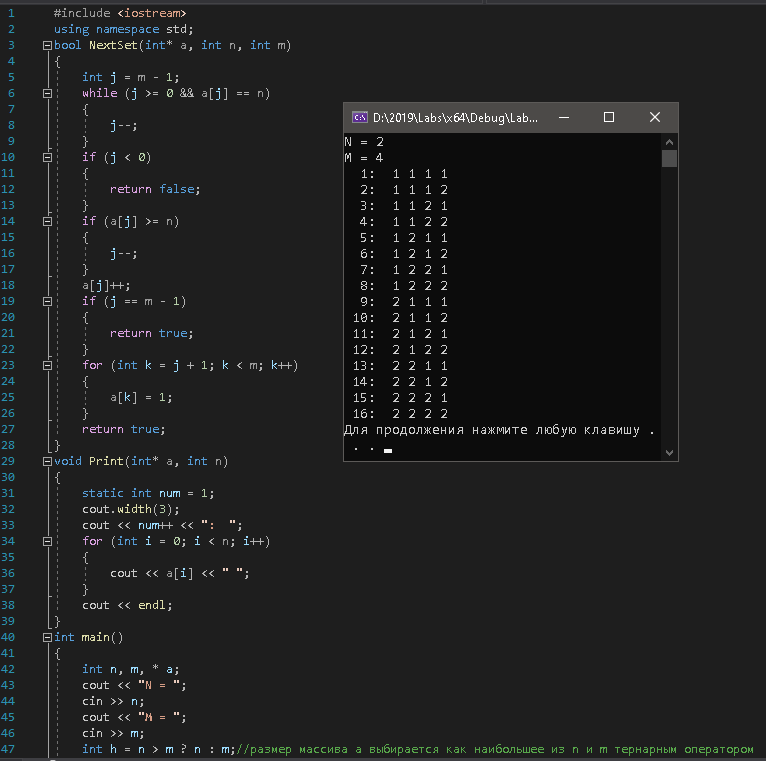


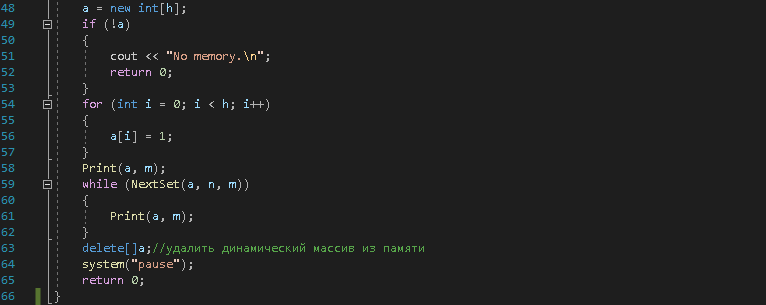


**Размещения с повторениями**

В случае если элементы из множества **N** могут повторяться в выборках по **M** элементов, общее количество таких размещений составляет NM.

При этом не накладывается ограничений на размерность рассматриваемого массива **N** по сравнению с **M**(может быть как **N≥M**, так и **N<M**). Примером такой задачи может служить подбор шифра сейфа из **M**<10 цифр, которые должны набираться последовательно и могут повторяться. Для генерации размещений с повторениями удобно было бы использовать **M** вложенных циклов, однако такое решение применимо только к задаче, в которой заранее известно число **M**. Поэтому рассмотрим более общий вариант генерации размещений с повторениями. Реализация на С++:





**Генерация сочетаний**

В комбинаторике сочетанием из **N** различных элементов по **M** называется набор **M** элементов, выбранных из множества **N** элементов. Такие наборы отличаются только вхождением в них **M** определенных элементов, порядок следования элементов в таком наборе не важен. Наборы, отличающиеся только порядком следования элементов (но не составом), считаются одинаковыми, и этим сочетания отличаются от [размещений](https://prog-cpp.ru/placement/).

**Сочетания без повторений**

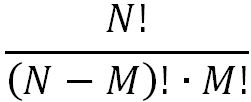
Задача: найти все возможные сочетания без повторений из множества элементов {1, 2, 3} по 2. Существуют следующие сочетания:

**1:** 1 2

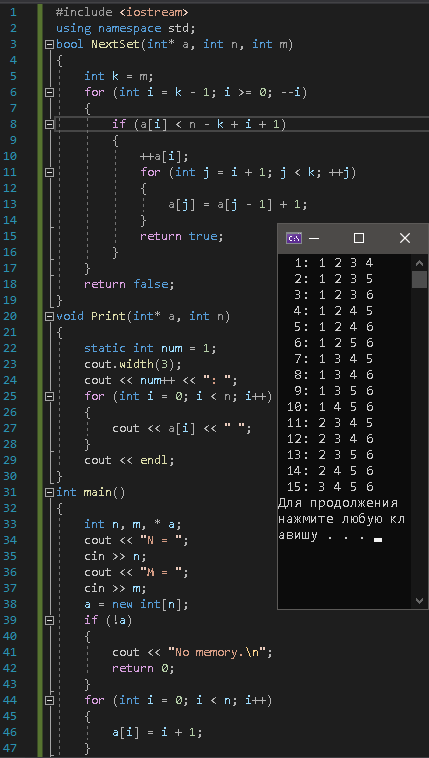
**2:** 1 3

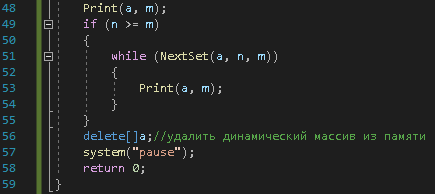
**3:** 2 3

Количество возможных сочетаний без повторений из **N** элементов по **M** можно определить по формуле (**N≥M**):



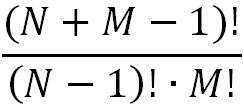
что в **M!** раз меньше соответствующего количества размещений без повторений (поскольку сочетания без повторений не зависят от порядка следования элементов). Рассмотрим задачу получения всех сочетаний для чисел **1…N** по **M**. Реализация на С++:





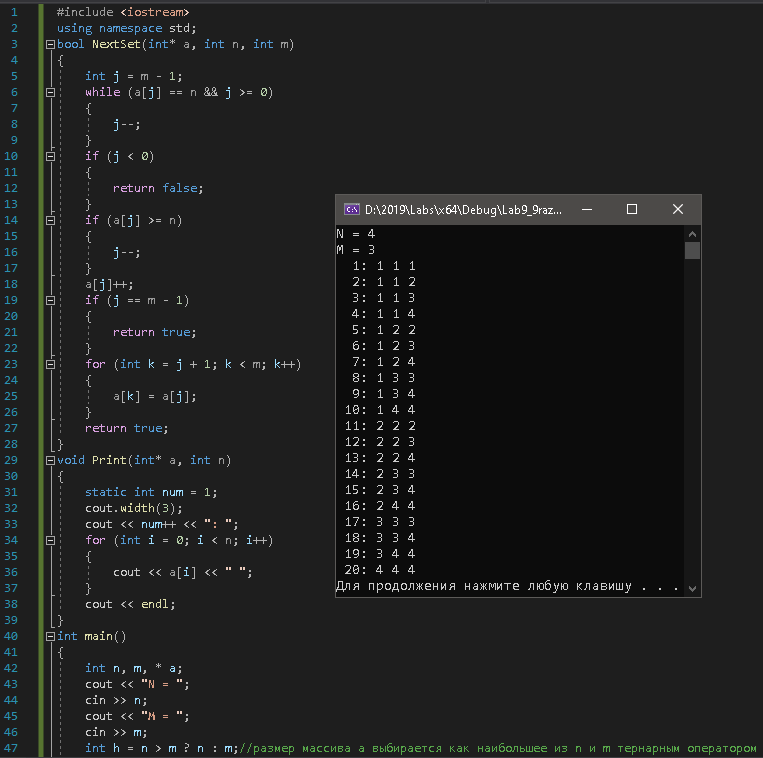
**Сочетания с повторениями**

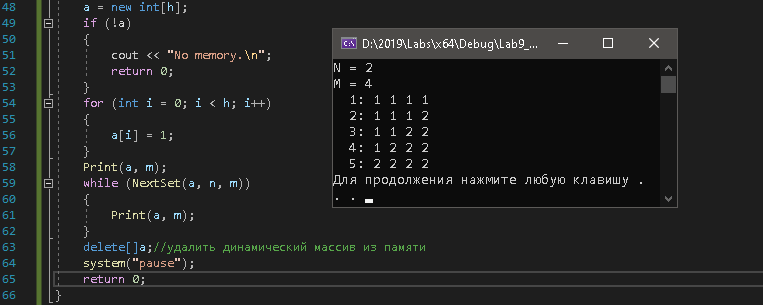
Сочетаниями с повторениями называются наборы по **M** элементов, в которых каждый элемент множества **N** может участвовать несколько раз. При этом на соотношение значений **M** и **N** не накладывается никаких ограничений, а общее количество сочетаний с повторениями составляет



Примером такой задачи может служить выбор **M**открыток из **N** всеми возможными способами.

Для генерации сочетаний с повторениями воспользуемся решением для генерации размещений с повторениями, рассмотренным [в этой статье](https://prog-cpp.ru/placement/#repeat). Реализация на С++

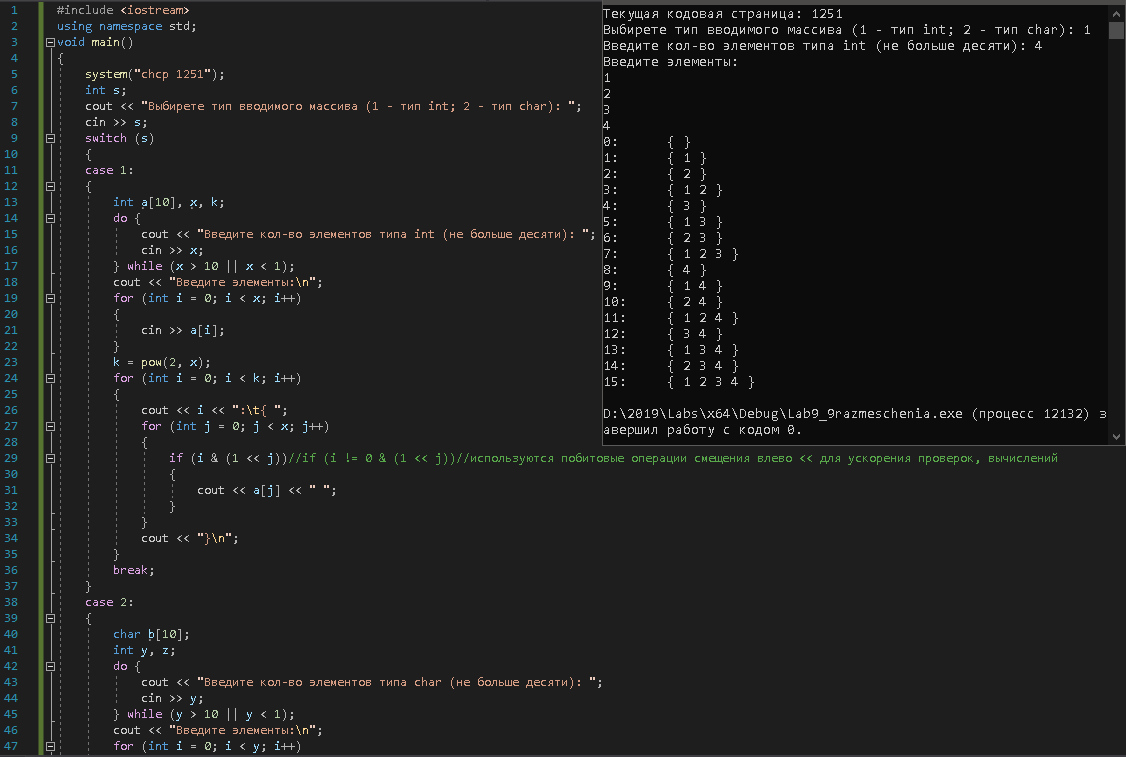


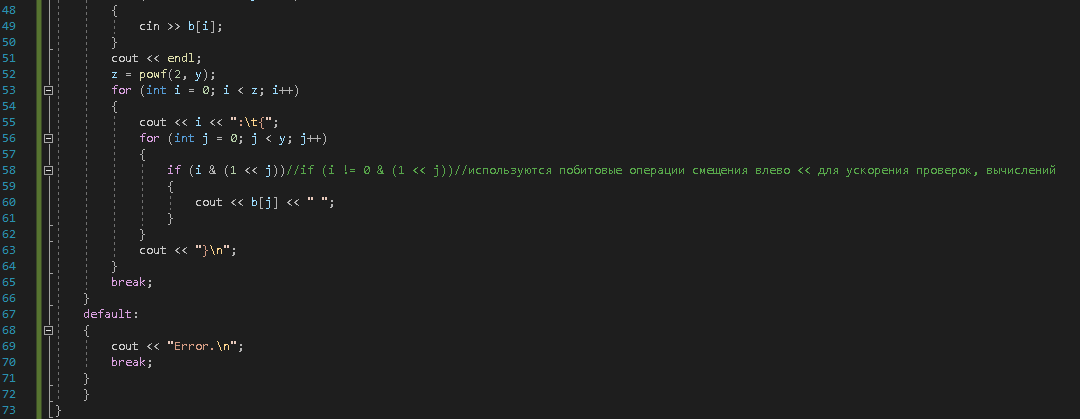


1. **Пример выполнения программы**

**Разбиение множества на подмножества**

Может встретиться задача разбиения данного множества на подмножества, которые состоят из элементов данного множества, причем они могут быть по длине меньше данного множества или равны ему. Будем считать, что переставлять элементы в текущем множестве нельзя (эту задачу мы уже решили ранее) и что подмножества могут быть длиной от нуля (пустое множество) до величины данного множества включительно (т.е. они совпадают).





1. **Задания по вариантам**
2. Дан целочисленный массив, вводимый пользователем с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
3. Дан вещественный массив, вводимый пользователем с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
4. Дан логического типа массив, вводимый пользователем с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
5. Дан символьный (английская раскладка) массив, вводимый пользователем с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
6. Программно сгенерировать целочисленный массив, размер которого вводит пользователь с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
7. Программно сгенерировать вещественный массив, размер которого вводит пользователь с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
8. Программно сгенерировать символьный (английская раскладка) массив, размер которого вводит пользователь с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
9. Программно сгенерировать массив логического типа, размер которого вводит пользователь с клавиатуры. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
10. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив короткого целочисленного типа. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
11. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив длинного целочисленного типа. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
12. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив короткого беззнакового целочисленного типа. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
13. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив длинного беззнакового целочисленного типа. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
14. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив типа float. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
15. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив типа double. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
16. Пользователь вводит с клавиатуры значения (неотсортированные, возможны повторения) в массив типа bool. Разбить данный массив на подмассивы, учитывая все варианты расположения элементов массива.
17. **Контрольные вопросы**
18. Что такое выборка?
19. Что такое упорядоченная выборка?
20. Что такое неупорядоченная выборка?
21. Что такое размещение?
22. Что такое размещение с повторениями?
23. Что такое размещение без повторений?
24. Что такое сочетание?
25. Что такое сочетание без повторений?
26. Что такое сочетание с повторениями?
27. Объясните, что такое разбиение множества на подмножества?

**Литература**

**Дейтел,** Х.М. Как программировать на С++ / Х.М. Дейтел, П.Дж. Дейтел . – М. : Бином-Пресс , 2018 . – 1456 с.

**Павловская**, Т.А. С++. Объектно-ориентированное программирование : практикум / Т.А. Павловская, Ю.А. Щупак . – СПб. : Питер , 2019 . – 265 с.

**Страуструп**, Б. Язык программирования С++ / Б. Страуструп . – СПб. : Бином-Пресс , 2019 . – 1054 с.

Преподаватель Шаляпин Ю.В.

|  |
| --- |
| Рассмотрено на заседании цикловой  комиссии ПОИТ № 10  Протокол №\_\_\_\_от «\_\_\_»\_\_\_\_\_\_\_\_\_\_2021 г.  Председатель ЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |