

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)

Кафедра прикладной математики и кибернетики

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ  
по дисциплине «Функциональное и логическое программирование»

Вариант   1  

Выполнил:

студент группы   ИП-216  

Андрущенко Филипп  
Александрович

ФИО студента

Работу проверил: Новожилов  
Дмитрий Иванович

ФИО преподавателя

Новосибирск 2024 г.

1.1. Исходный список преобразуйте в новый "двойной" список таким образом, чтобы каждый элемент исходного списка удвоился.

Например, [1,2,3]-> [1,1,2,2,3,3].

Листинг :

task\_1 :-

```
write('Введите список : '),  
read(InputList),  
double_elements(InputList, DoubledList),  
write('Удвоенный список : '),  
format('Удвоенный список : ~w\n', [DoubledList]).
```

```
double_elements([], []).
```

```
double_elements([Head|Tail], [Head,Head|DoubledTail]) :-  
    double_elements(Tail, DoubledTail).
```

Демонстрация :

```
?- task_1.  
Введите ].исок : [1,2,3,4  
Удвоенный список : Удвоенный список : [1,1,2,2,3,3,4,4]  
true.
```

2.1. Текстовый файл состоит из нескольких строк, длина каждой из которых меньше 80. Выровняйте строки текста по ширине (равномерным внесением дополнительных пробелов между словами). Длины новых строк должны быть равны 80. Сформируйте новый файл.

Листинг :

```
:- use_module(library(lists)).
```

```
task_2(InputFile, OutputFile) :-  
    process_file(InputFile, OutputFile).
```

```
process_file(InputFile, OutputFile) :-  
    open(InputFile, read, InStream),  
    open(OutputFile, write, OutStream),  
    process_lines(InStream, OutStream),  
    close(InStream),  
    close(OutStream).
```

```
process_lines(InStream, OutStream) :-  
    read_line_to_string(InStream, CurrentLine),  
    ( CurrentLine == end_of_file  
    -> true  
    ; split_string(CurrentLine, " ", "", WordsList),  
      justify_words(WordsList, JustifiedLine),  
      writeln(OutStream, JustifiedLine),  
      process_lines(InStream, OutStream)  
    ).
```

```
justify_words(WordsList, JustifiedLine) :-  
    length(WordsList, WordCount),  
    WordCount > 1,  
    maplist(string_length, WordsList, WordLengths),  
    sum_list(WordLengths, TotalWordLength),  
    RequiredSpaces is 80 - TotalWordLength,  
    NumberOfGaps is WordCount - 1,  
    BaseSpaces is RequiredSpaces // NumberOfGaps,  
    ExtraSpaces is RequiredSpaces mod NumberOfGaps,  
    distribute_spaces(WordsList, BaseSpaces, ExtraSpaces, JustifiedLine).
```

```
distribute_spaces([Word], _, _, Word).
```

distribute\_spaces([FirstWord, SecondWord | RemainingWords], BaseSpaces, ExtraSpaces, JustifiedLine) :-

```
( ExtraSpaces > 0
-> ExtraSpacesForThisGap is BaseSpaces + 1,
NewExtraSpaces is ExtraSpaces - 1
; ExtraSpacesForThisGap is BaseSpaces,
NewExtraSpaces is ExtraSpaces
),
create_spaces(ExtraSpacesForThisGap, SpaceString),
string_concat(FirstWord, SpaceString, TempLine),
distribute_spaces([SecondWord | RemainingWords], BaseSpaces, NewExtraSpaces,
RestLine),
string_concat(TempLine, RestLine, JustifiedLine).
```

create\_spaces(0, "").

create\_spaces(Number, Spaces) :-

```
Number > 0,
NextNumber is Number - 1,
create_spaces(NextNumber, SubSpaces),
string_concat(" ", SubSpaces, Spaces).
```

Демонстрация :

```
?- task_2('input.txt', 'output.txt').
true .
```

```
mazda toyota mitsubishi nissan
1 ford chevrolet dodge tesla
2 bmw mercedes porsche audi
```

mazda	toyota	mitsubishi	nissan
1 ford	chevrolet	dodge	tesla
2 bmw	mercedes	porsche	audi