Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01– «Информатика и вычислительная техника»

**Лабораторная работа № 1**

**по дисциплине**

**«Дискретная математика»**

**на тему**

**«Создание калькулятора множеств»**

Выполнил студент гр. ИВТ-23-1б

Попова Мария Вячеславовна

Проверил:

ст. преп.

Рустамханова Г.И.

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Содержание**

[Цель и задачи работы 3](#_Toc179224041)

[Этапы выполнения 4](#_Toc179224042)

[1 Выбор языка для реализации калькулятора 4](#_Toc179224043)

[2 Описание работы программы 4](#_Toc179224044)

[3 Описание основных функций программы 7](#_Toc179224045)

[Вывод по лабораторной работе 8](#_Toc179224046)

[Список использованных источников 9](#_Toc179224047)

# Цель и задачи работы

Цель работы спроектировать и написать калькулятор для множеств.

Задачи:

1. Калькулятор должен предоставлять возможность задать как минимум 3 множества.
2. Калькулятор должен быть представлен в универсуме из целых чисел от -50 до 50.
3. Реализовать возможность задать множество следующими способами:
   1. случайная наполненность множества,
   2. ручной ввод,
   3. совокупность условий.
4. Реализовать меню операций.
5. Предусмотреть возможность задать выражение по формуле.

# Этапы выполнения

## 1 Выбор языка для реализации калькулятора

Был выбран язык программирования C++, так как есть опыт работы на этом языке по выполнению лабораторных работ.

## 2 Описание работы программы

Пользователь взаимодействует с программой через терминал.

Реализована возможность создания множества тремя способами: ручной ввод (рис. 1), случайное наполнение (рис. 2), заполнение по условию (рис. 3). Элементы множества не могут повторяться.

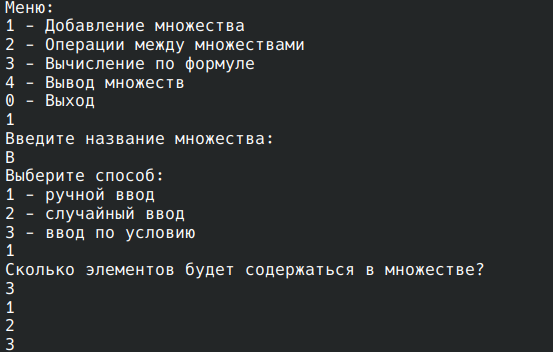


Рисунок 1 – Добавление множества вручную

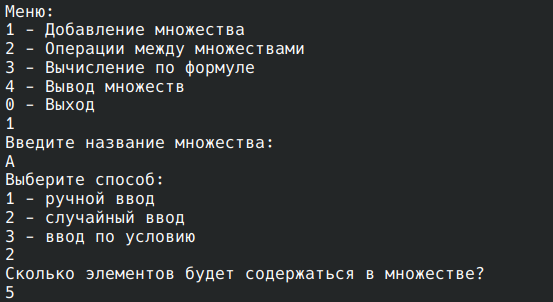


Рисунок 2 – Добавление множества со случайными элементами

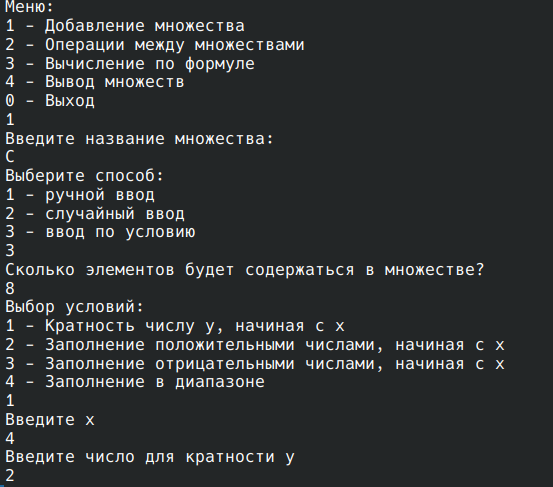


Рисунок 3 – Добавление множества с заполнением по условию

Пользователь может просмотреть все созданные множества (рис. 4).

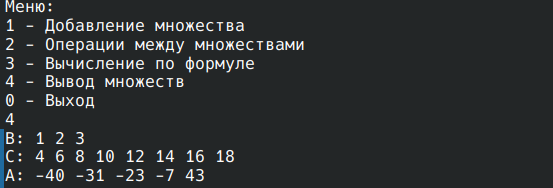


Рисунок 4 – Вывод созданных множеств

Реализовано меню операций (рис. 5, 6):

* u – операция объединения,
* i – операция пересечения,
* d – операция разности,
* s – операция симметрической разности,
* ! – дополнение.

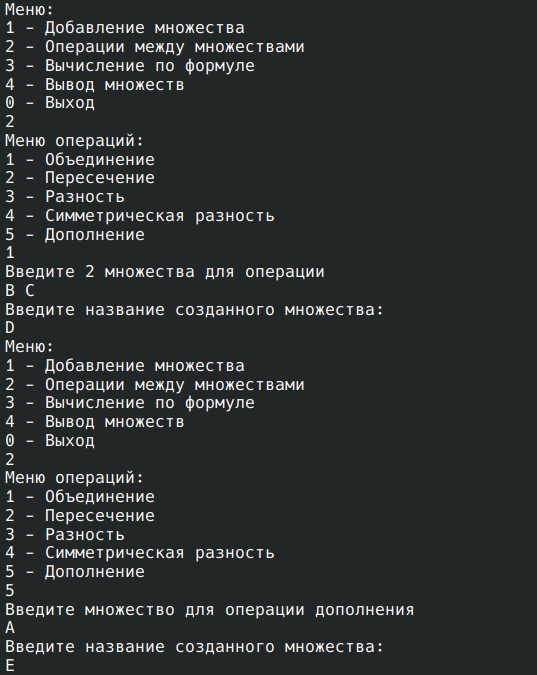


Рисунок 5 – Меню операций

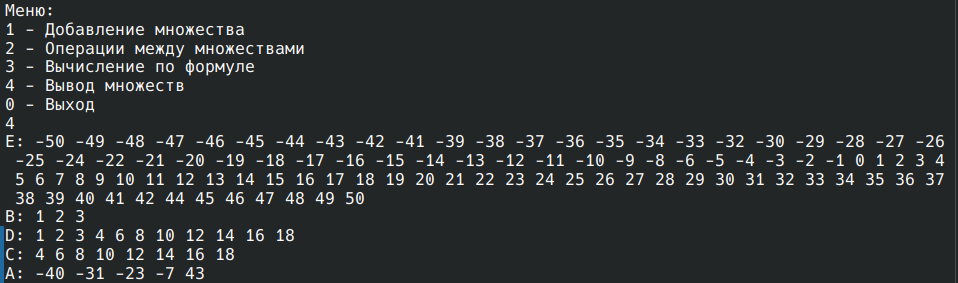


Рисунок 6 – Вывод новых множеств, полученных при выполнении операций

Пользователь может ввести выражение в виде формулы (рис. 7).

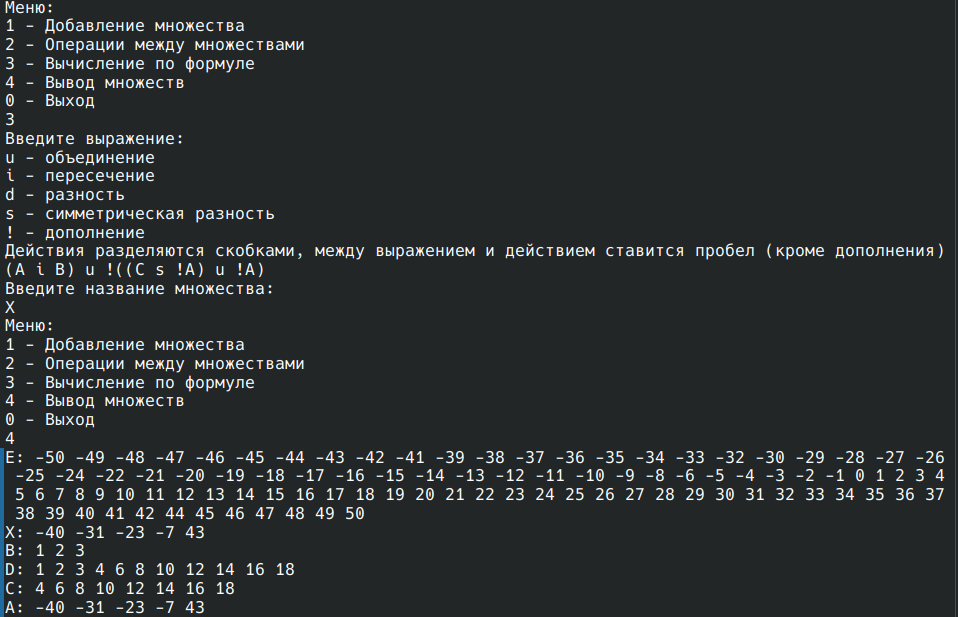


Рисунок 7 – Результат вычисления выражения

## 3 Описание основных функций программы

Весь исходный код можно посмотреть в репозитории на GitHub: https://github.com/RuFo2/Labs\_PSTU\_2023/tree/main/Sem\_3.gitkeep/discrete\_mathematics.gitkeep/Lab\_1.gitkeep

В функции main объявляется хэш-таблица, в которой будут храниться множества (ключ – название множества, значение – вектор целых чисел, которые содержаться в множестве). Также в main в цикле реализовано главное меню программы.

Функция handleOption обрабатывает выбор пользователя в главном меню и вызывает нужные функции.

Для добавления множества в хэш-таблицу используется функция getSets.

Меню операций реализовано в operation.

Вывод множеств реализован в printSet.

Для введения выражения (формулы) отвечает функция formula. Затем выражение передается в функцию getOperations. getOperations возвращает вектор строк, в котором содержаться последовательно все действия. Далее этот вектор строк передается в функцию calc, где все действия вычисляются.

Функция unionSet реализует объединение множеств. Объединение двух множеств A и B — это множество, содержащее все элементы, которые принадлежат хотя бы одному из множеств. Обозначается как A∪B [1].

Функция intersectionSet реализует пересечение множеств. Пересечение двух множеств A и B — это множество, содержащее все элементы, которые принадлежат одновременно обоим множествам. Обозначается как A∩B [1].

Функция differenceSet реализует разность множеств. Разность множества A и B — это множество, содержащее все элементы, которые принадлежат A, но не принадлежат B. Обозначается как A∖B [1].

Функция symmetricSet реализует симметрическую разность множеств. Симметрическая разность двух множеств A и B — это множество, содержащее элементы, которые принадлежат только одному из множеств, но не обоим сразу. Обозначается как AΔB [1].

Функция complimentSet реализует дополнение множества. Дополнение множества A относительно универсального множества U — это множество, содержащее все элементы, которые принадлежат U, но не принадлежат A. Обозначается как ˉA или !A [1].

# Вывод по лабораторной работе

Спроектирован и реализован калькулятор множеств на языке программирования C++. В ходе выполнения лабораторной работы были закреплены знания в теории множеств из дискретной математики.

# Список использованных источников

* + - 1. Множества. Операции над множествами. Отображение множеств. Мощность множества // MathProfi URL: http://mathprofi.ru/mnozhestva.html (дата обращения: 05.10.2024).