МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №3**

по дисциплине: Основы программирования

тема: «Алгоритмы разветвляющейся структуры»

Выполнил: ст. группы ПВ-223

Пахомов Владислав Андреевич

Проверили:

Притчин Иван Сергеевич

Черников Сергей Викторович

Код-ревьер: ст. группы ПВ-223

Голуцкий Георгий Юрьевич

Белгород 2022 г.

**Лабораторная работа № 3**

**Содержание отчёта:**

* Тема лабораторной работы
* Цель лабораторной работы
* Решения задач
  + Название задачи
  + Исходный код
  + Скриншот с codeforces с указанием вердикта тестирующей системы
* Код-ревью
* Работа над ошибками (код-ревью)
* Вывод

**Тема лабораторной работы:** Алгоритмы разветвляющейся структуры

**Цель лабораторной работы:** закрепление навыков написания линейных и разветвляющихся алгоритмов.

**Решения задач:**

1. Водяная лилия (1199B)

#include <stdio.h>  
  
int main() {  
 long long heightAboveWaterLevel, flowerSailDistance;  
 scanf("%lld %lld", &heightAboveWaterLevel, &flowerSailDistance);  
  
 double lakeDepth = (flowerSailDistance \* flowerSailDistance –

heightAboveWaterLevel \* heightAboveWaterLevel) /  
 (2.0 \* heightAboveWaterLevel);  
  
 printf("%.13f", lakeDepth);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1199B.c>



1. Даша и лестница (761A)

#include <stdio.h>  
#include <math.h>  
  
int main() {  
 int evenStairs, oddStairs;  
 scanf("%d %d", &evenStairs, &oddStairs);  
  
 if (abs(oddStairs - evenStairs) <= 1 && evenStairs + oddStairs)  
 printf("YES");  
 else  
 printf("NO");  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/761A.c>



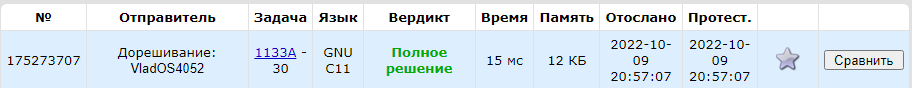
1. Середина контеста (1133A)

#include <stdio.h>  
  
#define **MINUTES\_IN\_HOUR** 60  
  
int main() {  
 int contestBeginHours, contestBeginMinutes, contestEndHours, contestEndMinutes;  
 scanf("%d:%d %d:%d", &contestBeginHours, &contestBeginMinutes, &contestEndHours,

&contestEndMinutes);  
  
 int contestBeginInMinutes = contestBeginHours \* **MINUTES\_IN\_HOUR** +

contestBeginMinutes;  
 int contestEndInMinutes = contestEndHours \* **MINUTES\_IN\_HOUR** + contestEndMinutes;  
 int contestMiddleInMinutes = (contestBeginInMinutes + contestEndInMinutes) / 2;  
 int contestMiddleHours = contestMiddleInMinutes / **MINUTES\_IN\_HOUR**;  
 int contestMiddleMinutes = contestMiddleInMinutes % **MINUTES\_IN\_HOUR**;  
  
 printf("%02d:%02d", contestMiddleHours, contestMiddleMinutes);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1133A.c>

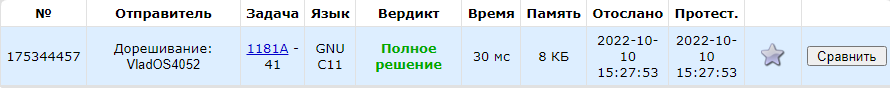


1. Чунга-Чанга (1181A)

#include <stdio.h>  
  
long long max(long long a, long long b) {  
 return a > b ? a : b;  
}  
  
int main() {  
 long long chizhiksSasha, chizhicksMasha, coconutPrice;  
 scanf("%lld %lld %lld", &chizhiksSasha, &chizhicksMasha, &coconutPrice);  
  
 long long coconuts = chizhiksSasha / coconutPrice + chizhicksMasha /

coconutPrice;  
 long long changeSasha = chizhiksSasha % coconutPrice;  
 long long changeMasha = chizhicksMasha % coconutPrice;  
 long long totalChange = changeMasha + changeSasha;  
  
 long long changeTransfer;  
 if (totalChange >= coconutPrice) {  
 coconuts++;  
 changeTransfer = coconutPrice - max(changeMasha, changeSasha);  
 } else  
 changeTransfer = 0;  
  
 printf("%lld %lld", coconuts, changeTransfer);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1181A.c>



1. Отопление (1260A)

#include <stdio.h>  
  
long long pow2(long long num) {  
 return num \* num;  
}  
  
int main() {  
 int taskAmount;  
 scanf("%d", &taskAmount);  
  
 for (int i = 0; i < taskAmount; i++) {  
 int allowedRadiators, requiredRadiatorSections;  
 scanf("%d %d", &allowedRadiators, &requiredRadiatorSections);  
  
 int minRadiatorLen = requiredRadiatorSections / allowedRadiators;  
 int maxRadiatorLen = minRadiatorLen + 1;  
 int maxRadiatorAmount = requiredRadiatorSections % allowedRadiators;  
 int gurgles = pow2(minRadiatorLen) \* (allowedRadiators –

maxRadiatorAmount) +  
 pow2(maxRadiatorLen) \* maxRadiatorAmount;  
  
 printf("%d\n", gurgles);  
 }  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1260A.c>



1. Комментаторские кабинки (990A)

#include <stdio.h>  
  
long long min(long long a, long long b) {  
 return a < b ? a : b;  
}  
  
int main() {  
 long long initialCommentaryBoxes, delegations, feeToBuild, feeToDemolish;  
 scanf("%lld %lld %lld %lld", &initialCommentaryBoxes, &delegations, &feeToBuild,

&feeToDemolish);  
  
 long long int extraCommentaryBoxes = initialCommentaryBoxes % delegations;  
 long long minFee = min(feeToDemolish \* extraCommentaryBoxes,  
 feeToBuild \* (delegations - extraCommentaryBoxes));  
  
 printf("%lld", minFee);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/990A.c>



1. Пересдача (991A)

#include <stdio.h>  
  
#define **MIN\_STUDENTS\_DIDNT\_PASS\_EXAMS** 1  
  
int main() {  
 int visitorsBugDonalds, visitorsBeaverKing, visitorsBothRestaurants,

totalStudents;  
 scanf("%d %d %d %d", &visitorsBugDonalds, &visitorsBeaverKing,

&visitorsBothRestaurants, &totalStudents);  
  
 int studentsDidntPassExams = totalStudents - visitorsBugDonalds –

visitorsBeaverKing + visitorsBothRestaurants;  
  
 if (studentsDidntPassExams < **MIN\_STUDENTS\_DIDNT\_PASS\_EXAMS** ||

visitorsBugDonalds < visitorsBothRestaurants ||  
 visitorsBeaverKing < visitorsBothRestaurants)  
 studentsDidntPassExams = -1;  
  
 printf("%d", studentsDidntPassExams);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/991A.c>



1. Высота функции (1036A)

#include <stdio.h>  
  
long long ceilFrac(long long a, long long b) {  
 return a % b ? a / b + 1 : a / b;  
}  
  
int main() {  
 long long points, square;  
 scanf("%lld %lld", &points, &square);  
  
 long long maxHeight = ceilFrac(square, points);  
  
 printf("%lld", maxHeight);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1036A.c>



1. Пара игрушек (1023B)

#include <stdio.h>  
  
long long min(long long a, long long b) {  
 return a < b ? a : b;  
}  
  
int main() {  
 long long totalToys, gurgles;  
 scanf("%lld %lld", &totalToys, &gurgles);  
  
 long long middleUpperPrice = gurgles / 2 + 1;  
  
 long long variationsToBuyToys;  
 if (middleUpperPrice > totalToys)  
 variationsToBuyToys = 0;  
 else {  
 long long biggestUpperPrice = min(totalToys, gurgles - 1);  
 variationsToBuyToys = biggestUpperPrice - middleUpperPrice + 1;  
 }  
  
 printf("%lld", variationsToBuyToys);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1023B.c>



1. Посмотрим футбол (195A)

#include <stdio.h>  
  
long long ceilFrac(long long a, long long b) {  
 return a % b ? a / b + 1 : a / b;  
}  
  
int main() {  
 int videoDataPerSecond, internetTrafficSpeed, videoTime;  
 scanf("%d %d %d", &videoDataPerSecond, &internetTrafficSpeed, &videoTime);  
  
 int waitTime = ceilFrac(videoTime \* (videoDataPerSecond - internetTrafficSpeed),

internetTrafficSpeed);  
  
 printf("%d", waitTime);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/195A.c>



11. Кто напротив? (1560B)

#include <stdio.h>  
#include <math.h>  
  
int main() {  
 int taskAmount;  
 scanf("%d", &taskAmount);  
  
 for (int i = 0; i < taskAmount; i++) {  
 int watcherA, watcherB, watcherC;  
 scanf("%d %d %d", &watcherA, &watcherB, &watcherC);  
  
 int expectedHalfTotalWatchers = abs(watcherA - watcherB);  
 int variantWatcherD1 = watcherC - expectedHalfTotalWatchers;  
 int variantWatcherD2 = watcherC + expectedHalfTotalWatchers;  
  
 int watcherD;  
 if (watcherA > 2 \* expectedHalfTotalWatchers || watcherB > 2 \*

expectedHalfTotalWatchers ||  
 watcherC > 2 \* expectedHalfTotalWatchers)  
 watcherD = -1;  
 else if (variantWatcherD1 >= 1 && variantWatcherD1 <= 2 \*

expectedHalfTotalWatchers)  
 watcherD = variantWatcherD1;  
 else if (variantWatcherD2 >= 1 && variantWatcherD2 <= 2 \*

expectedHalfTotalWatchers)  
 watcherD = variantWatcherD2;  
 else  
 watcherD = -1;  
  
 printf("%d\n", watcherD);  
 }  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1560B.c>



12. Старт олимпиады (1539A)

#include <stdio.h>  
  
long long min(long long a, long long b) {  
 return a < b ? a : b;  
}  
  
long long max(long long a, long long b) {  
 return a < b ? b : a;  
}  
  
int main() {  
 int taskAmount;  
 scanf("%d", &taskAmount);  
  
 for (int i = 0; i < taskAmount; i++) {  
 long long totalParticipants, contestInterval, contestWorkingTime;  
 scanf("%lld %lld %lld", &totalParticipants, &contestInterval,

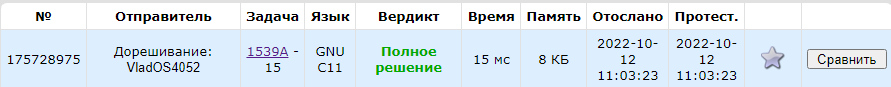
&contestWorkingTime);  
  
 long long maxUnsatisfiedLevel = contestWorkingTime / contestInterval;  
 long long annoyingParticipants = min(totalParticipants,

maxUnsatisfiedLevel);  
 long long totalUnsatisfiedLevel = ((annoyingParticipants - 1) \*  
 annoyingParticipants) / 2 +

maxUnsatisfiedLevel \*  
 max(totalParticipants –

maxUnsatisfiedLevel, 0);  
  
 printf("%lld\n", totalUnsatisfiedLevel);  
 }  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1539A.c>



13. Настя играет в компьютер (1136B)

#include <stdio.h>  
  
long long min(long long a, long long b) {  
 return a < b ? a : b;  
}  
  
int main() {  
 int hatchAmount, position;  
 scanf("%d %d", &hatchAmount, &position);  
  
 int stepsAmount = min(hatchAmount - position, position - 1) + 3 \* hatchAmount;  
  
 printf("%d", stepsAmount);  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1136B.c>



14. Странная таблица (1506A)

#include <stdio.h>  
  
int main() {  
 int taskAmount;  
 scanf("%d", &taskAmount);  
  
 for (int i = 0; i < taskAmount; i++) {  
 long long tableHeight, tableWidth, numInTable;  
 scanf("%lld %lld %lld", &tableHeight, &tableWidth, &numInTable);  
  
 long long row = (numInTable % tableHeight - 1 + tableHeight) % tableHeight;  
 long long column = ((numInTable - 1 + tableHeight) / tableHeight);  
 long long resultNum = row \* tableWidth + column;  
  
 printf("%lld\n", resultNum);  
 }  
  
 return 0;  
}

Ссылка на репозиторий: <https://github.com/IAmProgrammist/programming-and-algorithmization-basics/blob/c/lab3/1506A.c>



**Код-ревью:**

Ошибок не найдено

**Работа над ошибками (код-ревью):**

Ошибок не исправлено

**Вывод:** в ходе выполнения лабораторной работы закреплены навыки написания линейных и разветвляющихся алгоритмов.