Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

**Лабараторная работа №1**

На тему:

«Алгоритмы операцй над множествами и их реализация»

**Выполнил:**

Сечейко Н.В.

**Проверил:**

Рудьман И.О.

Минск 2024

*Цель*:

Научиться составлять алгоритм выполнения операций над множествами. Проверить выполнение составленного алгоритма в выбранной среде разработки на выбранном языке программирования(C++).

*Определения*:

**Множество** – набор(совокупность) каких-либо неповторяющихся элементов.

**Мощность множества** – характеристика, отражающая количество элементов в данном множестве.

**Универсум** – множество, содержащее все данные множества.

*Особенности:*

**1**. Мощность множества, значения элементов множества, начало и конец универсума вводятся пользователем самостоятельно в процессе выполнения программы.

**2**. Программа выполняет проверку на целочисленный и неоторицательный ввод(мощность множества), проверяет, чтобы значения множеств не выходили за пределы универсума (см. раздел «Алгоритм выполнения»).

*Язык реализации действий над множествами*: C++

*Среда разработки программы*: visual studio

*Алгоритм выполнения:*

**1**. Подключаем нужные библиотеки.



-iostream – стандартная библиотека( input/output stream) для работы на C++

**2**. Использования пространства имён std для облегчения использования функций cin и cout библиотеки iostream.



**3**. В стандартной функции main() инициализируем русский язык.



**4**. Объявляем переменные, массивы.



































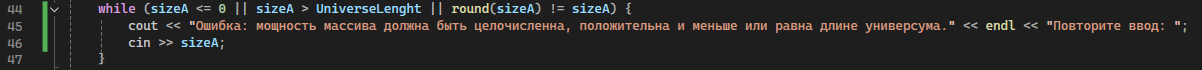




**5**. Запрашиваем от пользователя ввести мощности массивов A и B. Тут же делаем проверку на целочисленный ввод, посредством сравнения введённой пользователём переменной sizeA/sizeB и её округлением (функция round() из <cmath>). Если округление ввёденного числа не равно самому числу, значит было введенно вещественное(нецелое) число. С помощью цикла while, пока пользователь не введёт целочисленное и неотрицательное число, пользователь возвращается к вводу мощности массива.

Если всё-таки было введено целое число, переменные sizeA, sizeB принимают введённые значения.

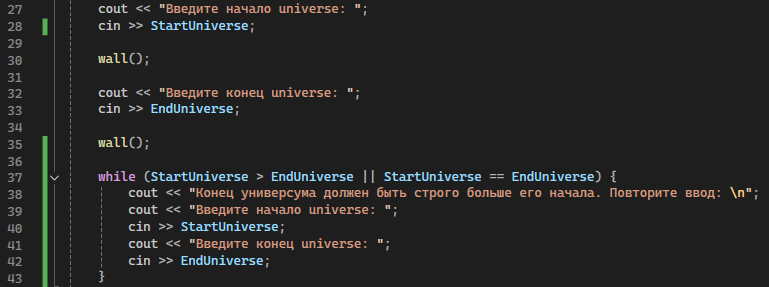




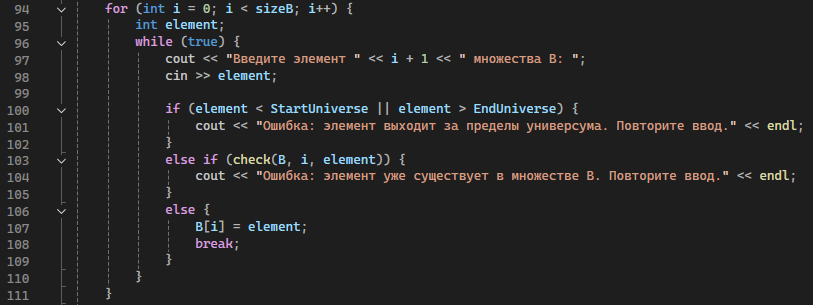
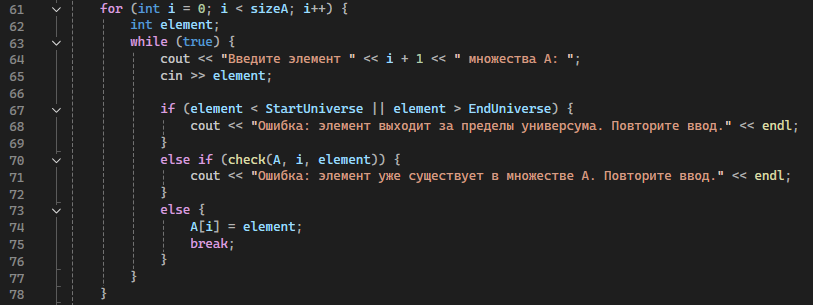




**6**. Запрашиваем у пользователя ввести начальное и конечное значение универсума, тут же делаем проверку, чтобы большее значение универсума было строго больше, чем меньшее. Аналогично с пунктом 5 возвращаем пользователя к вводу пределов массива до тех пор, пока не будет введено правильно значение. Использованы циклы for, while и ветвление с if else.

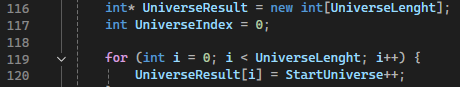


**7**. Говорим пользователю заполнить значения для элементов массива, одновременно проверяя, чтобы они не повторялись. В случае ввода одинаковых значений программа просит повторить ввод. Использованы циклы for, while и ветвление с if else.

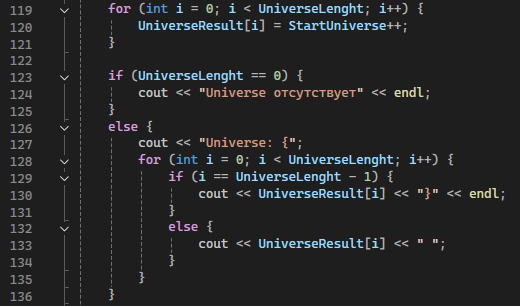
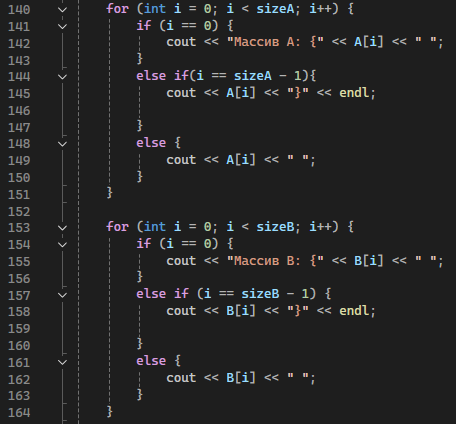


**8**. Также в пункте 7 идёт проверка, чтобы значения элементов массивов не выходили за пределы универсума. В случае, если значение элемента выходит за пределы универсума, программа просит повторить ввод.

**9**. Вносим элементы в массив универсума при помощи цикла for и ветвления с if else.



**10**. Выводим введённые значения пользователем, тоесть оба массива(A и B) и универсум, одновременно выполнив проверку на то, чтобы в массиве универсума был хотя бы один элемент. Использованы цикл for и ветвление с if else.



**11**. Объединение.

Ввод: даны множества A и B, пустое множество O.

*11.1*. Все элементы множества A занести во множество O.

*11.2*. Взять первый элемент множества B.

*11.3.* Взять первый элемент множества O.

*11.4.* Если выбранный элемент множества B не равен выбранному элементу O, то перейдите к пункту 11.7.

*11.5.* Если выбранный элемент множества В равен выбранному элементу множества O, то перейдите к пункту 11.11.

*11.6.* Если выбранный элемент множества O — последний, перейдите к пункту 11.10.

*11.7.* Если выбранный элемент множества O — не последний, то возьмите следующий элемент множества O.

*11.8*. Перейдите к пункту 11.5.

*11.9*. Добавьте выбранный элемент множества B во множество O.

*11.10*. Если выбранный элемент множества B — последний, то перейдите к пункту 11.13.

*11.11*. Если выбранный элемент множества B — не последний, то возьмите следующий элемент множества B.

*11.12*. Перейдите к пункту 11.4.

*11.13*. Множество O – есть искомое множество-объединение множеств A и B. Выводим его пользователю.

**12**. Пересечение.

Ввод: даны два множества A и B, пустое множество P(будущее множество пересечения).

*12.1*. Взять первый элемент множества А.

*12.2*. Взять первый элемент множества В.

*12.3*. Если выбранный элемент А равен выбранному элементу В, то перейти к пункту 12.5.

*12.4*. Если выбранный элемент А не равен выбранному элементу В, то перейти к пункту 12.6.

*12.5*. Добавить выбранный элемент множества А во множество P и перейти к пункту 12.9.

*12.6*. Если выбранный элемент множества В — последний, то перейти к пункту 12.19.

*12.7*. Если выбранный элемент множества В — не последний, то взять следующий элемент множества В.

*12.8*. Перейти к пункту 12.3.

*12.9*. Если выбранный элемент множества А — последний, то перейти к пункту 12.12.

*12.10*. Если выбранный элемент множества А — не последний, то взять следующий элемент множества А.

*12.11*. Перейти к пункту 12.2.

*12.12*. Вывести полученное множество P.

**13**. Разность A/B.

Ввод: даны два множества A и B, пустое множество R(будущее множество разности).

*13.1*. Взять первый элемент множества А.

*13.2*. Взять первый элемент множества В.

*13.3*. Если выбранный элемент множества А равен выбранному элементу множества В, то перейти к пункту 13.8.

*13.4*. Если выбранный элемент множества В является последним, перейти к пункту 13.7.

*13.5*. Если выбранный элемент множества В не является последним, взять

следующий элемент множества В.

*13.6*. Перейти к пункту 13.3.

*13.7*. Добавить взятый элемент множества А в множество R.

*13.8*. Если выбранный элемент множества А является последним, перейти к пункту 13.11.

*13.9*. Если выбранный элемент множества А не является последним, взять следующий элемент множества А.

*13.10*. Перейти к пункту 13.2.

*13.11*. Полученное множество R – искомое множество разности A и B. Выводим его пользователю.

**14**. Разность B/A.

Ввод: даны два множества A и B, пустое множество R(будущее множество разности).

*14.1*. Взять первый элемент множества B.

*14.2*. Взять первый элемент множества A.

*14.3*. Если выбранный элемент множества B равен выбранному элементу множества A, то перейти к пункту 14.8.

*14.4*. Если выбранный элемент множества A является последним, перейти к пункту 14.7.

*14.5*. Если выбранный элемент множества A не является последним, взять

следующий элемент множества A.

*14.6*. Перейти к пункту 14.3.

*14.7*. Добавить взятый элемент множества B в множество R.

*14.8*. Если выбранный элемент множества B является последним, перейти к пункту 14.11.

*14.9*. Если выбранный элемент множества B не является последним, взять следующий элемент множества B.

*14.10*. Перейти к пункту 14.2.

*14.11*. Полученное множество R – искомое множество разности B и A. Выводим его пользователю.

**15**. Симметричная разность.

Симметрическая разность будет состаять из объединения(п.11) разностей A/B(п.13) и B/A(п.14).

Ввод: даны два множества A и B. S – пустое множество(будущее множество симметрической разности). X – пустое множество( будущее промежуточное множество разности A/B), Y – пустое множество ( будущее промежуточное множество разности B/A).

*15.1*. Разность A/B.

*15.1.1.* Взять первый элемент множества А.

*15.1.2.* Взять первый элемент множества В.

*15.1.3*. Если выбранный элемент множества А равен выбранному элементу множества В, то перейти к пункту 15.1.8.

*15.1.4*. Если выбранный элемент множества В является последним, перейти к пункту 15.1.7.

*15.1.5*. Если выбранный элемент множества В не является последним, взять

следующий элемент множества В.

*15.1.6.* Перейти к пункту 15.1.3.

*15.1.7*. Добавить взятый элемент множества А в множество X.

*15.1.8*. Если выбранный элемент множества А является последним, перейти к пункту 15.1.11.

*15.1.9.* Если выбранный элемент множества А не является последним, взять следующий элемент множества А.

*15.1.10*. Перейти к пункту 15.1.2.

*15.1.11*. X – промежуточное множество, которое равно разности A/B. Перейти на пункт 15.2.

*15*.2. Разность B/A.

*15.2.1*. Взять первый элемент множества B.

*15.2.2*. Взять первый элемент множества A.

*15.2.3*. Если выбранный элемент множества B равен выбранному элементу множества A, то перейти к пункту *15.2.*8.

*15.2.4*. Если выбранный элемент множества A является последним, перейти к пункту *15.2.*7.

*15.2.5*. Если выбранный элемент множества A не является последним, взять

следующий элемент множества A.

*15.2.6*. Перейти к пункту *15.2.*3.

*15.2.7*. Добавить взятый элемент множества B в множество Y.

*15.2.8*. Если выбранный элемент множества B является последним, перейти к пункту 15.2.11.

*15.2.9*. Если выбранный элемент множества B не является последним, взять следующий элемент множества B.

*15.2.10*. Перейти к пункту *15.2.*2.

*15.2.11*. Y- промежуточное значение, равное разности B/A. Перейти на пункт 15.3.

15.3. Объединение X и Y.

15.3.*1*. Все элементы множества X занести во множество S.

15.3.*2*. Взять первый элемент множества Y.

15.3.*3.* Взять первый элемент множества S.

15.3.*4.* Если выбранный элемент множества Y не равен выбранному элементу S, то перейдите к пункту 15.3.7.

15.3.*5.* Если выбранный элемент множества Y равен выбранному элементу множества S, то перейдите к пункту 15.3.11.

15.3.*6.* Если выбранный элемент множества S — последний, перейдите к пункту 15.3.10.

15.3.*7.* Если выбранный элемент множества S — не последний, то возьмите следующий элемент множества S.

15.3.*8*. Перейдите к пункту 15.3.5.

15.3.*9*. Добавьте выбранный элемент множества Y во множество S.

15.3.*10*. Если выбранный элемент множества Y — последний, то перейдите к пункту 15.3.13.

15.3.*11*. Если выбранный элемент множества Y — не последний, то возьмите следующий элемент множества Y.

15.3.*12*. Перейдите к пункту 15.3.4.

15.3.*13*. Множество S – есть искомое множество симметрической разности множеств A и B. Выводим его пользователю.

**16**. Дополнения до универсвума A.

Ввод: дано множество A и универсум U. UA – пустое множество( будущее множество A, дополненное до универсума).

Дополнением до универсума множества A будет являтся разность U/A.

*16.1*. Взять первый элемент множества U.

*16.2*. Взять первый элемент множества A.

*16.3*. Если выбранный элемент множества U равен выбранному элементу множества A, то перейти к пункту 16.8.

*16.4*. Если выбранный элемент множества A является последним, перейти к пункту 16.7.

*16.5*. Если выбранный элемент множества A не является последним, взять

следующий элемент множества A.

*16.6*. Перейти к пункту 16.3.

*16.7*. Добавить взятый элемент множества U в множество UA.

*16.8*. Если выбранный элемент множества U является последним, перейти к пункту 16.11.

*16.9*. Если выбранный элемент множества U не является последним, взять следующий элемент множества U.

*16.10*. Перейти к пункту 16.2.

*16.11*. Полученное множество UA – искомое дополнение множества A до универсума. Выводим его пользователю.

**17**. Дополнения до универсвума B.

Ввод: дано множество B и универсум U. UB – пустое множество( будущее множество B, дополненное до универсума).

Дополнением до универсума множества B будет являтся разность U/B.

*17.1*. Взять первый элемент множества U.

*17.2*. Взять первый элемент множества B.

*17.3*. Если выбранный элемент множества U равен выбранному элементу множества B, то перейти к пункту 17.8.

*17.4*. Если выбранный элемент множества B является последним, перейти к пункту 17.7.

*17.5*. Если выбранный элемент множества B не является последним, взять

следующий элемент множества B.

*17.6*. Перейти к пункту 17.3.

*17.7*. Добавить взятый элемент множества U в множество UB.

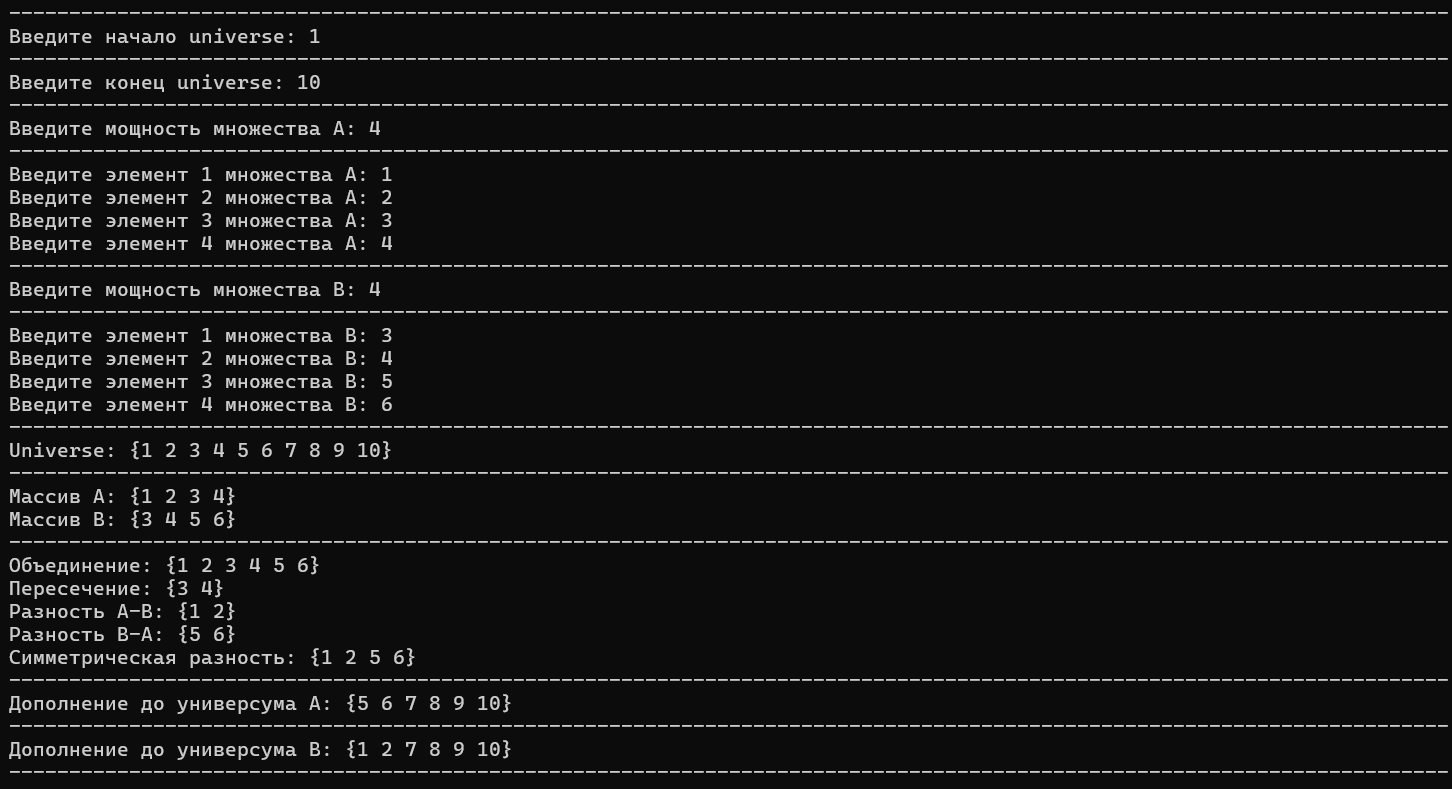
*17.8*. Если выбранный элемент множества U является последним, перейти к пункту 17.11.

*17.9*. Если выбранный элемент множества U не является последним, взять следующий элемент множества U.

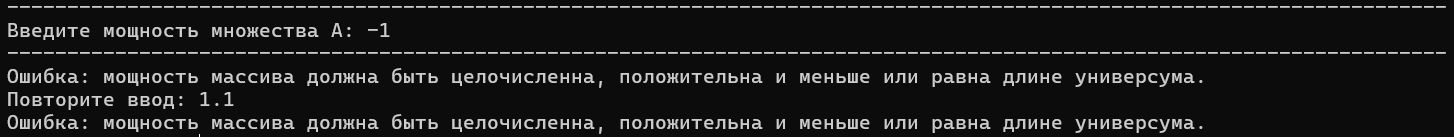
*17.10*. Перейти к пункту 17.2.

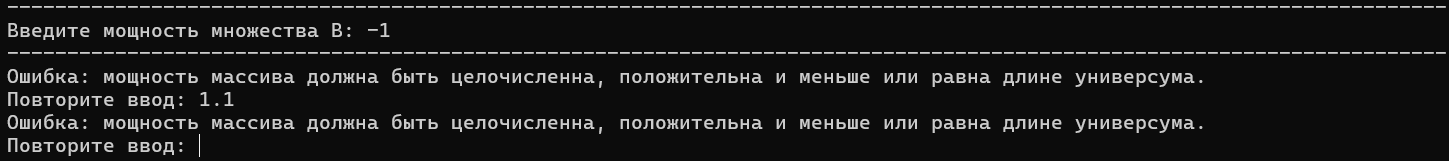
*17.11*. Полученное множество UB – искомое дополнение множества B до универсума. Выводим его пользователю.

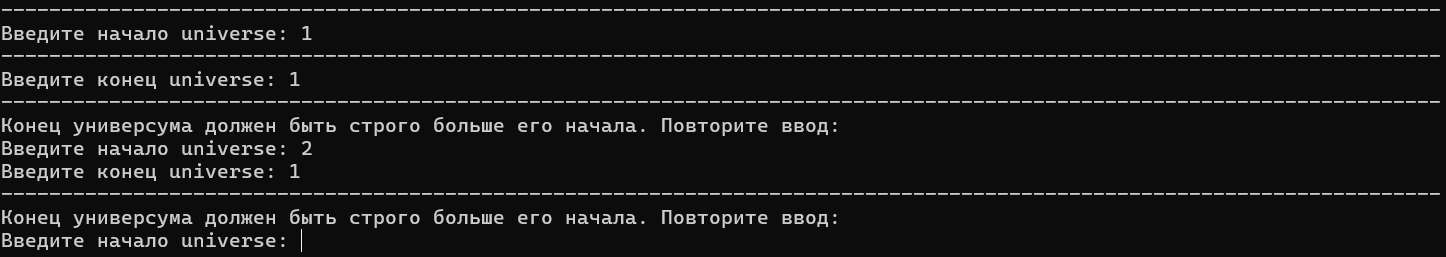
*Пример ввода данных и ответа пользователю:*

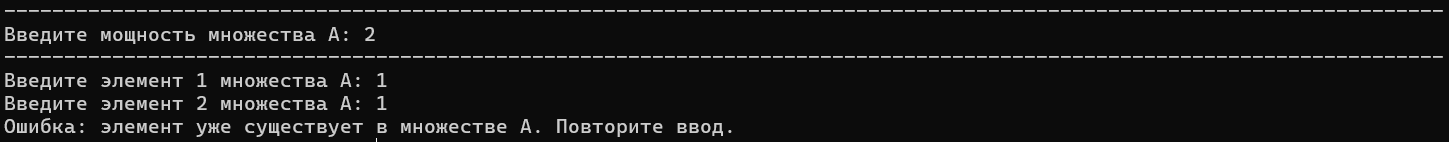
**

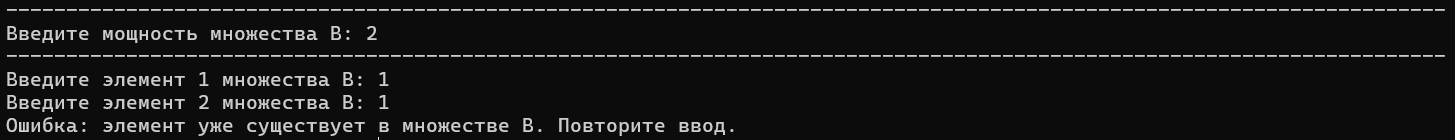
*Ответы в случае ошибок:*

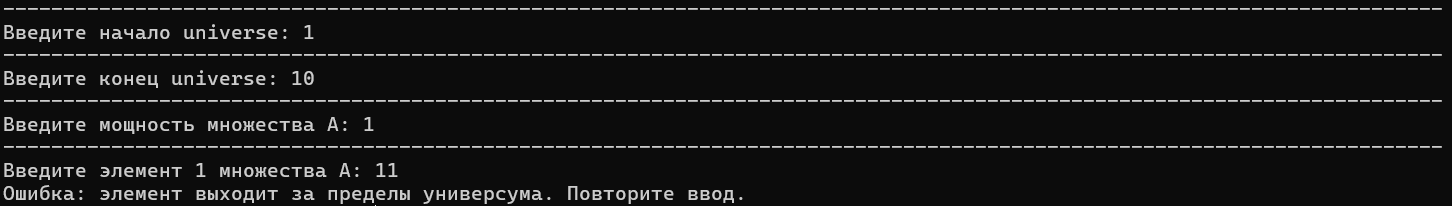
**

**

**

**

**

**

*Вывод:*

Мной были подробно спроектированны и описаны пункты алгоритма проведения различных элементарных действий над множествами.В результате работы я реализовал выполнение составленного мной алгоритма на языке программирования C++, используя среду разработки Visual Studio.