

Проблем 1: Укупан износ корпе за куповину

Текст проблема:

Корисник каже: „Потребна ми је једноставна функција за мој веб сајт. Купац додаје артикле у корпу, а сваки артикал има цену. Само желим да видим укупну цену свега у корпи. Понекад постоји попуст, па желим да будем у могућности да смањим цену. На пример, ако је укупан износ 100 динара, а попуст 20%, коначна цена треба да буде 80 динара. Ако нема попушта, само прикажите оригинални укупан износ.“

Шта спецификација треба да дефинише:

Улаз(и)

Излаз(и)

Правила и ограничења (нпр. Шта са празном корпом, негативним ценама?)

Примери са јасним улазима и очекиваним излазима.

Проблем 2: Валидатор корисничког имена

Текст проблема:

Креирамо корисничке налоге за нашу нову апликацију. Морамо да проверимо да ли је корисничко име које особа жели прихватљиво. Правила су: мора бити дуго најмање 5 знакова, али не дуже од 15. Може да садржи само слова (и велика и мала слова су дозвољена) и бројеве, без размака, без симбола. Такође, не може да почиње бројем. Функција би требало само да нам каже „да, ово је валидно“ или „не, није“.

Шта спецификација треба да дефинише:

Улаз(и)

Излаз(и) (тип података и значење)

Прецизан списак свих правила валидације.

Проблем 3: Проналазач књига

Текст проблема:

Имам списак књига у својој библиотеци. Свака књига има наслов и аутора. Желим да укуцам име аутора и добијем списак свих књига које имам од тог аутора. Ако немам ниједну књигу тог аутора, требало би да добијем празну листу. Претрага не би требало да води рачуна о великим или малим словима. Дакле, претрага за „џаџ МаЈић“ требало би да пронађе књиге Џаџа Мајића.

Шта спецификација треба да дефинише:

Улаз(и) (структура података за библиотеку и термин за претрагу)

Израз(и) (структура података резултата)

Правила тачног подударења (осетљивост на велика и мала слова? делимична подударења?).

Примери са примером скупа података библиотеке.

Проблем 4: Додељивач оцена

Текст проблема:

Наши наставници треба брзо да конвертују нумерички резултат теста (од 0 до 100) у коначну оцену. Ево система који користимо: 90 или више је '10', 80 до 89 је '9', 70 до 79 је '8', 60 до 69 је '7', 50 до 59 је „6“, а све испод 50 је "5". Функција треба да узме резултат и избаци оцену.

Шта спецификација треба да дефинише:

Улаз(и) и њихов валидан опсег.

Излаз(и).

Прецизне границе за сваку оцену (шта се дешава тачно на 90? 80?).

Како руковати неважећим улазима (нпр. резултат 101 или -5).

Проблем 5: Дневни циљ корака

Текст проблема:

Правим функцију за апликацију за фитнес. Корисник поставља дневни циљ корака. Током дана, апликација бележи колико су корака корисници направили. Потребна ми је функција која проверава да ли су остварили свој циљ. Такође би требало да им каже колико им је корака преостало ако га још нису остварили. Ако су прекорачили циљ, требало би да каже за колико. На пример, ако им је циљ 10.000 корака, а направили су 12.000, требало би да каже да су прекорачили за 2.000 корака.

Шта спецификација треба да дефинише:

Улаз(и) (циљ и тренутни кораци).

Излаз(и) (порука или подаци које треба вратити).

Различити логички случајеви (испуњено, није испуњено, прекорачено) и тачан излаз за сваки.

Претпоставке (нпр., да ли кораци или циљ могу бити негативни?).