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–№ 28 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Шатилова Ивана Романовича | | | | | |
|  |  |  |  | | 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](#_Toc185878851)

[Постановка задачи «Упаковки молока» 4](#_Toc185878852)

[Ход решения задачи «Упаковки молока» 5](#_Toc185878853)

[Постановка задачи «Отбор в разведку» 9](#_Toc185878854)

[Ход решения задачи «Отбор в разведку» 10](#_Toc185878855)

[Постановка задачи «Интенсификация производства» 12](#_Toc185878856)

[Ход решения задачи «Интенсификация производства» 13](#_Toc185878857)

[Постановка задачи «Золотая рыбка» 17](#_Toc185878858)

[Ход решения задачи «Золотая рыбка» 19](#_Toc185878859)

[Заключение 23](#_Toc185878860)

[Список используемой литературы 24](#_Toc185878861)

**Введение**

C# — современный объектно-ориентированный язык программирования, разработанный компанией Microsoft. Язык используется для создания различных приложений, включая веб-приложения, настольные программы и игры. C# поддерживает парадигмы программирования, такие как функциональное и асинхронное программирование.

Основные особенности языка включают строгую типизацию, автоматическое управление памятью с помощью сборщика мусора и богатую стандартную библиотеку. C# активно используется в разработке на платформе .NET, что обеспечивает кроссплатформенность и интеграцию с различными сервисами.

Язык также поддерживает современные подходы к разработке, такие как использование LINQ для работы с данными и асинхронные методы для повышения производительности. C# является популярным выбором для разработчиков благодаря своей простоте, мощным возможностям и активному сообществу.

В данной расчетно-графической работе язык C# будет использован с целью решения четырех олимпиадных задач.

## **Постановка задачи «Упаковки молока»**

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 «Упаковки молока»**

Исходя из условия задачи, для определения минимальной стоимости литра молока у некоторой фирмы, нам нужно вычислить стоимость молока каждой фирмы. Мы знаем, что стоимость упаковки молока складывается из стоимости самого молока и стоимости тары.

Для каждой фирмы получим размеры двух видов упаковок (X1, Y1, Z1 и X2, Y2, Z2 соответственно) и их стоимости (C1 и C2) из консольного ввода.

Вычислим объем первой и второй упаковки: V1 = X1 \* Y1 \* Z1, V2 = X2 \* Y2 \* Z2. Также вычислим площади поверхности первой и второй упаковки: S1 = 2(X1 \* Y1 + Y1 \* Z1 + X1 \* Z1), S2 = 2(X2 \* Y2 + Y2 \* Z2 + X2 \* Z2).

Составим систему уравнений цен упаковок молока: , где q — стоимость одного сантиметра упаковки, а p — стоимость одного литра молока. Выразив q из первой строки и подставив его во вторую строку, выполним дальнейшие преобразования и получим итоговую формулу:

Следует уточнить, что мы подсчитываем стоимость молока в литрах, но считали размеры упаковок в сантиметрах, поэтому итоговое значение необходимо умножить на 1000.

Данные подсчеты в цикле повторяются то количество раз, сколько имеется различных фирм. Причем если упаковка молока n-ой фирмы дешевле, чем текущая минимальная стоимость, то устанавливается новая минимальная стоимость, пока не будет найдена самая маленькая.

В результате в консоль будет выведена минимальная стоимость молока за литр, а также номер фирмы, обладающей минимальной стоимостью упаковки.

Составим блок-схему алгоритма:

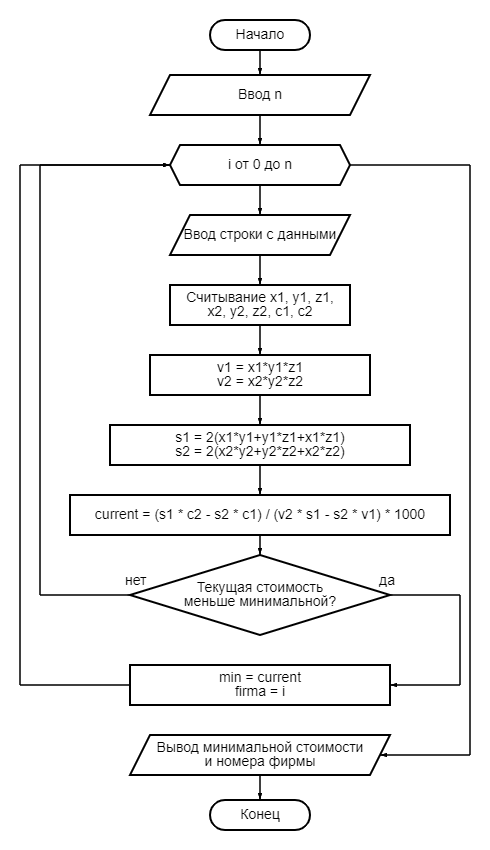


Рисунок 1 – Блок-схема алгоритма задачи 1

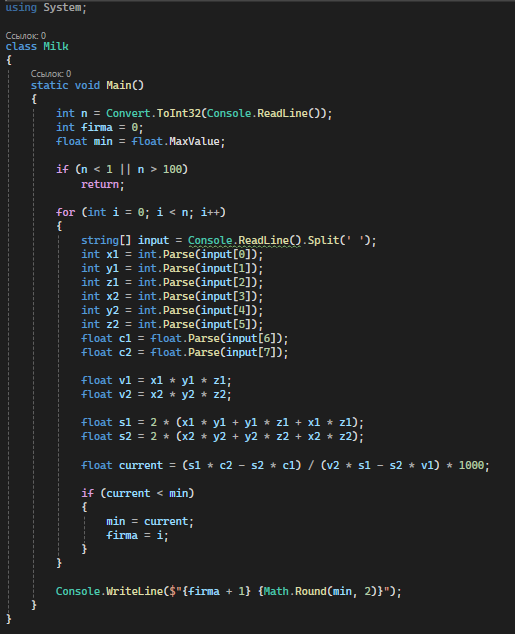


Рисунок 2 – Код решения задачи 1

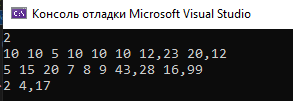


Рисунок 3 – Первый тест задачи 1

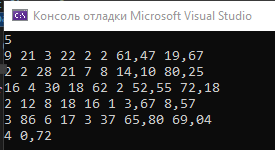
****

Рисунок 4 – Второй тест задачи 1

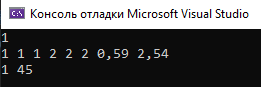


Рисунок 5 – Третий тест задачи 1

## **Постановка задачи «Отбор в разведку»**

Из **N** солдат, выстроенных в шеренгу, требуется отобрать троих в разведку. Для того чтобы сделать это, выполняется следующая операция: если солдат в шеренге больше 3, то шеренга разбивается на две, одна из которых состоит из солдат, стоящие на четных позициях, а вторая – стоящих на нечетных позициях. Эта процедура повторяется для всех полученных шеренг до тех пор, пока в каждой из них не останется 3 или менее солдат. Если солдат осталось трое, то данную группу можно послать в разведку.

Требуется определить, сколько групп по 3 человека может быть сформировано из исходной шеренги.

**Входной файл** содержит число **N –** количество солдат в исходной шеренге. (0 < **N** ≤ 10000000).

**Выходной файл** должен содержать количество вариантов формирования групп разведки.

## **Ход решения задачи «Отбор в разведку»**

Чтобы решить задачу о формировании групп из трех солдат, можно использовать рекурсивный подход. Будем разбивать шеренгу на две подгруппы (четные и нечетные позиции) до тех пор, пока в каждой из них не останется не более трех солдат. После этого будем подсчитывать количество групп по три человека.

Код программы:

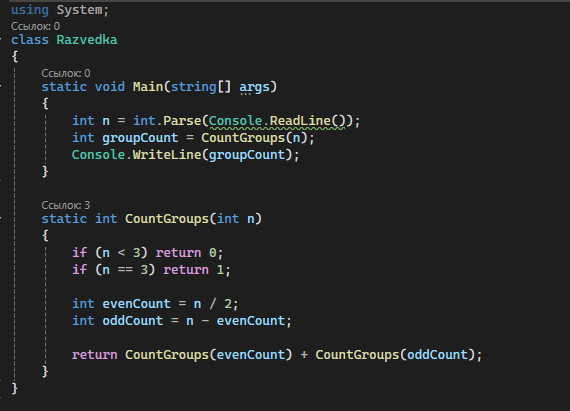


Рисунок 6 – Код решения задачи 2

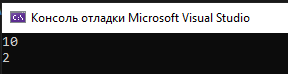


Рисунок 7 – Первый тест задачи 2

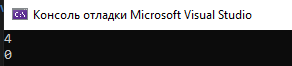


Рисунок 8 – Второй тест задачи 2

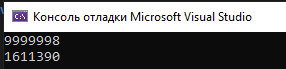


Рисунок 9 – Третий тест задачи 2

## **Постановка задачи «Интенсификация производства»**

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 «Интенсификация производства»**

Для решения данной задачи нам нужно вычислить разницу между двумя данными датами. Суть алгоритма заключается в том, что для каждой даты будет посчитано абсолютное количество дней (то есть начиная с 00.00.0000), учитывая високосные года. Посчитав абсолютное количество дней для обоих дат, количество дней между двумя датами находится как модуль разности количества дней. Затем при помощи цикла находится итоговое значение суммарного объема продукции.

Код программы:

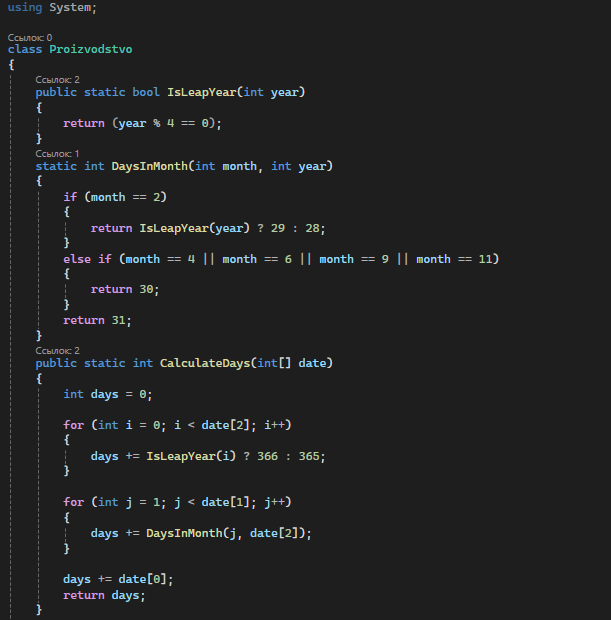


Рисунок 10 – Код решения задачи 3

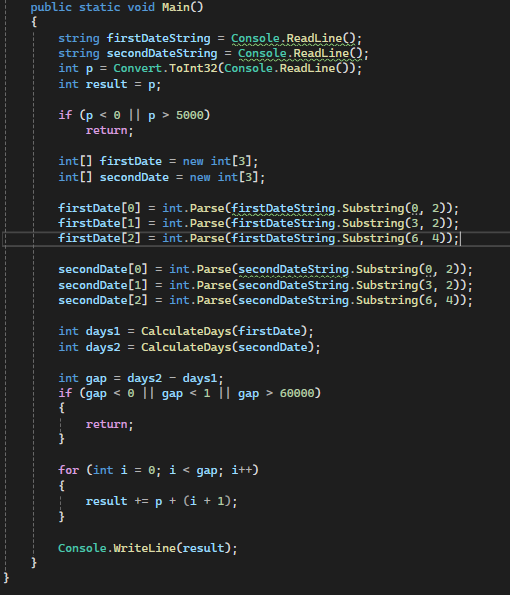


Рисунок 11 – Код решения задачи 3 (продолжение)

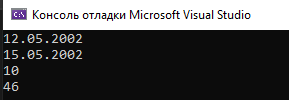


Рисунок 12 – Первый тест задачи 3

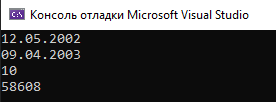


Рисунок 13 – Второй тест задачи 3

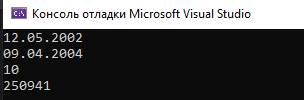


Рисунок 14 – Третий тест задачи 3

## **Постановка задачи «Золотая рыбка»**

*Жил старик со своею старухой у самого синего моря;  
Они жили в ветхой землянке ровно тридцать лет и три года.  
Старик ловил неводом рыбу, старуха пряла свою пряжу.  
Раз он в море закинул невод, - пришел невод с одною тиной.  
Он в другой раз закинул невод, - пришел невод с травою морскою.  
В третий раз закинул он невод, - пришел невод с одною рыбкой,  
С непростою рыбкой, - золотою.   
Как взмолится золотая рыбка! Голосом молвит человечьим:  
“Отпусти ты, старче, меня в море! Дорогой за себя дам откуп:***\***Дам я тебе список предлинный, все волшебными словами заполненный.  
Как скажешь ты слово волшебное - вмиг желанье твое и исполниться.  
Только есть здесь одно обстоятельство, для данного часа существенное.  
Все слова в данный час начинаться должны на буквы определенные.  
И заканчиваться они также должны на буквы отдельно заданные.  
А для каждой из букв указанных есть ограничение по использованию.  
Сами буквы же и все ограничения в списке слов также указаны.”  
Вот стоит теперь старик и думает: сколько желаний он может выполнить  
Если каждое из слов указанных можно только один раз использовать.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**\*** *Заимствовано у А.С. Пушкина*

**Примечание:**

* слова состоят только из маленьких английских букв;
* все слова уникальны.

**Входной файл**

Первая строка содержит целое число **N** - общее количество волшебных слов (1  **N**  30000).

Следующие **N** строк содержат по одному слову. Длина слова не превышает 15 символов.

Следующая строка содержит целое число **F** - количество букв, на которые могут начинаться искомые волшебные слова (1  **F**  26).

Следующие **F** строк содержат описания ограничений на каждую букву в формате “**b** **k**”, где **b** - допускаемая в использовании буква; **k** - количество слов, начинающихся на эту букву, которые можно использовать (1  **k**  1000).

Следующая строка содержит целое число **L** - количество букв, на которые могут заканчиваться искомые волшебные слова (1  **L**  26).

Следующие **L** строк содержат описания ограничений на каждую букву в формате “**b** **k**”, где **b** - допускаемая в использовании буква; **k** - количество слов, заканчивающихся на эту букву, которые можно использовать (1  **k**  1000).

**Выходной файл** должен содержать одно целое число - количество желаний, которые может выполнить старик.

## **Ход решения задачи «Золотая рыбка»**

Алгоритм решения задачи заключается в том, что программа проходит по каждому слову и проверяет, удовлетворяет ли оно условиям (начинается на одну из допустимых букв и заканчивается на другую). Если слово соответствует условиям, итоговый счетчик увеличивается, и количество доступных слов для данной буквы уменьшается. Цикл повторяется, пока остаются непроверенные слова.

Код программы:

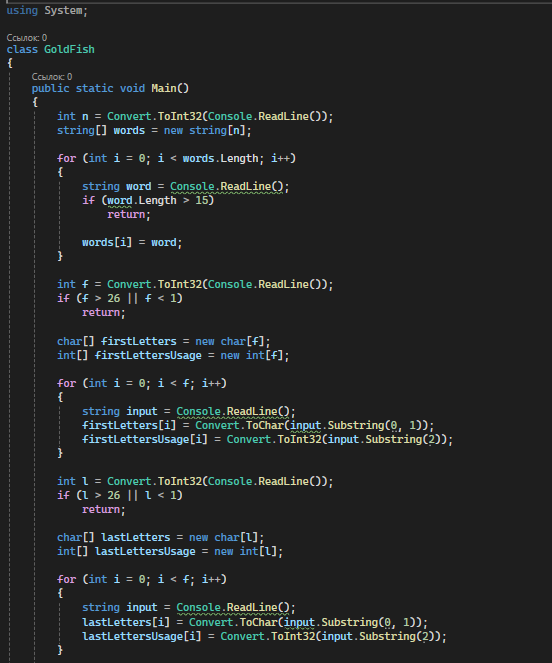


Рисунок 15 – Код решения задачи 4

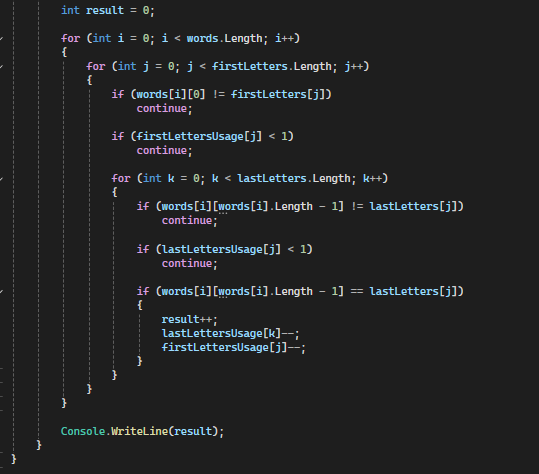


Рисунок 16 – Код решения задачи 4 (продолжение)

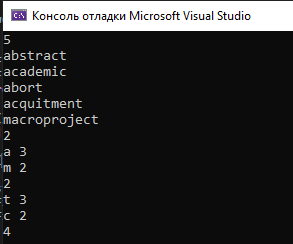


Рисунок 17 – Первый тест задачи 4

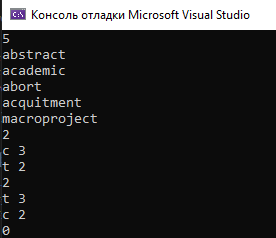


Рисунок 18 – Второй тест задачи 4

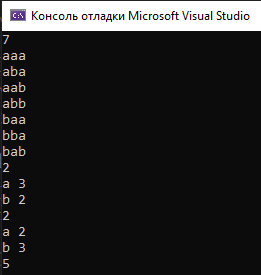


Рисунок 19 – Третий тест задачи 4

## **Заключение**

В ходе выполнения расчетно-графической работы по дисциплине «Алгоритмизация и программирование» были изучены синтаксис и основные концепции языка, такие как переменные, типы данных, операторы, условные конструкции и циклы. Это дало понимание базовых принципов программирования в C#.

Выполнение работы позволило не только закрепить теоретические знания, но и приобрести практические навыки программирования на языке C#. Полученные знания и опыт будут полезны при дальнейшем изучении программирования и разработке более сложных проектов.

## **Список используемой литературы**

1. Документация Microsoft C# <https://learn.microsoft.com/ru-ru/dotnet/csharp/> дата обращения 20.12.2024.
2. Stack Overflow <https://stackoverflow.com> дата обращения 19.12.2024
3. Metanit <https://metanit.com/sharp/tutorial/> дата обращения 20.12.2024
4. Habr <https://habr.com> дата обращения 20.12.2024
5. LearnCS <https://learncs.org/> дата обращения 20.12.2024