

вариант	ф. номер	група	поток	курс	специалност
K2.1					
Име:					

Контролна работа по логическо програмиране  
28 май 2022 г.

Да няма лист, на който е писано по повече от една задача!

За всеки дефиниран предикат да се попълни подходящият/те шаблон(и):

1. При параметри ..., предикатът ... разпознава дали ...
2. При параметри ..., предикатът ... генерира ... в ...
3.  $p(\dots)$  е истина тогава и само тогава, когато ... Следното условие е достатъчно, за да няма зацикляне с предиката: ...

Решения на задачи, в които това отсъства, ще бъдат оценявани с 0 точки.

**Зад. 1.** Разглеждаме крайни множества от крайни затворени интервали от реални числа с непразна вътрешност, чиито крайща са цели числа. Нека  $A$  е едно такова множество. Казваме, че  $A$  е *нормално*, ако всеки два различни негови елемента имат празно сечение; обединението на елементите на  $A$  ще наричаме *обем* на  $A$ . *Представяне* на  $A$  е списък  $[\ell_1, \ell_2, \dots, \ell_n]$ , където  $n$  е броят на елементите на  $A$ ,  $\ell_i$  за  $1 \leq i \leq n$  е двуелементен списък  $[a_i, b_i]$  от цели числа,  $a_i < b_i$  и  $\{x \in \mathbb{R} \mid a_i \leq x \leq b_i\} \in A$ .

Да се дефинира на пролог двуместен предикат `normalize_c(L,X)`, който по даден списък  $L$ , представляващо множество  $A$  от интервали от разглеждания вид, генерира в  $X$  списък, представляващо нормално множество  $B$  от интервали, което има същия обем като  $A$ .

**Зад. 2.** Да приложим операцията *обмен на делител* към един списък от положителни цели числа означава да изберем два негови члена  $a_i$  и  $a_j$ , които не са равни и не са взаимно прости и да ги заместим съответно с  $a_i d$  и  $a_j / d$  или съответно с  $a_i / d$  и  $a_j d$ , където  $d$  е най-малкият прост делител на  $a_i$  и  $a_j$ . За списък от цели положителни числа ще казваме, че е *стабилен*, ако всеки два негови члена са равни или са взаимно прости. Да се дефинира на пролог двуместен предикат `min_stable(L,N_exchange)`, който по даден списък  $L$  от положителни цели числа генерира в  $N\_exchange$  минималния брой прилагания на операцията обмен на делител, с които започвайки от  $L$  се достига до стабилен списък.

Пожелаваме ви приятна и успешна работа!

вариант	ф. номер	група	поток	курс	специалност
K2.2					
Име:					

Контролна работа по логическо програмиране  
28 май 2022 г.

Да няма лист, на който е писано по повече от една задача!

За всеки дефиниран предикат да се попълни подходящият/те шаблон(и):

1. При параметри ..., предикатът ... разпознава дали ...
2. При параметри ..., предикатът ... генерира ... в ...
3.  $p(\dots)$  е истина тогава и само тогава, когато ... Следното условие е достатъчно, за да няма зацикляне с предиката: ...

Решения на задачи, в които това отсъства, ще бъдат оценявани с 0 точки.

**Зад. 1.** Разглеждаме крайни множества от крайни отворени интервали от реални числа с непразна вътрешност, чиито крайща са цели числа. Нека  $A$  е едно такова множество. Казваме, че  $A$  е *нормално*, ако всеки два различни негови елемента имат празно сечение; обединението на елементите на  $A$  ще наричаме *обем* на  $A$ . *Представяне* на  $A$  е списък  $[\ell_1, \ell_2, \dots, \ell_n]$ , където  $n$  е броят на елементите на  $A$ ,  $\ell_i$  за  $1 \leq i \leq n$  е двуелементен списък  $[a_i, b_i]$  от цели числа,  $a_i < b_i$  и  $\{x \in \mathbb{R} \mid a_i < x < b_i\} \in A$ .

Да се дефинира на пролог двуместен предикат `normalize_o(L,X)`, който по даден списък  $L$ , представящ множество  $A$  от интервали от разглеждания вид, генерира в  $X$  списък, представящ нормално множество  $B$  от интервали, което има същия обем като  $A$ .

**Зад. 2.** Да приложим операцията *обмен на единица* към един списък от положителни цели числа означава да изберем два негови члена  $a_i$  и  $a_j$ , който не са взаимно прости и да ги заместим съответно с  $a_i - 1$  и  $a_j + 1$  или съответно с  $a_i + 1$  и  $a_j - 1$ . За списък от цели положителни числа ще казваме, че е *солиден*, ако всеки два негови члена са взаимно прости.

Да се дефинира на пролог двуместен предикат `min_solid(L,N_exchange)`, който по даден списък  $L$  от положителни цели числа генерира в  $N\_exchange$  минималния брой прилагания на операцията обмен на единица, с които започвайки от  $L$  се достига до солиден списък.

*Пожелаваме ви приятна и успешна работа!*