

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
Санкт-Петербургский государственный политехнический университет

**Веренинов И. А., Петров А. В.,
Фёдоров С. А.**

Программирование
Упражнения и лабораторные работы

Редакция: 12.12.2014

Санкт-Петербург

Содержание

Глава 1. Дисциплина программирования.....	3
§ 1.1 Организация проектов.....	3
§ 1.2 Организация исходного кода.....	3
§ 1.3 Основы профессионального стиля оформления кода.....	5
Глава 2. Эффективная работа разработчика в редакторе Vim.....	6
§ 2.1 Основные команды по работе с исходным кодом.....	6
§ 2.2 Предлагаемый файл конфигурации vimrc.....	6
Глава 3. Упражнения.....	7
§ 3.1 Общие требования к упражнениям.....	7
§ 3.2 Опорные варианты упражнений.....	7
§ 3.3 Указания по использованию встроенных средств.....	7
Глава 4. Лабораторные работы.....	9
§ 4.1 Общие требования к лабораторным работам.....	9
§ 4.2 Лабораторная работа № 1. Разработка структур данных.....	9
4.2.1 Общая часть задания.....	9
4.2.2 Опорный вариант лабораторной работы № 1.....	10
4.2.3 Индивидуальные задания.....	10

Глава 1. Дисциплина программирования

§ 1.1 Организация проектов

- Каждое задание выполняется в виде отдельного проекта программного обеспечения;
- проект содержит директории:
 - src — исходные файлы;
 - bin — исполняемые файлы и выходные файлы;
 - data — входные файлы
 - obj — объектные файлы;
- сборочный файл (Makefile) размещается в корне директории проекта:
 - цель all содержит компиляцию каждого исходного файла по отдельности и их компоновку в исполняемый файл;
 - компилятор и его опции задаются через переменные;
 - обязательные опции при компиляции и компоновке:
 - -Wall — вывод всех предупреждений;
 - -std=f2008ts — проверка на соответствие исходного кода стандарту ISO/IEC 1539-1:2010 Information technology — Programming languages — Fortran - Part 1: Base language (Fortran 2008) и техническому описанию ISO/IEC TS 29113:2012 Information technology — Further interoperability of Fortran with C;
 - -Wno-maybe-uninitialized — отключение предупреждений о *возможно* не инициализированных переменных;
 - -fimplicit-none — запрет неявного задания типов;
 - -static-libgfortran — статическая компоновка библиотеки Fortran для переносимости исполняемого кода;
 - -flto — оптимизация времени компоновки (link time optimization);
 - цель clean должна удалять все файлы в директориях bin и obj:

```
rm -f obj/*.  
rm -f bin/*.  
rm -f data/*.  
rm -f *.o
```

- проект не должен содержать *ошибок и предупреждений* на этапе сборки;
- каждый проект использует модуль Environment;
- входные и выходные файлы:
 - при необходимости сравнения символа строки с определённым символом используется код этого символа (необходимые коды брать с www.unicode.org/charts):

```
if (string(1:1) == Char(Int(z'041C'), CH_) & ! Сравнение с «М».
```

- при разработке использовать текстовый редактор Vim и компилятор gfortran.

§ 1.2 Организация исходного кода

Исходный код пишется кратко и лаконично для возможности его повторного использования как литературный текст «без запутанного сюжета и с наименьшим числом героев». При этом:

- головная программа содержит только вызовы процедур — подпрограмм и функций, — каждая решающая свою задачу (кроме первой части лабораторной работы № 1):

```
program reference_lab_1_5  
  use Environment
```

```

implicit none
type (student), pointer :: GroupList => Null()

call Read_class_list(INPUT_FILE, GroupList)
call Output_class_list(OUTPUT_FILE, GroupList)
call Average_marks(GroupList)
call Sort_class_list(GroupList, STUD_AMOUNT)
call Output_sorted_class_list(OUTPUT_FILE, GroupList)
end program reference_lab_1_6

```

- головная программа и модули обязательно содержат оператор:

```
implicit none
```

- в исполняемых операторах используются **только** именованные константы, а не безымянные:

```

integer, parameter :: M = 24
allocate (A(M))
allocate (A(10)) ! Не допустимо.

```

- описания всех процедур, а также содержательные участки кода снабжаются комментариями;
- номера устройств файлов присваиваются **только** автоматически:

```
open (file=InputFile, encoding=E_, newunit=In)
```

- **везде**, где это возможно, используется средства регулярного программирования. Например удаление нулевых строк и столбцов из матрицы проводится за **3 оператора**:

```

NotNullRows(:,1) = Any(A /= 0, dim=2) ! Позиции ненулевых строк (N, 1).
NotNullCols(1,:) = Any(A /= 0, dim=1) ! Позиции ненулевых столбцов (1, N).
A = Reshape( Pack(A, MatMul(NotNullRows, NotNullCols)), &
            [Count(NotNullRows), Count(NotNullCols)])

```

- все данные задаются в отдельных входных файлах;
- формат ввода/вывода программируется, а сам ввод/вывод описывается с использованием регулярного программирования и списков:

```

stud_form = '(3(a, 1x), ' // MARKS_AMOUNT // 'i1)'
read (In, stud_form, iostat=IO) (Surnames(i), Initials(i), Sex(i), Marks(i, :),
i = 1, STUD_AMOUNT)

```

- процедуры:
 - процедуры ввода/вывода при чтении одинаковых структур данных используются повторно:

```

call Output_list(OutputFile, List, "Исходный список: ", "rewind")
call Output_list(OutputFile, SortedList, "Отсортированный список: ", "append")

```

здесь в процедуре Output_list используется последний входной параметр position:

```
open (file=InputFile, encoding=E_, newunit=In, position=position)
```

- **все** процедуры кроме процедур ввода/вывода описываются чистыми (с квалификатором pure);
- вид связи (intent) указывается для всех параметров;
- процедуры используются с соблюдением требования наибольшей независимости по данным;
- глобальные переменные внутри процедур не используются;
- массивы и строки передаются как перенимающие форму:

```
subroutine ReadClassList(Input_File, Surnames, Initials)
```

```

character(*)      Input_File
character(kind=CH_) Surnames(:, :), Initials(:, :)
intent (in)       Input_File
intent (out)      Surnames, Initials

```

§ 1.3 Основы профессионального стиля оформления кода

Стараться придерживаться стилю по именованию объектов:

- **только** имена констант пишутся строчными буквами с нижним подчёркиванием:

```
character(*) :: STUD_AMOUNT = 12;
```

- подряд идущие операторы объявления выравниваются по символу разделения типа и оператору присваивания:

```

character(SURNAME_LEN, kind=CH_) :: Surnames(STUD_AMOUNT)      = CH_" "
character(INITIALS_LEN, kind=CH_) :: Initials(STUD_AMOUNT)    = CH_" "
character(kind=CH_)               :: Sex(STUD_AMOUNT)          = CH_" "
integer                           :: Marks(STUD_AMOUNT, MARKS_AMOUNT) = 0
real(R_)                         :: AverMarks(STUD_AMOUNT)     = 0.

```

- имена объектов должны быть лаконичным и отражать их назначение:

ClassList

- имена процедур пишутся с большой буквы с нижним подчёркиванием, а их параметры задаются через запятую:

```
subroutine Read_file(file_name, list)
```

- индексы и триплеты массивов задаются через запятую и пробел:

```
A(1:K, 1:L) = 0
```

- имена переменных и типов пишутся с большой буквы без нижнего подчёркивания:

```
type(Student), pointer :: ClassList
```

- все ключевые слова языка пишутся с маленькой буквы:

```
select case(IO)
```

Строго придерживаться стилю оформления кода:

- операторы описания отделяются от исполняемых операторов пустой строкой;
- описание входных параметров процедур отделяются от локальных переменных пустой строкой;
- операторы внутри каждой программной единицы и конструкции отделяются слева 3 пробелами:

```

if (Associated(InitialCurrent)) &
  allocate (DiffCode)

```

- после имени каждого оператора ставится пробел:

```
open (file=InputFile, encoding=E_, newunit=In)
```

- символы операций и присваивания отделяются пробелами, а имена дополнительных параметров в операторах и процедурах передаются без пробелов:

```
Aver_marks = Real(Sum(marks, dim=2), R_) / MARKS_AMOUNT
```

Глава 2. Эффективная работа разработчика в редакторе Vim

§ 2.1 Основные команды по работе с исходным кодом

:help windows	— справка по эффективной работе с окнами
:help quickfix	— справка по эффективной работе с исходным кодом
:compiler gfortran	— установка поддержки сообщений от gfortran
:make [arg]	— вызов программы сборки
:copen	— список быстрых исправлений
:cclose	— закрытие списка быстрых исправлений
:clist	— список ошибок
:cc [nr]	— переход к месту текущей ошибки или к ошибке номер nr
:cn[ext]	— переход к следующему месту ошибки
:cp[revious]	— переход к предыдущему месту ошибки

§ 2.2 Предлагаемый файл конфигурации vimrc

```
syntax on
filetype plugin indent on
let fortran_do_enddo=1
set smartindent
set autoindent
set expandtab
set tabstop=3
set shiftwidth=3
compiler! gfortran

" Save all and build
map <F7> :wall \ | make all<Cr>
" Run executable
map <F8> :make run<Cr>
" Go to next error
map <F6> :cn<Cr>zvzz:cc<Cr>
" Go to previous error
map <F5> :cp<Cr>zvzz:cc<Cr>
```

Глава 3. Упражнения

При выполнении упражнений, создавая своё произведение, необходимо следовать дисциплине программирования, описанной в главе 1. При сдаче принимается исходный код ПО и результаты его работы.

§ 3.1 Общие требования к упражнениям

1. Все исходные данные вводятся из входного форматированного файла.
2. Все входные и выходные форматированные файлы имеют формат UTF-8.
3. Текст во входных и выходных файлах написан на русском языке (если присутствует).
4. В выходной форматированный файл отдельными процедурами выводится как вся введенная информация, так и полученный результат.
5. Каждое упражнение выполняется в императивном и регулярном стиле. Регулярный стиль отлаживается и оформляется в комментариях.

§ 3.2 Опорные варианты упражнений

Номера опорных упражнений: 1.пример, 1.3, 2.пример, 3.пример, 4.пример, 4.2а, 4.3а, 5.пример, 6.пример, 7.5а, 7.14а, 7.24, 7.30, 7.48, 8.пример1, 8.пример2, 8.22

§ 3.3 Указания по использованию встроенных средств

Не забывайте использовать встроенные математические функции, а также Abs, Exp, SqRt.

- 1.2 forall ($X(i) = x^{i-1}$), $C = [f, e, d, c, b, a]$, Dot_product ($p = X \cdot C$).
- 2.6 Max, Min.
- 3.2 forall, Product.
- 3.3 do (построить массив факториалов), forall (записать в него же каждый множитель), Product.
- 3.4 Sum (для сечения).
- 3.8 forall (построить маску диагональной матрицы), Sum (или сразу Sum((a(i, i), i = 1, N))).
- 3.16 Dot_product.
- 4.1 Сформировать $X(N) \Rightarrow Y(N)$.
- 4.2а Элементные функции, forall — заполнять по строкам.
- 4.3-6 Sum, элементные функции.
- 5.1 Pack, массив индексов.
- 5.2 forall (построить маску), Pack (или $a(2:n-1) < a(1:n)$).
- 5.4 MinLoc, Sum.
- 5.5 Sum, Count.
- 5.6 Any.

- 5.10 сечение из четных и нечетных элементов.
- 5.11-13 Только MinVal, MinLoc. Иногда в 5.12 с параметром back = .true.
- 6.2 Проверить точность.
- 7.2 Count, A = [Pack(...), Pack(...)]
- 7.3 C = [A, B], сортировка.
- 7.4 Записать на 5-ый элемент, Count (к обеим частям в отдельности, определив, в какое место вставлять), CShift (циклически сдвинуть соответствующую часть).
- 7.5a Pack, Unpack.
- 7.6 Sum, Product, SqRt.
- 7.7 Sum, Product.
- 7.9a Sum.
- 7.9b MaxVal, Sum, Abs.
- 7.9в Sum, MaxVal.
- 7.10 Sum, Abs.
- 7.12 MinLoc, Sum.
- 7.14 Forall.
- 7.15 Одним оператором: MinLoc, MaxLoc.
- 7.17 Max, Min, Reshape, Pack, Spread (см. раздаточные 7.x).
- 7.20 Pack. Проводить вычисления для получения маски дорого. Маска — это шахматная доска: Mask(::2, ::2) = .true.; Mask(2::2, 2::2) = .true.
- 7.22 MaxVal.
- 7.27 Сечение в виде каждого второго столбца. Forall.
- 7.30 Any, Reshape, Pack, MatMul, Count.
- 7.31 MaxLoc.
- 7.32 Count или MinLoc, Cshift.
- 7.33 MaxVal.
- 7.34 MaxLoc, forall.
- 7.36 Unpack.
- 7.40 Pack. Проводить вычисления для получения маски дорого. Маска — это шахматная доска: Mask(::2, ::2) = .true.; Mask(2::2, 2::2) = .true.
- 7.41 Reshape, Transpose, Pack.
- 7.44 MaxVal.
- 7.50 Pack, Reshape.
- 8.1 Transpose, MatMul.
- 8.6 Sum.
- 8.9 Transpose, MatMul.

Глава 4. Лабораторные работы

При выполнении лабораторных работ, создавая своё произведение, необходимо следовать дисциплине программирования, описанной в главе 1. При сдаче принимается исходный код ПО и результаты его работы.

Для студентов, обучающихся на базе профессионального образования, шестая лабораторная работа является необязательной.

§ 4.1 Общие требования к лабораторным работам

6. Все исходные данные вводятся из входного форматированного файла.
7. Все входные и выходные форматированные файлы имеют формат UTF-8.
8. Текст во входных и выходных файлах написан на русском языке.
9. В выходной форматированный файл отдельными процедурами выводится как вся введенная информация, так и полученный результат.
10. При разработке кода с модулями:
 1. должен быть один модуль ввода/вывода и один модуль с обработкой данных, содержащий *только чистые* процедуры;
 2. основной тип данных описывается в модуле ввода/вывод.

§ 4.2 Лабораторная работа № 1. Разработка структур данных

4.2.1 Общая часть задания

1. Одна и та же информация из входного файла вводится по-разному в оперативную память с целью освоения работы с различными структурами данных — задание выполняется в виде 5 отдельных программных проектов, где необходимо использовать:

	Проект лабораторной работы № 1				
Средства	1	2	3	4	5
массивы строк	+				
массивы символов		+			
внутренние процедуры головной программы		+			
массивы структур			+	+	
файлы записей ¹			+	+	
модули			+	+	+
хвостовая рекурсия				+	+
однонаправленные списки заранее неизвестной длины ²					+

1 Файл записей формируется отдельной процедурой из форматированного файла *запись за записью*, а затем из него происходит чтение *одной записи* — массива структур.

2 Каждая строка исходного форматированного файла рассматривается как элемент списка, файл с неизвестным числом строк читается до конца, длину списка разрешается запомнить при чтении и использовать только при сортировке.

регулярное программирование	+	+	+	+	+
-----------------------------	---	---	---	---	---

2. Необходимо прочитать список известной длины из не менее чем 12 строк (кроме проекта № 5). Данные в одной строке имеют заданный формат и отделяются друг от друга дополнительным пробелом.

4.2.2 Опорный вариант лабораторной работы № 1

Дан список группы с результатами сессии и с неподсчитанным средним баллом в виде:

ФАМИЛИЯ ИНИЦИАЛЫ ПОЛ РЕЗУЛЬТАТЫ_СЕССИИ СРЕДНИЙ_БАЛЛ
15 симв. 5 симв. 1 симв. 5 симв. 3 симв.

Пример входного файла:

```
Дудиков      Д. Р. М 43453 0.0
Тихонов      Л. П. М 55353 0.0
Садовникова П. О. Ж 43543 0.0
Степин       К. Д. М 55445 0.0
Воробьева    Е. Р. Ж 44353 0.0
```

Рассчитать средний балл для каждого из учащихся. Отсортировать по убыванию среднего балла списки юношей и девушек по отдельности.

Успеваемость юношей:

```
Степин       К. Д. М 55445 4.60
Тихонов      Л. П. М 55353 4.20
Дудиков      Д. Р. М 43453 3.80
```

Успеваемость девушек:

```
Воробьева    Е. Р. Ж 44353 3.80
Садовникова  П. О. Ж 43543 3.80
```

4.2.3 Индивидуальные задания

1. Дан список группы в виде:

ФАМИЛИЯ ПОЛ РЕЗУЛЬТАТЫ СЕССИИ
15 симв. 1 симв. 4 симв.

Пример входного файла:

```
Иванов      М 4455
Петрова     Ж 3554
```

Определить лидеров среди мужчин и женщин по успеваемости и их средний балл (у мужчин и женщин отдельно). Пример выходного файла

Лидеры по успеваемости:

```
Иванов      4455 ср. балл 4.50
Петрова     3554 ср. балл 4.30
```

Средний балл среди мужчин: 4.2

Средний балл среди женщин: 4.1

2. Дан список группы в виде:

ФАМИЛИЯ И. О. ПОЛ ПРОПИСКА СРЕДНИЙ_БАЛЛ
15 симв. 5 симв. 1 симв. 1 симв. 4 симв.

Пример входного файла (в графе прописки буква П стоит у петербурцев, с — у гостей Санкт-Петербурга):

```
Иванов      И. И. М П 4.35
```

Отсортировать по убыванию среднего балла по отдельности списки петербуржцев и гостей Санкт-Петербурга. Пример выходного файла:

Петербуржцы:

Иванов	И. И. М 4.35
Барабашкин	И. И. М 4.32

Гости города:	
Петров	И. И. М 4.35
Петрыкин	И. И. М 4.32

3. Дан список группы в виде:
- | | | | |
|---------|---------|---------|--------------|
| ФАМИЛИЯ | И. О. | ПОЛ | ГОД РОЖДЕНИЯ |
| 15 сив. | 5 симв. | 1 симв. | 4 симв. |

Пример входного файла:

Иванов	И. И. М 1995
--------	--------------

Определить средний возраст юношей в группе. Примеры выходного файла:

Средний возраст юношей в группе: 21 год
Средний возраст юношей в группе: 22 года
Средний возраст юношей в группе: 25 лет

4. Дан список группы в виде:

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	СЛУЖБА В АРМИИ	ПРОПИСКА	ПОЛ
15 симв.	4 симв.	3 симв.	1 симв.	1 симв.

Пример входного файла (в графе прописки буква П стоит у петербурцев, с — у гостей Санкт-Петербурга):

Иванов	1992 нет П М
Петров	1993 да С М

Отсортировать в алфавитном порядке по отдельности списки петербуржцев и гостей Санкт-Петербурга, служивших в армии. Пример выходного файла:

Служившие петербуржцы:	
Барабашкин	1992 М
Служившие гости города:	
Петров	1993 М

5. Дан список сотрудников научно-исследовательской лаборатории в виде:

ФАМИЛИЯ	ДОЛЖНОСТЬ
15 симв.	15 симв.

Пример входного файла:

Иванов	техник
--------	--------

Определить число одинаковых должностей. Пример выходного файла:

ведущий инженер	- 2
старший инженер	- 3
инженер	- 8
техник	- 2

6. Дан список товаров с указанием цены вида:

НАИМЕНОВАНИЕ ВЕЩИ	ЦЕНА	ВАЛЮТА
20 симв.	5 симв.	5 симв.

Пример входного файла:

Стол	2500 руб.
------	-----------

Определить три самые дорогие товара. Пример выходного файла:

Самые дорогие товары:

- | | |
|----------|-------------|
| 1. Диван | - 12000 руб |
| 2. Стул | - 9000 руб. |
| 3. Тумба | - 6000 руб. |

7. Дан список группы в виде:

ФАМИЛИЯ	И. О.
15 сив.	5 симв.

Пример входного файла:

Иванов	И. И.
--------	-------

Отсортировать список по алфавиту с учетом инициалов методом сравнения «один со всеми». Пример выходного файла:

Иванов	И. А.
Иванов	И. И.
Иванова	И. И.

8. Дан список владельцев телефонов в виде:

ФАМИЛИЯ	ТЕЛЕФОН
15 симв.	10 симв.

Пример входного файла:

Петров	9111634576
Фёдоров	9111635687

Отсортировать этот список в порядке убывания номеров телефонов, используя метод вставок. Пример выходного файла:

Фёдоров	9111635687
Петров	9111634576

9. Дан список группы в виде:

ФАМИЛИЯ	И. О.	ГОД РОЖДЕНИЯ
15 сив.	5 симв.	4 симв.

Пример входного файла:

Иванов	И. И.	1994
Петров	Д. Т.	1992

Выделить первого по алфавиту и самого молодого. Пример выходного файла:

Первый по алфавиту:	
Иванов	И. И. 1994
Самый молодой:	
Петров	Д. Т. 1992

10. Дан список товаров в виде:

НАИМЕНОВАНИЕ	ЦЕНА	ВАЛЮТА	ПРОЦЕНТ ИЗНОСА
10 симв.	5 симв.	5 симв.	3 симв.

Пример входного файла:

Стол	2500 руб.	45
------	-----------	----

Используя процент износа, сформировать продажную комиссионную цену. Пример выходного файла:

Комиссионная цена вещей:	
Стол	1375 руб.

11. Дан список сотрудников научно-исследовательской лаборатории в виде:

ФАМИЛИЯ ДОЛЖНОСТЬ

15 симв. 15 симв.

Пример входного файла:

Иванов техник

Отсортировать список в порядке повышения должности от «техника» до «вед. инженера». Пример выходного файла:

Иванов техник
Петров старший инженер

12. Дан список группы в виде:

ФАМИЛИЯ И. О. ГОД РОЖДЕНИЯ ПРОПИСКА ПОЛ

15 симв. 5 симв. 4 симв. 1 симв. 1 симв.

Пример входного файла (в графе прописки буква П стоит у петербурцев, С — у гостей Санкт-Петербурга):

Иванов И. И. 1995 П М
Петрова Х. Л. 1994 С Ж

Выделить из них трёх наиболее молодых петербуржцев мужчин. Пример выходного файла:

Иванов И. И. 1995
Галкин В. И. 1997

13. Дан список группы в виде:

ФАМИЛИЯ ИМЯ ОТЧЕСТВО

15 симв. 10 симв. 15 симв.

Пример входного файла:

Безруков Сергей Викторович
Лебедев Пётр Станиславович

Определить число встречающихся имен. Пример выходного файла:

Сергей - 2
Пётр - 1

14. Дан список владельцев телефонов в виде:

ФАМИЛИЯ ТЕЛЕФОН

15 симв. 10 симв.

Пример входного файла:

Петров 9111634576
Фёдоров 9111635687
Петров 9111634573

Найти строку с первой по алфавиту фамилией и с наименьшим номером телефона. Пример выходного файла:

Петров 9111634573

15. Дан список группы в виде:

ФАМИЛИЯ ИМЯ ОТЧЕСТВО

15 симв. 10 симв. 15 симв.

Пример входного файла:

Безруков Сергей Викторович
Лебедев Пётр Станиславович
Глухих Сергей Алексеевич

Удалить из списка всех, у которых имена совпадают, кроме первого. Пример выходного файла:

Безруков	Сергей	Викторович
Лебедев	Пётр	Станиславович

16. Дан список группы в виде:

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	ПОЛ
15 симв.	4 симв.	1 симв.

Пример входного файла:

Иванов	1985	М
Петрова	1983	Ж

Найти самого пожилого мужчину и самую молодую женщину. Пример выходного файла:

Самый пожилой мужчина:	
Иванов	1985
Самая молодая женщина:	
Петрова	1983 Ж

17. Дан список группы в виде:

ФАМИЛИЯ	ПОЛ	РЕЗУЛЬТАТЫ СЕССИИ
15 симв.	1 симв.	4 симв.

Пример входного файла:

Иванов	М 4455
Петрова	Ж 3554

Отсортировать по убыванию среднего балла мужчин и женщин по отдельности. Пример выходного файла:

Мужчины:	
Иванов	М 4455 4.50
Женщины:	
Петрова	Ж 3554 4.25

18. Дан список группы в виде:

ФАМИЛИЯ	И. О.	ПОЛ	ГОД РОЖДЕНИЯ	ПОЛ
15 симв.	5 симв.	1 симв.	1 симв.	

Пример входного файла:

Иванов	И. Л.	М	1985
Петрова	Д. О.	Ж	1983

Сформировать отсортированные по убыванию возраста списки мужчин и женщин. Использовать для сортировки метод “выбором”. Пример выходного файла:

Мужчины:	
Иванов	И. Л. М 1985
Женщины:	
Петрова	Д. О. Ж 1983

19. Дан список сотрудников лаборатории в виде:

ФАМИЛИЯ	ПРОФЕССИЯ
15 симв.	10 симв.

Пример входного файла:

Иванов	повар
Петрова	фармацевт

Сформировать список профессий с указанием их количества. Пример выходного файла:

повар	- 1
фармацевт	- 1