

Петрозаводский государственный университет
Институт математики и информационных технологий

09.03.04 - Программная инженерия

Отчет о прохождении производственной практики

Выполнил студент группы 22207:

Чекарев Лев Олегович

Направление подготовки:

Системное и прикладное программное обеспечение

Место прохождения практики:

Кафедра информатики и математического обеспечения

Сроки прохождения практики:

29.05.23-09.06.23

Руководитель:

Богоявленская Ольга Юрьевна, к.т.н., доцент

Оценка _____

Дата _____

Содержание

Введение	3
1 Интенсивы	3
1.1 Дивизионы алгоритмического программирования	3
1.2 Прохождение интенсивов	4
2 Чемпионат	4
2.1 Формат чемпионата	4
2.2 Задача А. Анализ хода матча	4
2.2.1 Формулировка задачи	4
2.2.2 Идея алгоритма	5
2.2.3 Решение задачи	6
2.2.4 Результат решения задачи	7
2.3 Задача С. Выделяем римские анаграммы	7
2.3.1 Формулировка задачи	7
2.3.2 Идея алгоритма	9
2.3.3 Решение и результат задачи	9
2.4 Задача F. Если лодку украли...	9
2.4.1 Формулировка задачи	9
2.4.2 Идея алгоритма	11
2.4.3 Решение задачи	11
2.4.4 Результат решения задачи	12
2.5 Задача J. Играем в угадайку?	12
2.5.1 Формулировка задачи	12
2.5.2 Идея алгоритма	14
2.5.3 Решение задачи	14
2.5.4 Результат решения задачи	16
2.6 Задача L. Крутые номера	17
2.6.1 Формулировка задачи	17
2.6.2 Идея алгоритма	18
2.6.3 Решение задачи	18
2.6.4 Результат решения задачи	20

Введение

Цель практики: развитие навыков решения олимпиадных задач в рамках чемпионата по алгоритмическому программированию RuCode.

Задачи учебной практики:

1. Пройти онлайн интенсивы по спортивному программированию дивизиона E и изучить теоритический материал по алгоритмическому программированию;
2. Используя полученные теоритические знания, выполнить задания чемпионата.

План прохождения практики:

В рамках "Всероссийского учебного фестиваля по искусственному интеллекту и программированию RuCode" проводились интенсивы и чемпионат по двум направлениям: искусственный интеллект и алгоритмическое программирование (далее АП). Мной было принято участие во направлении АП.

В течение 5 дней проводились онлайн интенсивы по АП. В установленную дату, 21.05.2023, был проведен чемпионат по АП. Одной из площадок для его проведения был Петрозаводский государственный университет, где мы с командой из 3-х человек приняли участие. Чемпионат проходил в течение 5 астрономических часов.

Наша команда «Имитаторы» организовала работу по достижению целей и выполнению задач, разделив рабочие обязанности между всеми членами команды:

1. Поиск и проектирование алгоритма для решения конкретной задачи;
2. Программная реализация алгоритма.

1 Интенсивы

1.1 Дивизионы алгоритмического программирования

Обучение на интенсивах разделялось на 4 дивизиона:

1. Дивизион C;

2. Дивизион D;
3. Дивизион E;
4. Дивизион F.

Сложность алгоритмов, проходимых в рамках дивизиона, по убывающей (дивизион C - самый сложный, дивизион F - самый простой).

1.2 Прохождение интенсивов

Возможность проходить интенсивы в течение 5 дней была как дистанционная, так и очная, в стенах ПетрГУ. Мной был выбран дистанционный вариант.

На основании анализа своих навыков в алгоритмическом программировании был выбран дивизион E.

2 Чемпионат

2.1 Формат чемпионата

Чемпионат проводился в различных площадках городов России, одной из площадок был Петрозаводский Государственный университет, где мы приняли участие. Участвовали в чемпионате дивизиона C-F.

Чемпионат проходил 21.05.2023 10:00-15:00.

Все программы нашей команды написаны на языке C++.

2.2 Задача A. Анализ хода матча

2.2.1 Формулировка задачи

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

В спин-оффе популярного фильма Алиса играет шахматный матч с Бобом. Матч состоит из нескольких игр и начинается со счёта 0:0. Если какой-то игрок выигрывает игру, к её/его баллам прибавляется 1, в противном случае (в случае ничьей) каждый игрок

получает по 0.5 балла за эту партию. Вам дан текущий счёт. Найдите общее количество сыгранных партий, а также наименьшее и наибольшее возможное количество ничьих в матче.¹

Формат входных данных

Первая строка входных данных содержит одно вещественное число A — количество баллов, набранных Алисой. Вторая строка содержит одно целое число B — количество баллов, набранных Бобом ($0 \leq A, B \leq 6$). Числа даны без лишних знаков (то есть без лишних нулей после десятичной точки и без самой десятичной точки в случае целых чисел). Гарантируется, что существует такая последовательность результатов партий, при которой счёт $A:B$ будет достигнут.

Формат выходных данных

В первой строке выведите общее количество сыгранных партий. Во второй — наименьшее возможное количество ничьих в матче. В третьей — наибольшее возможное количество ничьих в матче.

Примеры

стандартный ввод	стандартный вывод
1.5	2
0.5	1
	1
0	0
0	0
	0

2.2.2 Идея алгоритма

Число ходов - сумма целых частей $a + b$ и $+1$ (если a заканчивается на .5)

¹Источник задачи A и всех ниже задач: <https://rucode.net/>

Минимальное количество ничей - 0, если оба целые, иначе 1

Максимальное - $b/0.5$

2.2.3 Решение задачи

Фрагмент кода решения задачи А представлено ниже в листинге 1.

Листинг 1 – Задача А. Анализ хода матча

```
#include <bits/stdc++.h>

using namespace std;

void solve(){
    double a, b;
    cin >> a >> b;
    if(a < b)
        swap(a, b);
    int ans = (int)a + (int)b;
    if(int(a * 10)%10 == 5)
        ans++;
    cout << ans << "\n";
    if(int(a * 10)%10 == 5)
        cout << 1 << "\n";
    else
        cout << 0 << "\n";
    cout << b/0.5 << "\n";
}

int main(){
    int t=1;
```

```
while(t--)  
    solve();  
  
return 0;  
}
```

2.2.4 Результат решения задачи

Со 1-ой попытки программа прошла все тесты.

Результат: Решена.

2.3 Задача С. Выделяем римские анаграммы

2.3.1 Формулировка задачи

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Вам дана строка S , составленная из букв I, V, X, L, C, D и M. Ваша задача — подсчитать количество различных анаграмм этой строки, которые являются корректными римскими числами (сама строка также считается своей анаграммой). Таблица из Википедии показывает, каким образом записываются римские числа:

Разряды	Тысячи	Сотни	Десятки	Единицы
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

Заметим, что:

- Обозначения для 4, 9, 40, 90, 400 и 900 используют так называемую инвертированную запись, когда первый символ вычитается из последующего (например, для 40 (XL) X (10) вычитается из L (50)). Это единственные случаи инвертированной записи.
- Число, содержащее несколько десятичных цифр, строится конкатенацией римской записи каждой цифры, слева направо от большей к меньшей.
- Если в каком-то разряде стоит 0, то в соответствующем месте ничего не приписывается.
- Наибольшее число, которое может быть представлено в римской записи, это 3,999 (MMMCMXCIX).

Формат входных данных

Первая строка входных данных содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 77\,777$). Каждая из последующих T строк

содержит один тестовый пример — слово s , состоящее из букв I, V, X, L, C, D и M ($1 \leq |s| \leq 15$).

Формат выходных данных

Для каждого тестового примера выведите одно целое число — количество анаграмм заданной строки, являющихся корректными римскими числами.

Примеры

стандартный ввод	стандартный вывод
2	1
IC	1
XV	

2.3.2 Идея алгоритма

Решение "в лоб". Идея - записать все варианты в условия и перебрать всевозможные положения букв, проверить на корректность с помощью большого количества условий. Если получалась корректная римская цифра — добавлял в сумму.

2.3.3 Решение и результат задачи

По причине нехватки времени программа не была реализована в полном объеме, не написаны все необходимые условия.

Результат: Не решена.

2.4 Задача F. Если лодку украли...

2.4.1 Формулировка задачи

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

У знаменитого изобретателя Леонардо украли лодку. Но ему надо перевезти на другой берег свою библиотеку. Леонардо решил сколотить вместо лодки плавучий ящик в форме прямоугольного параллелепипеда. Быстро написав трактат «О прямоугольных лодках», он определил, что длина, ширина и высота, измеренные в сантиметрах, должны быть целыми числами, ширина лодки должна быть в точности равна $1/3$ от длины, а высота должна быть в точности равна $1/2$ от ширины. Сам ящик (вне зависимости от размеров — всё же не зря Леонардо считают гением) весит 50 килограммов. Библиотека Леонардо весит W килограммов, и он хочет перевезти её за один раз. Напоминаем, что лодка с грузом плавает, если суммарный вес лодки и груза в точности равен весу воды, вытесненной лодкой. Верхняя сторона ящика и поверхность воды параллельны. Леонардо хочет, чтобы расстояние от верхней стороны ящика до ватерлинии (линии контакта с поверхностью воды) была не менее 30 сантиметров. Напишите программу, которая найдёт минимальное значение длины, ширины и высоты ящика, который должен сделать Леонардо. Напоминаем, что 1000 кубических сантиметров воды весят 1 килограмм

Формат входных данных

Входные данные состоят из одного целого числа W ($1 \leq W \leq 1000$) — суммарный вес библиотеки Леонардо в килограммах.

Формат выходных данных

В первой строке выведите общее количество сыгранных партий. Во второй — наименьшее возможное количество ничьих в матче. В третьей — наибольшее возможное количество ничьих в матче.

Примеры

стандартный ввод	стандартный вывод
42	216 72 36
128	240 80 40

2.4.2 Идея алгоритма

Вспомнив формулу силы Архимеда 1 из курса физики, выразили все условия. $l=3w=6h=x$ После получили итоговую формулу 2, которую будем проходить в цикле.

$$F_A = q * g * V \quad (1)$$

Где F_A - сила Архимеда, q - плотность жидкости, g - коэффициент свободного падения, V - объем тела.

$$(50 + W) * 1000 \leq 12 * x * x * (x - 30) \quad (2)$$

Где W - вес груза.

2.4.3 Решение задачи

Фрагмент кода решения задачи F представлено ниже в листинге 2.

Листинг 2 – Задача F. Если лодку украли...

```
#include <bits/stdc++.h>
using namespace std;

void solve(){
    int w;
    cin >> w;
    for(int i=0; i<100000; i++){
```

```

        if((50 + w)*1000 <= 12 * i * i * (i - 30)){
            cout << i*6 << " " << i * 2 << " " << i;
            break;
        }
    }
}

int main(){
    int t=1;
    while(t--)
        solve();

    return 0;
}

```

2.4.4 Результат решения задачи

Со 1-ой попытки программа прошла все тесты.

Результат: Решена.

2.5 Задача J. Играем в угадайку?

2.5.1 Формулировка задачи

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Это интерактивная задача Программа жюри загадала целое число N от 1

до 10^6 , и ваша задача — отгадать его. Для этого вы можете работать со специальным регистром X . В начале игры $X = N$. Вы можете задавать запросы вида $? d$, где d — целое число между 0 и 10^6 . Программа жюри заменяет текущее значение X значением $X + d$. Если $X + d > 2^{20}$, то вы проиграли. В противном случае программа жюри отвечает 1, если $X + d$ — квадрат целого числа, и 0 в противном случае. Вы можете задать не более 2023 запросов. Если вы считаете, что у вас хватает информации, чтобы назвать N , вы сообщаете ответ программе жюри в формате $! N$. Если вы угадали, решение зачтено. Это действие не считается запросом. Гарантируется, что интерактор не является адаптивным, то есть значение N определяется перед началом взаимодействия и не меняется в процессе.

Протокол взаимодействия

Входные данные состоят из одного целого числа W ($1 \leq W \leq 1000$) — суммарный вес библиотеки Леонардо в килограммах. Взаимодействие начинается ваша программа, выводя запрос. Запрос имеет вид $? d$ ($0 \leq d \leq 10^6$). В ответ на это программа жюри выведет 0 или 1 в зависимости от нового значения регистра X (1, если для некоторого целого q $q \cdot q = X$, и 0 в противном случае). После чего вы задаёте следующий вопрос и так далее. Если вы хотите вывести ответ, выводите его в виде $! N$. Не забывайте сбрасывать буфер вывода после каждого запроса или вывода ответа, иначе ваша программа может получить вердикт `Idleness Limit`.

Пример

стандартный ввод	стандартный вывод
1	? 2
0	? 8
1	? 8
1	? 75
	! 7

2.5.2 Идея алгоритма

Пытаемся высчитать расстояние от начальной позиции до ближайших квадратов впереди числа. Если получается найти число, расстояние от которого до следующих квадратов совпадает, то оно является ответом.

2.5.3 Решение задачи

Фрагмент кода решения задачи J представлено ниже в листинге 3.

Листинг 3 – Задача J. Играем в угадку?

```
#include <bits/stdc++.h>

using namespace std;

int n = 1000001;

void solve(){
    vector <int> a(2*n+1, inf);
    for(int i=0; i<=1300; i++)
        a[i*i] = 0;
```

```

int t = 0;
for(int i=1300*1300; i>0; i--){
    if(a[i] == 0)
        t=0;
    a[i] = t++;
}

vector <int> ans;

int cnt = 0;
int tmp;
for(int i=0; i<2022; i++){
    cout << "? " << "1" << " ";
    cout.flush();
    cin >> tmp;
    if(tmp == 1)
        ans.push_back(i+1);
    cnt++;
}

for(int i=1; i<=1000000; i++){
    bool isK = 1;
    for(int j=0; j<ans.size(); j++){
        if(a[i+ans[j]] != 0){
            isK = 0;
            break;
        }
    }
    if(!isK){
        continue;
    }
}

```

```

    }
    while(cnt < 2021){
        cnt++;
        cout << "? " << a[i + ans[ans.size()-1]] << "
        cout.flush();
        cin >> tmp;
        if(tmp == 1)
            ans.push_back(ans[ans.size()-1]);
        else{
            isK = 0;
            break;
        }
    }
    if(isK){
        cout << "! " << i << "\n";
        cout.flush();
    }
}
}

```

2.5.4 Результат решения задачи

Код оказался нерабочим. Программа сломалась на 10 тесте. Идея алгоритма оказалась неверной.

Результат: Не решена.

2.6 Задача L. Крутые номера

2.6.1 Формулировка задачи

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Автомобильные номерные знаки в Берляндии имеют следующий формат: letter letter digit digit digit letter letter digit digit. Здесь letter обозначает строчную латинскую букву, а digit — десятичную цифру. Допустима любая комбинация букв и цифр в рамках этих ограничений. «Крутыми» номерами в Берляндии считаются номера, в которых и цифровая часть, и буквенная являются палиндромами. Например, номер ab123ba21 — крутой, так как и abba, и 12321 — палиндромы, а номер aa111aa22 — нет, так как число 11122 палиндромом не является. Дорожная инспекция Байтландии засекла нарушителя на крутом автомобиле с крутым номером. Номер записался на камеру, однако есть проблема: из-за высокой скорости распознать символ на некоторых местах не получается. Соответствующие буквы (или цифры) в строке номера заменены звёздочкой (*), одна звёздочка соответствует одной цифре или букве в зависимости от номера позиции. Вам дана распознанная строка. Найдите количество возможных крутых номеров, которые могли быть на фотографии.

Формат входных данных

Входные данные состоят из одной строки длины 9. Символы на позициях 0,1,5,6 — или строчные латинские буквы, или '*'; символы на остальных позициях — или цифры, или '*'.

Формат выходных данных

Выведите одно целое число — количество крутых номеров, которые соответ-

ствуют данным с фотографии.

Примеры

стандартный ввод	стандартный вывод
a*****b**	0
*****	676000
aa777aa77	1

2.6.2 Идея алгоритма

Проверяем на корректность. Если все ок, то начинаем считать варианты: Мы хотим, чтобы номер был палиндромом. То есть, у каждого элемента строки есть парный с другой стороны, с которым он должен совпадать (кроме центрального элемента. Число символов нечетное — значит один не парный и в нем может стоять что угодно)

1. Если центральный элемент не зафиксирован — умножаем ответ на 10 (т.к. на этой позиции число);
2. если парный элемент зафиксирован, а текущий символ — нет, то ответ не увеличивается (умножается на 1);
3. если парный элемент и текущий символ не зафиксированы, умножаем ответ на 26, если должна стоять буква, и на 10, если должно быть число.

2.6.3 Решение задачи

Фрагмент кода решения задачи L представлено ниже в листинге 4.

Листинг 4 – Задача L. Крутые номера

```
#include <bits/stdc++.h>

using namespace std;
```

```

void solve(){
    string str;
    cin >> str;
    int ans = 1;
    for(int i=0; i<str.size(); i++){
        if(str[i] == '*' && i == 4){
            ans *= 10;
        }
        if(str[i] == '*' && (i == 0 || i == 1)){
            if(i==0 && str[6] == '*')
                ans *= 26;
            if(i==1 && str[5] == '*')
                ans *= 26;
        }
        if(str[i] == '*' && (i == 2 || i == 3)){
            if(i==2 && str[8] == '*')
                ans *= 10;
            if(i==3 && str[7] == '*')
                ans *= 10;
        }
    }

    if(str[0] != str[6] && str[0] != '*' && str[6] != '*')
        ans = 0;

    if(str[1] != str[5] && str[1] != '*' && str[5] != '*')
        ans = 0;

    if(str[2] != str[8] && str[2] != '*' && str[8] != '*')

```

```
        ans = 0;

        if(str[3] != str[7] && str[3] != '*' && str[7] != '*')

            ans = 0;

        cout << ans << "\n";

    }
```

2.6.4 Результат решения задачи

Со 1-ой попытки программа прошла все тесты.

Результат: Решена.

Заключение

Нашей командой были успешно решены 3 задачи из 12 и занято 100 место из 227.

Решение 5-ой задачи мы фактически нашли, но не смогли его правильно оформить. В ходе практики мной были приобретены знания алгоритмов и опыт работы с языком C++, а также опыт работы в команде.

Список литературы

1. Котельников, И. А. LaTeX по-русски [Электронный ресурс]. / И. А. Котельников, П. З. Чеботаев. — 3-е изд., перераб. и доп. — Новосибирск : Сибирский хронограф, 2004. — 496 с. — URL: https://www.researchgate.net/publication/235255954_LaTeX_po-russki
2. Вики-конспекты // Wiki URL: <https://neerc.ifmo.ru/wiki/> (дата обращения: 27.11.2022)

3. Каталог алгоритмов // Codeforces URL: <https://codeforces.com/catalog>
(дата обращения: 27.11.2022)