

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Областен кръг

15 февруари 2020 г.

Група В, 9-10 клас

## Задача В1. Два контейнера

Дадени са  $n$  предмета с целочислени тегла  $a_1, a_2, \dots, a_n$ . Разполагаме с два контейнера. Първият може да събере предмети с общо тегло, най-много равно на  $W_1$ , а вторият – с общо тегло, най-много равно на  $W_2$ . Как да поставим предметите в контейнерите, така че да вземем предмети с възможно най-голямо сумарно тегло? Всеки от дадените предмети може да поставим само в един от контейнерите, без да разделяме предмета на части. Възможно е някои предмети да не поставим в нито един от контейнерите.

Напишете програма **double**, която намира максималното общо тегло на предметите, които можем да поставим в контейнерите.

**Вход.** От първия ред на стандартния вход се въвеждат стойностите на  $n$ ,  $W_1$  и  $W_2$ , разделени с интервали. От втория ред се въвеждат стойностите  $a_1, a_2, \dots, a_n$ , разделени с интервали.

**Изход.** На един ред на стандартния изход изведете едно цяло число, равно на търсеното максимално тегло.

**Ограничения.**  $1 < n < 200$ ,  $0 < W_1 < 200$ ,  $0 < W_2 < 200$ ;  $n$ ,  $W_1$  и  $W_2$  са цели числа.

Теглата  $a_1, a_2, \dots, a_n$  са цели положителни числа и са по-малки от 100.

## Примери

Пример 1	Пример 2
<b>Вход</b> 3 10 2 3 4 5	<b>Вход</b> 5 10 14 4 5 6 7 8
<b>Изход</b> 9	<b>Изход</b> 23

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Областен кръг  
15 февруари 2020 г.  
Група В, 9-10 клас

## Задача В2. Излишни скоби

Даден е символен низ, който представлява правилен аритметичен израз. В този израз се съдържат операнди, знаците за четирите операции '+', '-', '\*', '/' (събиране, изваждане, умножение и деление), както и малки скоби '(' и ')'. Всички операнди са означени с един и същи символ '&'. **Всички операции са с по два операнда (т.е. те са бинарни операции)** и приоритетът им е като общоприетия – събиране и изваждане са с еднакъв приоритет и умножение и деление са с еднакъв приоритет, който е по-висок от този на събирането и изваждането. При липса на скоби, операциите с по-висок приоритет се изпълняват преди операциите с по-нисък приоритет. Последователните операции с еднакъв приоритет се изпълняват от ляво надясно. Скобите, естествено, се използват за промяна на реда на изпълнение на операциите. Ето три примера за правилни аритметични изрази:

1.  $&+&-&$     2.  $(&-(&+&)*&)-&$     3.  $&/&*(&+&)$

**Забележка:** Символът '&' е абстрактно означение за операнд, а не име на променлива. В един и същи израз на мястото на '&' могат да стоят различни числа. Например, аритметични изрази, съответстващи на пример 1, може да бъдат:  $5+6-7$ ,  $20+1-9$  и т.н.

Погледнете добре аритметичния израз от пример 2 – в него има излишни скоби. Веднага се вижда, че той е еквивалентен на израза  $&-(&+&)*&-&$ . **Два аритметични израза се наричат еквивалентни, ако дават една и съща стойност при заместване на операндите им с едни и същи стойности, придвижвайки се отляво надясно.**

Двойка съответни отваряща и затваряща скоби ще наричаме „излишна“, ако след премахването на тези две скоби, се получава аритметичен израз, еквивалентен на първоначалния.

**Внимание:** Премахването на скобите е съвсем „механично“, т.е. не е разкриване на скобите според математическите правила, а просто премахване на двете съответни скоби от символния низ, без да се променя нищо друго в него. Например, да премахнем скобите в израза  $&-(&-&)$  означава да получим израза  $&-&-&$ , който не е еквивалентен на първоначалния (а не израза  $&-&+&$ , както би се получило, ако разкрием скобите според съответното аритметично правило).

Напишете програма **brackets**, която в зададен правилен аритметичен израз премахва всички излишни двойки скоби.

**Вход.** От един ред на стандартния вход се въвежда символен низ, представляващ правилен аритметичен израз. Той може да съдържа само символите '&', '+', '-', '\*', '/', '(' и ')'. Между символите няма интервали.

**Изход.** На един ред на стандартния изход изведете символен низ, който представлява правилен аритметичен израз, получен от входния, като са премахнати всички излишни двойки скоби.

**Ограничения.**  $3 \leq$  брой символи във входния низ  $\leq 200\,000$ .

Във входния символен низ **не се срещат** поднизове от вида (&), т.е. между всеки две съответни отваряща и затваряща скоби има поне една операция.

**Оценяване.** В 40% от тестовете присъстват само двете операции '+' и '\*' и скоби.

### Примери

	Вход	Изход
1.	$((&+&)*&)/&$	$(&+&)*&/&$
2.	$&+(&*(&-&)-(&-&))$	$&+&*(&-&)-(&-&)$
3.	$&+(((&/&)))/&$	$&+&/&/&$

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Областен кръг  
15 февруари 2020 г.  
Група В, 9-10 клас

## Задача В3. U-конгруентност

NU-редица наричаме всяка редица от  $N$  цели положителни числа, елементите на която не надвишават цялото положително число  $U$ . Означаваме такава редица с  $a_{\langle N,U \rangle} = \{a_1, a_2, \dots, a_N\}$ , където  $1 \leq a_i \leq U$  са цели числа за всяко  $1 \leq i \leq N$ . Ще казваме, че две NU-редици  $v_{\langle N_1,U_1 \rangle}$  и  $w_{\langle N_2,U_2 \rangle}$  са U-конгруентни (и ще записваме  $v \cong w$ ), ако са изпълнени следните три условия:

- $N_1 = N_2$ ;
- $U_1 = U_2 = U$ ;
- Съотношението между **всеки два** елемента в едната редица е точно същото, каквото е между съответните елементи в другата.

Ще изясним последното условие. Ако в редицата  $v$  разгледаме кои да е два елемента, например този на място  $i$  и този на място  $j$ , означени съответно с  $v_i$  и  $v_j$ , то:

- ако  $v_i < v_j$ , то за елементите на тези места в редицата  $w$  е в сила същото отношение:  $w_i < w_j$ ;
- ако  $v_i > v_j$ , то  $w_i > w_j$ ;
- ако  $v_i = v_j$ , то  $w_i = w_j$ .

За редиците  $a_{\langle 4,10 \rangle} = \{3, 1, 7, 3\}$  и  $b_{\langle 4,10 \rangle} = \{5, 2, 10, 5\}$  е изпълнено  $a \cong b$ : те са с еднакъв брой елементи, с еднаква горна граница и съотношението на кои да е два елемента от едната е същото, каквото между съответните елементи в другата.

Напротив, за  $a_{\langle 4,10 \rangle} = \{3, 9, 7, 4\}$  и  $b_{\langle 4,10 \rangle} = \{5, 9, 6, 7\}$  е изпълнено  $a \not\cong b$ : първите две условия са изпълнени, част от съотношенията са същите, но  $a_3 > a_4$ , докато  $b_3 < b_4$ . Наличието на поне едно нарушение на подредбата прави редиците неконгруентни.

Поради очевидните си свойства рефлексивност, симетричност и транзитивност, описаната релация „U-конгруентност“ разбива множеството на всички NU-редици при фиксирани  $N$  и  $U$  на непресичащи се семейства взаимно U-конгруентни редици. Можете да съобразите, че едно такова семейство може да е доста голямо. Нека броят на редиците в него е  $n$  и нека си представим редиците в това семейство, лексикографски нарастващо наредени и номерирани от 1. „Средна редица“ в семейството ще наричаме редицата, която се намира на място  $\lceil (n+1)/2 \rceil$ , т.е. „в средата“ на списъка (тук  $\lceil x \rceil$  означава „цялата част на  $x$ “). Напишете програма **ucongruence**, която намира средната редица  $w$  в семейство редици, U-конгруентни на зададена редица  $v$ .

### Вход

От стандартния вход се въвеждат:

- ред 1: целите положителни числа  $N$  и  $U$ , разделени с интервал;
- ред 2:  $N$  цели положителни числа, всяко не по-голямо от  $U$ , разделени с интервал, представляващи една NU-редица.

### Изход

Програмата трябва да извежда на стандартния изход един ред, който съдържа  $N$  цели положителни числа, разделени с интервал: средната редица в семейството от NU-редици, U-конгруентни с въведената.

### Ограничения

$N \leq 1\,000\,000$ ;

$U \leq 60$ ;

В 30% от тестовите примери  $U \leq 30$ .

### Пример

Вход	Изход	Обяснение
4 26 4 6 6 4	8 17 17 8	Елементите в семейството са : $\{\{1, 2, 2, 1\}, \{1, 3, 3, 1\}, \{1, 4, 4, 1\}, \dots, \{1, 26, 26, 1\}, \{2, 3, 3, 2\}, \{2, 4, 4, 2\}, \dots, \{2, 26, 26, 2\}, \dots, \{24, 25, 25, 24\}, \{24, 26, 26, 24\}, \{25, 26, 26, 25\}\}$ . Броят на елементите в него е $n=325$ , средната редица е на място 163.