Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно–графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Программная реализация задач |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020–РГР–02.03.02–№ 20 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Рыговской Елены Александровны | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | 1 |  | Группа | | ФИТ-242 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | Фундаментальная информатика и информационные технологии | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  |  | | | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2024

**Содержание**

[Введение 3](#_Toc26999)

[Задача 1 – Крестьянин и чёрт. 6](#_Toc9848)

[Постановка задачи 1 6](#_Toc5289)

[Ход решения задачи 1 8](#_Toc32296)

[Задача 2 – Упаковки молока 11](#_Toc31337)

[Постановка задачи 2 11](#_Toc7322)

[Задача 3 – Постройка дома 15](#_Toc28058)

[Постановка задачи 3 15](#_Toc7826)

[Ход решения задачи 3 16](#_Toc32692)

[Задача 4 –Интенсификация производства 19](#_Toc8286)

[Постановка задачи 4 19](#_Toc20892)

[Ход решения задачи 4 20](#_Toc9694)

[Заключение 22](#_Toc19482)

[Список используемой литературы 23](#_Toc8079)

# **Введение**

C# — это язык программирования от компании Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и даже графику для метавселенных. Один из ведущих разработчиков языка — легендарный Андерс Хейлсберг, который до C# успел сделать Turbo Pascal и Delphi, а после — TypeScript (майкрософтовский JS на стероидах).

Если коротко, этот язык:

-*Кросс-платформенный* — запускается почти на любом железе.

-*Объектно-ориентированный* — состоит из классов и объектов, которые умеют передавать свойства друг другу.

-*Постоянно развивается* — для тех, кто любит учиться.

-*Дружит с экосистемой Windows* — для этого и был написан.

C# — [пятый](https://www.tiobe.com/tiobe-index/" \t "https://skillbox.ru/media/code/yazyk-s-gde-ispolzuyut-chto-pishut-kak-poyavilsya-i-chem-khorosh/_blank) по популярности язык программирования в мире. Его используют банки, диджитал-агентства, провайдеры связи и крупные IT-компании. Вот что пишут на C# российские и зарубежные корпорации:

Microsoft — приложения для Windows и Xbox.

Tesla — корпоративные веб-сервисы и программы.

Stack Overflow — серверную логику сайтов.

Сбербанк — 3D-графику и программы виртуальной реальности.

Ozon — складские и логистические системы.

«Яндекс» — приложения для автоматизации продаж.

Преимущества C#:

* Независимость от железа. Программисту не надо адаптировать программу под разные платформы и системы — за него это делает виртуальная машина, вшитая в .NET Framework. В итоге один и тот же код можно запускать на любых устройствах — смартфонах, компьютерах, серверах, банкоматах и даже умных часах.
* Отличная совместимость с Windows. Не зря же язык разработали именно в Microsoft. Так же как [Swift](https://skillbox.ru/media/code/na_kakom_yazyke_pishut_prilozheniya_dlya_ios/?utm_source=media&utm_medium=link&utm_campaign=all_all_media_links_links_articles_all_all_skillbox" \t "_blank) идеально подходит для программирования под экосистему Apple, C# прекрасно вписывается в экосистему Windows.
* Управление памятью. Чтобы программа работала стабильно, её надо иногда чистить от ненужных объектов, ссылок, кэша и прочего мусора. В C# это происходит автоматически — разработчику не надо следить за расходом памяти, бороться с её утечками или удалять мёртвые куски кода.
* Строгая типизация. Когда вы объявляете переменную в C#, надо сначала указать, что в ней лежит — строка, число или массив. Так разрабатывать чуть дольше, зато ваш код работает предсказуемо — числа взаимодействуют с числами, строки со строками и так далее. В языках со слабой типизацией свободы и драйва больше, но есть шанс пропустить ошибку, которая всплывёт в готовой программе.
* Большое сообщество. На С# пишут более миллиона программистов по всему миру. В соцсетях полно чатов и сообществ «шарпистов», где можно задать вопрос, обсудить сложную тему или найти готовое решение. В теории можно даже найти ментора, который поделится знаниями и поможет быстрее освоить язык.
* Синтаксический сахар. В С# есть много способов сократить код, не нарушая логику программы. Программисты называют такие приёмы «синтаксическим сахаром» — они помогают сделать код проще, понятнее и в целом симпатичнее. Сравните, например, как выглядит сложение чисел с «сахаром» и без.

Есть у C# и недостатки, причём довольно серьёзные, — давайте разберём и их тоже.

* Скорость. Когда мы запускаем программу на C#, код исполняется не сразу, а сначала адаптируется под нужное железо. Так мы охватываем больше платформ, но теряем в скорости — программе нужно сделать двойную работу, чтобы просто стартовать. Из-за этого интерфейсы на С# иногда подтормаживают при первом запуске.
* Безопасность. Эксперты говорят, что код на C# легко декомпилировать — то есть перевести из машинного обратно в человеческий. Проблема в том, что так программу может легко прочитать хакер или конкурент — и изучить её уязвимости, украсть фрагменты кода или написать для неё вредоносный софт.
* Мало доступа к железу. Так как С# — язык высокого уровня, на нём редко пишут проекты, где нужно полное взаимодействие с железом, — игровые движки, операционные системы, авиационный софт и так далее. Та же Unity целиком написана на низкоуровневом языке C++, хотя и умеет исполнять С#-команды.

# **Задача 1** – **Крестьянин и чёрт.**

# **Постановка задачи 1**

Идет крестьянин и плачется: "Эхма! Жизнь моя горькая! Заела нужда совсем! Вот в кармане только несколько монет, да и те сейчас нужно отдать. И как это у других бывает, что на всякие свои деньги они еще деньги получают? Хоть бы кто помочь мне захотел".

Только успел это сказать, как глядь, а перед ним черт стоит и говорит: "Вот видишь этот мост через реку. Стоит тебе перейти через мост, и у тебя будет вдвое больше денег, чем есть. Перейдешь опять, и снова станет вдвое больше. Но за то, что я у тебя деньги удваиваю, после каждого перехода ты мне должен отдавать по K монет".

"Ой ли," - сказал крестьянин -"ну-ка, попробуем". Перешел мост, и деньги у него удвоились. Отдал он черту K монет, перешел мост еще раз, и опять деньги удвоились. Снова отдал крестьянин черту K монет.

Однако после Z переходов и отдач черту по K монет оказалось, что у крестьянина не осталось ни одной монеты.

Требуется определить, сколько комбинаций условий перехода через мост может быть, если известно, что у крестьянина изначально было не более MaxN монет. Комбинацией условий перехода является тройка чисел N, K, Z, где N - начальное количество монет у крестьянина, K - количество монет, отдаваемых черту после каждого перехода, Z - количество переходов. Естественно, что для этой тройки должно выполняться условие, что после Z циклов у крестьянина не должно остаться монет.

Входной файл содержит целое число MaxN - максимальное количество, которое может быть изначально у крестьянина (1 £ MaxN £ 2000000000).

Выходной файл должен содержать одно целое число - количество комбинаций условий перехода через мост.

*Пример:*

Input.txt

3

Output.txt

4

*Пояснение: возможны комбинации условий перехода*

N K Z

1 2 1

2 4 1

3 6 1

3 4 2

# **Ход решения задачи 1**

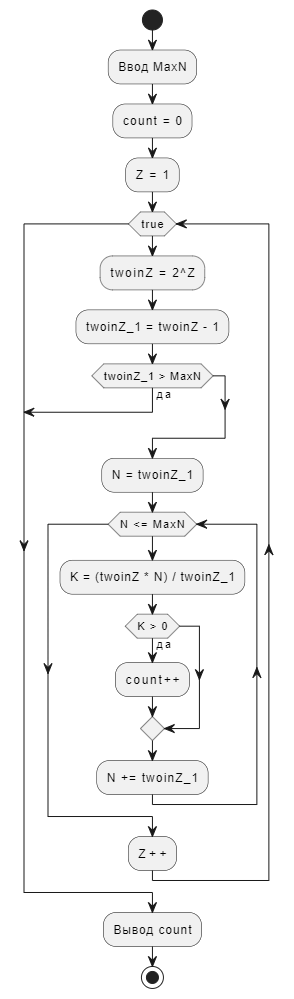


Рисунок 1 — блок-схема алгоритма

Из задачи можно вывести формулу, которая считает количество монет после Z переходов:

2Z⋅N−K⋅(2Z−1)

Выражаем К - количество монет, которые чёрт забирает после каждого перехода:

K=(2Z⋅N)/(2Z−1)

К - количество монет, следовательно, число целое. Чтобы это условие выполнялось, 2Z⋅N должно нацело делиться на 2Z−1, следовательно Nдолжно нацело делиться на 2Z−1.

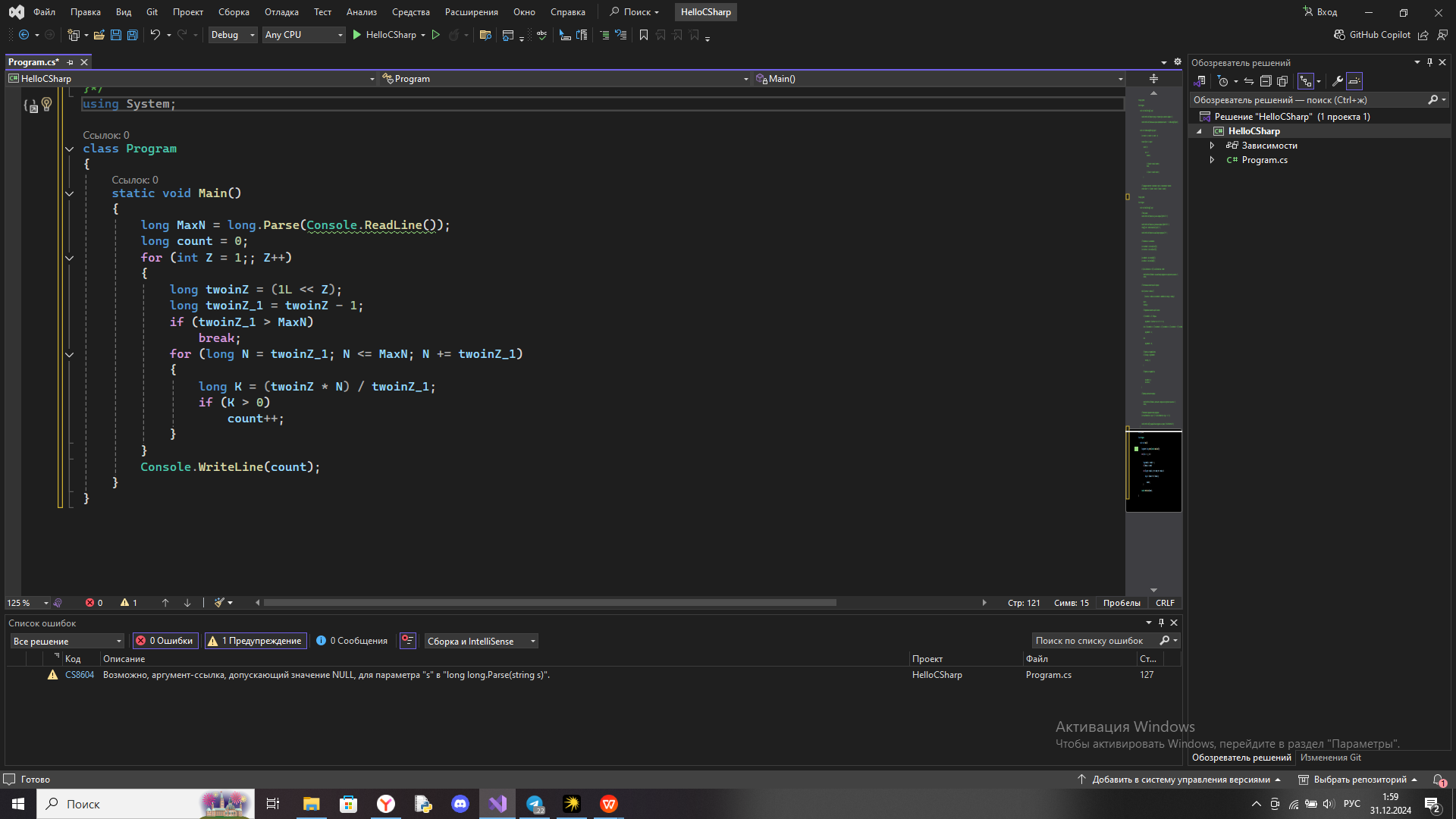


Рисунок 2 ­­­— Код программы к задаче 1

2024-12-31_02-02-56

Рисунок 3 — Первый тест задачи 1

2024-12-31_02-07-07

Рисунок 4 — Второй тест задачи 1

2024-12-31_02-05-44

Рисунок 5 — Третий тест задачи 1

# **Задача 2** – **Упаковки молока**

# **Постановка задачи 2**

На оптовой базе имеется молоко, выпущенное несколькими фирмами.

Молоко каждой фирмы расфасовано в два вида упаковок, представляющих собой параллелепипеды. Для каждого вида упаковки каждой из фирм известна стоимость, которая включает как стоимость материала тары, так и стоимость собственно молока.

Требуется определить фирму, у которой стоимость одного литра собственно молока минимальна, а также эту стоимость.

Примечание

Считать, что материал тары абсолютно тонкий и все плоскости параллелепипеда состоят из одного слоя материала.

Считать, что у двух упаковок одной фирмы стоимость единицы площади материала одинакова.

Считать, что у двух упаковок одной фирмы стоимость одного литра собственно молока одинакова.

Входной файл

Первая строка содержит целое число N - количество фирм (1 <= N <= 100).

Следующие N строк содержат шесть целых чисел Xi1, Yi1, Zi1, Xi2, Yi2, Zi2 - размеры двух видов упаковок i-ой фирмы в сантиметрах (0 < Xi1, Yi1, Zi1, Xi2, Yi2, Zi2 <= 100; 1 <= i <= N), а также два вещественных числа Ci1 и Ci2 - стоимости первой и второй упаковок соответственно у i-ой фирмы в рублях (0 < Ci1, Ci2 <= 1000.0). В стоимости упаковок включаются как стоимость материала тары, так и стоимость собственно молока.

Выходной файл

Должен содержать одну строку, состоящую из целого и вещественного чисел, разделенных пробелом - номер фирмы, у которой стоимость одного литра собственно молока минимальна, а также эту стоимость в рублях (стоимость выводить с двумя знаками после запятой).

Если имеется несколько фирм с одинаковой минимальной стоимостью собственно молока, то вывести ту из них, номер которой минимален.

**Ход решения задачи 2**

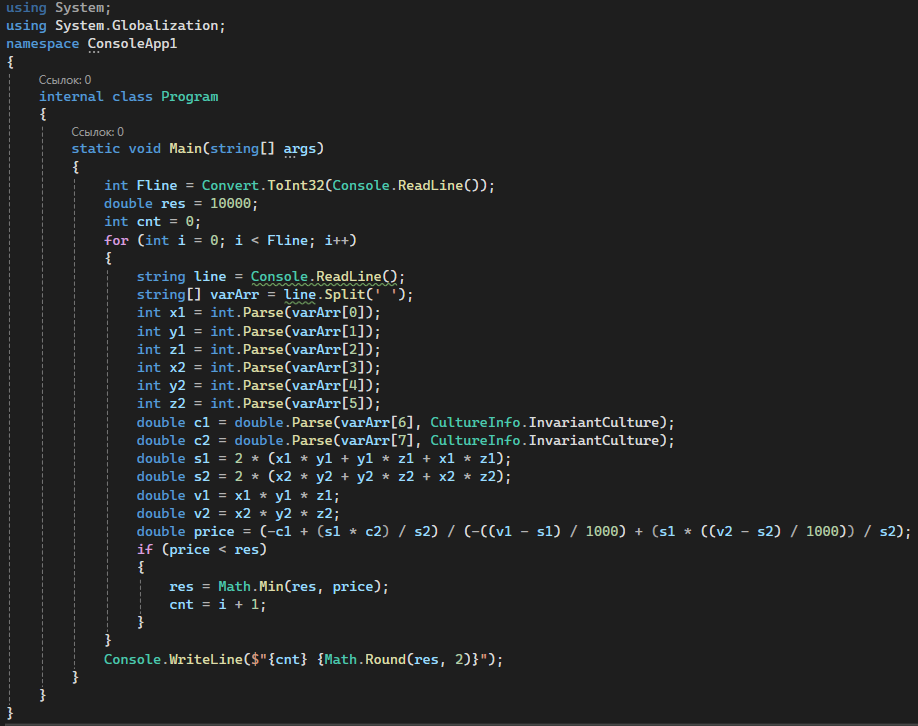


Рисунок 6 ­­­— Код к задаче 2

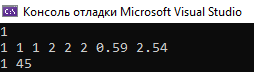


Рисунок 7 — Первый тест программы к задаче 2

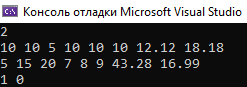


Рисунок 8 — Второй тест программы к задаче 2

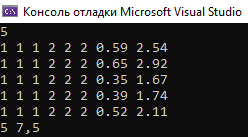


Рисунок 9 — Третий тест программы к задаче 2

# **Задача 3** – **Постройка дома**

# **Постановка задачи 3**

На месте разрушенного дома, от которого уцелела одна стена, необходимо построить новый дом с основанием в виде прямоугольника размером X х Y м. Длина уцелевшей стены - L м. При постройке дома следует учитывать, что:

* ремонт погонного метра уцелевшей стены, для использования ее в новом доме, обходится в C1 руб.;
* разбор погонного метра уцелевшей стены обходится в С2 руб.;
* строительство погонного метра из материала, полученного при разборе стены, обходится в С3 руб.;
* строительство погонного метра из нового материала обходится в С4 руб.;
* стоимость погонного метра нового материала составляет С5 руб.;
* вывоз на свалку погонного метра материала, образованного при разборе стены, обходится в С6 руб.;
* требуется определить минимальную сумму (в руб.), которую необходимо потратить при строительстве дома на постройку стен.

Примечание:

* после строительства на стройке не должно остаться мусора в виде неиспользуемого куска старой стены и каких-либо неиспользуемых материалов.;
* считать стены абсолютно тонкими.

Входной файл содержит строку, состоящую из девяти целых чисел X, Y, L, С1, C2, С3, С4, С5, С6, разделенных пробелами (1 ≤ X, Y, L, С1, С2, С3, С4, С5, С6 ≤ 100).

Выходной файл должен одно целое число - минимальную сумму (в руб.), которую необходимо потратить на строительство стен нового дома.

# **Ход решения задачи 3**

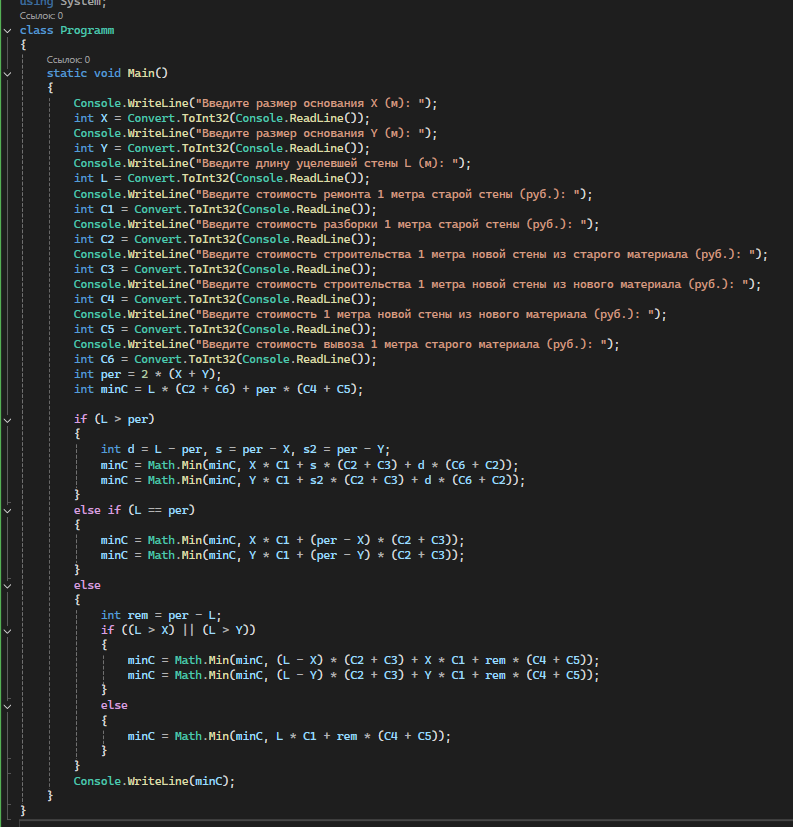


Рисунок 10 — Код программы к задаче 3

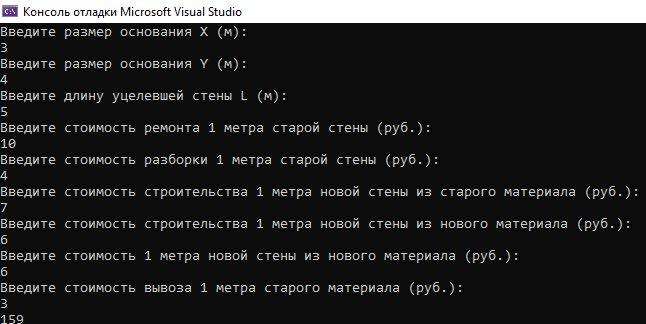


Рисунок 11 — Первый тест программы к задаче 3

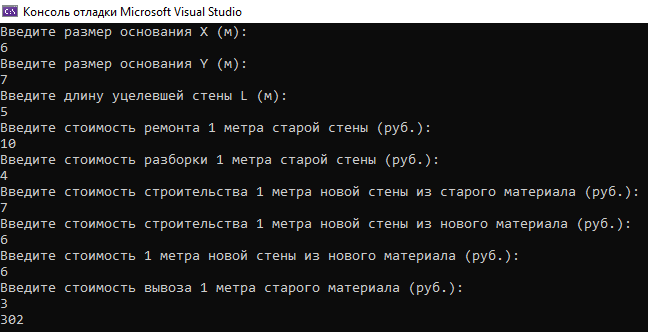


Рисунок 12 — Второй тест программы к задаче 3

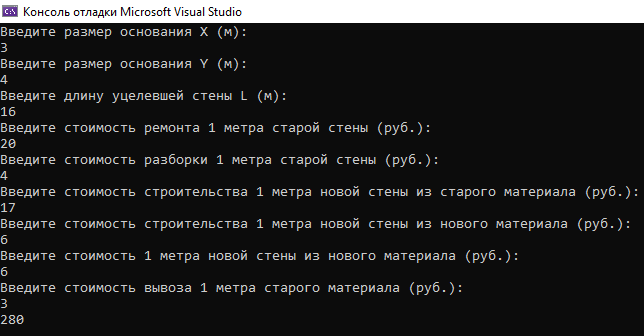


Рисунок 13 — Третий тест программы к задаче 3

# **Задача 4** –**Интенсификация производства**

# **Постановка задачи 4**

Перед коллективом предприятия “Ни шагу назад” была поставлена задача наращивать каждый день производство продукции на 1.

Требуется определить, какой суммарный объем продукции будет выпущен предприятием за заданный период, если в первый день периода предприятие выпускало P единиц продукции.

Примечания:

период задается в виде двух календарных дат;

длительность периода лежит в диапазоне от 1 до 60000;

високосные годы учитываются по упрощенному правилу: високосным считается год, делящийся нацело на 4;

день начала периода и день его окончания учитываются при подсчете суммарного объема продукции и длительности периода;

все даты заданы корректно.

Входной файл содержит:

в первой строке – дата начала периода в формате ДД.ММ.ГГГГ;

во второй строке – дата окончания периода в формате ДД.ММ.ГГГГ;

в третьей строке целое число – начальный выпуск продукции P (0 ≤ P ≤ 5000).

Выходной файл должен содержать суммарный объем продукции.

### Ход решения задачи 4

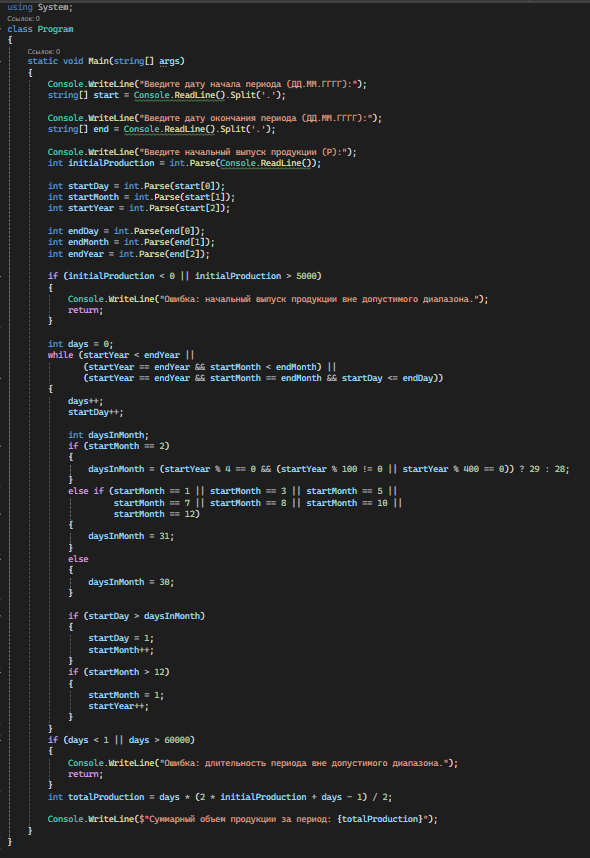


Рисунок 14 — Код задачи 4

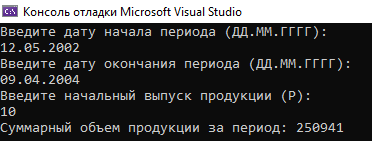


Рисунок 15 — Первый тест программы к задаче 4

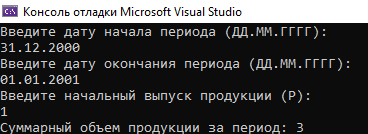


Рисунок 16 — Второй тест программы к задаче 4

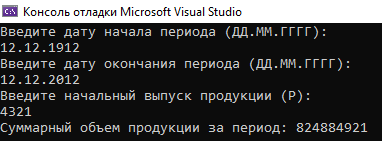


Рисунок 17 — Третий тест программы к задаче 4

# **Заключение**

В ходе выполнения расчётно-графической работы на тему олимпиадных задач по программированию на языке C# были использованы различные алгоритмы, условные операторы, циклы, а также изучены принципы работы с переменными, способы вводы и вывода информации. Также расчётно-графиеская работа помогла улучшить навыки программирования на C#, развить логическое мышление. Этот опыт определённо поможет в будущем.

# **Список используемой литературы**

1. <https://dotnet.microsoft.com/ru-ru/languages/csharp> (дата обращения 30.12.2024).
2. Васильев А.Н. Программирование на C# для начинающих – Москва, 2019 – 860 с. (дата обращения 10.12.2024).
3. Троелсен Э., Джапура К. Язык программирования C# 10 и платформа .NET 6. – Питер, Санкт-Петербург, 2022. – 960 с (дата обращения 10.12.2024)
4. <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (Дата обращения: 12.12.2024)
5. https://timeweb.com/ru/community/articles/chto-takoe-csharp (дата обращения 24.12.2024).