

Упражнения: Функционално програмиране

Problem 1. Action

Напишете програма, която чете колекция от низове от конзолата и след това ги отпечата на конзолата. Всяко име трябва да бъде отпечатано на нов ред. Използвайте `Action<T>`

Примери

Вход	Изход
Pesho Gosho Adasha	Pesho Gosho Adasha

Problem 2. Рицари на честта

Напишете програма, която чете колекция от имена като **символни низове** от **конзолата** след това добавя "Sir" пред всяко име и го отпечата обратно върху конзолата. Използвайте `Action<T>`

Примери

Вход	Изход
Pesho Gosho Adasha StanleyRoyce	Sir Pesho Sir Gosho Sir Adasha Sir StanleyRoyce

Problem 3. Собствена Min функция

Напишете една проста програма, която чете от конзолата набор от цели числа и отпечата обратно върху конзолата най-малкото число от колекцията. Използвайте `Func<T, T>`.

Примери

Вход	Изход
1 4 3 2 1 7 13	1

Problem 4. Намерете четно или нечетно

Дадени са Ви долна и горна граница за интервал от числа. След това команда определя, ако трябва да се изброят всички четни и нечетни числа в дадения интервал. Използвайте `Predicate<T>`

Примери

Вход	Изход
1 10 odd	1 3 5 7 9
20 30 even	20 22 24 26 28 30

Problem 5. Приложна аритметика

Напишете програма, която изпълнява някои математически операции върху дадена колекция. На първия ред са дадени списък от числа. На следващите редове се подават различни команди, които трябва да се прилагат за всички числа в списъка: **"Add"** -> добави 1 към всяко число; **"multiply"** -> умножава всяко число по 2; **"subtract"** -> изважда 1 от всяко число; **"print"** -> извежда колекцията. Входът ще приключи с команда **"end"**. Използвайте функции.

примери

Вход	Изход
1 2 3 4 5 add add print end	3 4 5 6 7
5 10 multiply subtract print end	9 19

Problem 6. Обърни и изпълни

Напишете програма, която обръща колекция и премахва елементи, които са се дели на дадено число n . Използвайте предикати/функции.

Примери

Вход	Изход
1 2 3 4 5 6 2	5 3 1
20 10 40 30 60 50 3	50 40 10 20

Problem 7. Предикати за имена

Напишете програма, която филтрира списък с имена по тяхната дължина. На първия ред ще ви бъде дадено **цяло число n** , представляващо дължина на име. На втория ред ще ви бъде дадени някои имена като **символни низове**, разделени с интервал. Напишете функция, която отпечатва само имената, чиято дължина е по-малка или равна на n .

Примери

Вход	Изход
4 Kurnelia Qnaki Geo Muk Ivan	Geo Muk Ivan
4 Karaman Asen Kiril Yordan	Asen

Problem 8. Потребителски сравнител

Напишете потребителски компаратор, който сортира всички четни числа преди всички нечетни такива във възходящ ред. Подайте го на функция `Array.Sort()` и изведе резултата. Използвайте функции.

Examples

Input	Output
1 2 3 4 5 6	2 4 6 1 3 5
-3 2	2 -3

Problem 9. Списък от предикати

Find all numbers in the range $1...N$ that are divisible by the numbers of a given sequence. On the first line you will be given an integer **N** – which is the end of the range. On the second line you will be given a sequence of integers which are the dividers. Use predicates/functions.

Намерете всички числа в диапазона **1... N**, които са делими от числата на определена последователност. На първия ред ще ви бъде дадена цяло число **N** –, което е краят на интервала. На втория ред ще ви бъде дадена последователност от числа, които са делители. Използвайте предикати/функции.

Примери

Вход	Изход
10 1 1 1 2	2 4 6 8 10
100 2 5 10 20	20 40 60 80 100

Problem 10. Предикатно парти!

Родителите на Иванчо са на почивка за празниците и той планира епично парти у дома. За съжаление неговите организационни умения са почти никакви и така ви се предоставя задачата да му помогнете с резервации. На първия ред получавате списък с всички хора, които ще дойдат. На следващите редове докато получите командата **"Party!"**, може да бъдете помолени да удвоите или премахнете всички хора, които удовлетворяват дадени критерии. Има три различни критерии: 1. всеки, чието име започва с даден низ; 2. всеки, чието име завършва с даден низ; 3. всеки, чието име е с дадена дължина. Накрая отпечатвате всички гости, които ще са на партито, разделени с "," и след това добавете окончанието **"are going to the party!"**. Ако няма никакви гости, които да отидат на партито отпечатайте **"Nobody is going to the party!"**. Вижте примерите по-долу:

Примери

Вход	Изход
Pesho Misho Stefan Remove StartsWith P Double Length 5 Party!	Misho, Misho, Stefan are going to the party!
Pesho Double StartsWith Pesh Double EndsWith esho	Pesho, Pesho, Pesho, Pesho are going to the party!

Party!	
Pesho Remove StartsWith P Party!	Nobody is going to the party!

Problem 11. Модул Филтър за резервация на парти

You need to implement a filtering module to a party reservation software. First, to the Party Reservation Filter Module (PRFM for short) is **passed a list** with invitations. Next the PRFM receives a **sequence of commands** that specify whether you need to add or remove a given filter.

Each PRFM command is in the given format **{command;filter type;filter parameter}**

You can receive the following PRFM commands: **"Add filter"**, **"Remove filter"** or **"Print"**. The possible PRFM filter types are: **"Starts with"**, **"Ends with"**, **"Length"** and **"Contains"**. All PRFM filter parameters will be a string (or an integer only for the **"Length"** filter). Each command will be valid e.g. you won't be asked to remove a non-existent filter.

The input will **end** with a **"Print"** command after which you should print all the party-goers that are left after the filtration. See the examples below:

Вие трябва да реализирате филтриращ модул за софтуер за резервации на партита. Първо, на модула Филтър за резервация на партита (PRFM за кратко) е подаден списък с покани. После PRFM получава последователност от команди, които указват дали е необходимо да добавите или премахнете даден филтър. Всички команди към PRFM се във формат {команда; тип на филтъра; филтриращ параметър} можете да получите следните PRFM команди: : **"Add filter"**, **"Remove filter"** или **"Print"**. Всички PRFM филтърни типове са: **"Starts with"**, **"Ends with"**, **"Length"** и **"Contains"**. Всички PRFM филтър параметри ще бъдат низове (или цяло число само за филтъра **"Length"**). Всяка команда ще бъде валидна например вие няма да бъдете помолени да премахнете несъществуващ филтър. Входът ще завърши с команда **"Print"**, след което трябва да отпечатате всички, които отиват на парти, останали след филтриране. Вижте примерите по-долу:

Примери

Вход	Изход
Pesho Misho Slav Add filter;Starts with;P Add filter;Starts with;M Print	Slav
Pesho Misho Jica Add filter;Starts with;P Add filter;Starts with;M Remove filter;Starts with;M Print	Misho Jica

Problem 12. Inferno III.

"Sum Left" tests if a gem's **power added** to the gem standing to **its left** gives a **certain value**. **"Sum Right"** is the same but looks to a gem's **right peer**. **"Sum Left Right"** sums the gems power with **both its left and right** neighbors. If a gem has **no neighbor** to its right or to its left (first or last element), then simply **add 0** to the gem.

На **първия ред** са дадени **поредица от числа**. Всяко число е скъпоценност и стойността представлява **силата** му. На **следващите** редове, докато получите командата **"Forge"**, ще получавате команди в следния

формат: {команда; тип филтър; филтриращ параметър}. Командите могат да бъдат: "Exclude", "Reverse" или "Forge". Възможните типове Филтър са: "Sum Left", "Sum Right" и "Sum Left Right". Всички филтриращи параметри ще бъдат **цяло число**.

"Exclude" маркира скъпоценност за изключване от множеството, ако тя отговаря на дадени условия,

"Reverse" изтрива предишното изключване.

"Sum Left" проверява дали силата на скъпоценния камък, добавен към скъпоценните камъни, стоящи отляво, дава определена стойност

"Sum Right" е същото, но за дясната страна на скъпоценния камък.

"Sum Left Right" сумира силата на скъпоценния камък с левия и десния съсед.

Ако скъпоценен камък няма съсед отдясно или отляво (първия или последния елемент), тогава просто се добавя 0 за скъпоценност.

Имайте предвид, че **промени** в поредицата **се прилагат** само **след коване Forge**. Това означава, че скъпоценностите са фиксирани на техните позиции и всяка функция се случва на оригиналния набор, така че всяка скъпоценност се взема предвид, без значение дали е маркирана да бъде изключена или не. За по-добро разбиране на проблема, вижте примерите по-долу:

Примери

Вход	Изход	Коментари
1 2 3 4 5 Exclude;Sum Left;1 Exclude;Sum Left Right;9 Forge	2 4	1. Marks for exclusion all gems for which the sum with neighbors to their left equals 1, e.g. $0 + 1 = 1$ 2. Marks for exclusion all gems for which the sum with neighbors to their left and their right equals 9, e.g. $2 + 3 + 4 = 9$ $4 + 5 + 0 = 9$
1 2 3 4 5 Exclude;Sum Left;1 Reverse;Sum Left;1 Forge	1 2 3 4 5	1. Marks for exclusion all gems for which the sum with their gem peers to the left equals 1, e.g. $0 + 1 = 1$ 2. Reverses the previous exclusion.

Problem 13. TriFunction

Напишете програма, която получава колекция от имена и връща първото име, чийто сбор от символи е равен на или по-голям от дадено число N, което ще бъде изведено на един ред. Използвайте функция, която приема друга функция като един от нейните параметри. Започнете с изграждането на регулярна функция, която да държи основната логика на програмата. Нещо по **Func<string, int, bool>**. След това създайте главна функция, която трябва да приеме първата функция като един от нейните параметри.

Примери

Вход	Иход
800 Qvor Qnaki Petromir Saddam	Petromir

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма **"Обучение за ИТ кариера"** на МОН за подготовка по професия "Приложен програмист".



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).



SoftUni
Foundation

