Контролно по УП

Име:			
Специалност:	Група:	Фн:	Дата:

Задача 1. Наблюдаваме приближаваща колона от индианци откъм предния ѝ край. За един индианец казваме, че е "*видим*", ако всички индианци пред него са по-ниски от него (т.е. пред него няма по-висок или равен по ръст индианец, който да го закрива изцяло). Да се напише функция **`visibleIndians`**, която приема масив от цели числа, съдържащ височините на индианците в реда, в който са подредени в колоната, и връща като резултат броя на видимите индианци.
Пример: Ако масивът е { 150, 163, 158, 180, 173 }, видимите индианци са 3 на брой.

Задача 2. Казваме, че един масив от цели неотрицателни числа е "*балансиран*", ако сумата на елементите в първата му половина е равна на сумата на елементите във втората му половина. Да се напише функция **`isBalanced`**, която по подаден масив с елементи от тип `unsigned int` връща булева стойност, указваща дали масивът е балансиран. Да се демонстрира използването на функцията **`isBalanced`** в кратка **`main`** функция.
Примери: Масивът { 5, 10, 4, 1, 7, 3, 9 } е балансиран, а масивът { 5, 10, 4, 1, 7, 6, 9 } не е.

Задача 3. Да се напише функция **`maxOccurs`**, която получава като аргумент символен низ и връща броя на срещанията на най-срещаната буква в низа. Ако в низа няма нито една буква, функцията да връща 0. Функцията не трябва да прави разлика между главни и малки букви, т.е. прибавят се към една и съща бройка.
Пример: `maxOccurs("OnomatOpoeia") → 4` (буквата `'O'` се среща 4 пъти)

Задача 4. Да се напише функция **`oneBit`**, която получава като аргументи две цели числа без знак и връща булева стойност, указваща дали техният двоичен запис се различава точно в един бит.
Примери: `oneBit(4,5) → true, oneBit(20,28) → true, oneBit(21,22) → false.`

Задача 5. Да се напише програма, която прочита от стандартния вход число **n** и извежда на стандартния изход елхичка с **n** клонки на най-ниския ред, както е показано по-долу, когато **n** е нечетно и текста `"Happy New Year!"` ако **n** е четно. Да не се извеждат интервали в края на редовете. Числото **n** не трябва да надхвърля 52 (броя седмици в годината).

Примери:

Вход	**Изход**

5	<pre> +
	   o * o
	  o * * o
	 o * * * o
	    |</pre>

Вход	**Изход**
3	<pre> + o * o |</pre>
6	`Happy New Year!`

Тема 2

Контролно по УП

Име:			
Специалност:	Група:	Фн:	Дата:

Задача 1. Наблюдаваме отдалечаваща се колона от индианци откъм задния ѝ край. За един индианец казваме, че е "*видим*" ако всички индианци зад него са по-ниски от него (т.е. зад него няма по-висок или равен по ръст индианец, който да го закрива изцяло от нашата гледна точка). Да се напише функция **visibleIndians**, която приема масив от цели числа, съдържащ височините на индианците в реда, в който са подредени в колоната отпред назад, и връща като резултат броя на видимите индианци.
Пример: Ако масивът е { 173, 180, 158, 163, 150 }, видимите индианци са 3 на брой.

Задача 2. "*Тежест*" на масив от неотрицателни числа наричаме сумата на цифрите на елементите на масива. Да се напише функция **compareArrays**, която по подадени два масива с елементи от тип `unsigned int` връща 1, 0 или -1 в зависимост дали тежестта на първия масив е съответно по-голяма (1), равна (0), или по-малка (-1) от тежестта на втория масив. Да се демонстрира използването на функцията **compareArrays** в кратка **main** функция.
Пример: Масивът { 5, 124, 77, 10 } е по-тежък от масива { 42351, 1000, 32, 225 }.

Задача 3. Да се напише функция **minOccurs**, която получава като аргумент символен низ и връща броя на срещанията на най-рядко срещаната буква в низа. Ако в низа няма нито една буква, функцията да връща 0. Функцията не трябва да прави разлика между главни и малки букви, т.е. прибавят се към една и съща бройка.
Пример: `minOccurs("Mama & papa") → 2`

Задача 4. Да се напише функция **oneBit**, която приема като аргументи две цели числа без знак и връща булева стойност, която указва дали при премахване на една цифра от двоичния запис на едното число се получава другото.
Примери: `oneBit(4,12) → true`, `oneBit(27,13) → true`, `oneBit(21,22) → false`

Задача 5. Да се напише програма, която прочита от стандартния вход число **n** и извежда на стандартния изход пясъчен часовник с ширина на основата **n**, както е показано по-долу. Когато **n** е четно в средата на часовника трябва да има точка. Да не се извеждат интервали в края на редовете. Числото **n** не трябва да надхвърля 60 (минутите в един час).

Примери:

Вход	**Изход**

7	1 2 3 4 5 6 7
	  2 3 4 5 6
	  3 4 5
	  4
	  3 4 5
	  2 3 4 5 6
	1 2 3 4 5 6 7

3	1 2 3 2 1 2 3

Вход	**Изход**

6	1 2 3 4 5 6
	  2 3 4 5
	  3 4
	  .
	  3 4
	  2 3 4 5
	1 2 3 4 5 6

Тема 3

Контролно по УП

Име:			
Специалност:	Група:	Фн:	Дата:

Задача 1. Наблюдаваме приближаваща колона от индианци откъм предния ѝ край. За един индианец казваме, че е "*невидим*" ако пред него има по-висок или равен по ръст индианец, който го закрива изцяло. Да се напише функция **`invisibleIndians`**, която приема масив от цели числа, съдържащ височините на индианците в реда, в който са подредени в колоната, и връща като резултат броя на невидимите индианци.
Пример: Ако масивът е { 150, 163, 158, 180, 173 }, невидимите индианци са 2 на брой.

Задача 2. Казваме, че един масив от цели неотрицателни числа е "*двуцветен*", ако сумата на цифрите на числата на четни индекси в масива е равна на на съответната сума на елементите на нечетни индекси. Да се напише функция **`isBicolor`**, която по подаден масив с елементи от тип `unsigned int` връща булева стойност, указваща дали масивът е двуцветен. Да се демонстрира използването на функцията **`isBicolor`** в кратка **`main`** функция.
Примери: Масивът { 121, 845, 76, 24, 42 } е двуцветен, а масивът { 121, 84, 76, 24, 42 } не е.

Задача 3. Да се напише функция **`maxOccurs`**, която получава като аргумент непразен символен низ и връща броя на срещанията на най-срещаната цифра в низа. Ако в низа няма цифри, функцията да връща 0.
Пример: `maxOccurs("(8+(2*3)-(4*8))") → 2`

Задача 4. Да се напише функция **`sameBits`**, която приема като аргументи две цели числа без знак и връща булева стойност, указваща дали едното число може да се получи от другото чрез разместване на цифрите в двоичния му запис. Да не се отчитат водещите нули.
Примери: `sameBits(5,6) → true, sameBits(12,3) → false`

Задача 5. Да се напише програма, която прочита от стандартния вход число **n** и извежда на стандартния изход перка с **n** реда, както е показано по-долу. Когато **n** е четно число оста се представя с вертикална черта. Да не се извеждат интервали в края на редовете. Числото **n** не трябва да надхвърля 100.

Примери:

Вход	**Изход**

7	1 2 3 4
	  2 3 4
	  3 4
	  4
	  4 5
	  4 5 6
	  4 5 6 7

3	1 2
	  2
	  2 3

Вход	**Изход**

6	1 2 3 |
	  2 3 |
	  3 |
	  | 4
	  | 4 5
	  | 4 5 6

Тема 4

Контролно по УП

Име:			
Специалност:	Група:	Фн:	Дата:

Задача 1. Наблюдаваме отдалечаваща се колона от индианци откъм задния ѝ край. За един индианец казваме, че е "*невидим*" ако зад него има по-висок индианец, който го закрива изцяло от нашата гледна точка. Да се напише функция **`invisibleIndians`**, която приема масив от цели числа, съдържащ височините на индианците в реда, в който са подредени в колоната отпред назад, и връща като резултат броя на невидимите индианци.
Пример: Ако масивът е { 173, 180, 158, 163, 150 }, невидимите индианци са 2 на брой.

Задача 2. Две числови редици **A** и **B** с равна дължина наричаме "*подобни*", ако за всеки индекс **i** е вярно, че сумата на цифрите на **A[i]** съвпада със сумата от цифрите на **B[i]**. "*Префикс*" на масив наричаме последователност от елементи, започваща от началото на масива. Да се напише функция **`longestSimilar`**, която която по два подадени масива с елементи от тип `unsigned int` връща дължината на най-дългите им подобни префикси. Да се демонстрира използването на функцията **`longestSimilar`** в кратка **`main`** функция.
Примери:

Масиви	**Резултат**
`{ 12, 35,  54,  72, 999 }` `{ 21, 80, 117, 121,  91 }`	**3**, защото 1+2 = 2+1, 3+5 = 8+0, 5+4=1+1+7, но 7+2 ≠ 1+2+1
`{  4,  11, 215, 1000 }` `{ 13, 200,  71,    1, 42, 21 }`	**4**, защото 4 = 1+3, 1+1 = 2+0+0, 2+1+5 = 7+1, 1+0+0+0 = 1. Така всички елементи на по-късия масив формират префикса.

Задача 3. Да се напише функция **`minOccurs`**, която получава като аргумент непразен символен низ и връща броя на срещанията на най-малко срещаната цифра в низа.
Пример: `minOccurs("1337 15 57331") → 2`

Задача 4. Да се напише функция **`altBits`**, която приема като аргумент цяло число без знак и и връща булева стойност, указваща дали двоичният запис на числото се състои изцяло от последователно редуващи се битове 1 и 0.
Пример: `altBits(21) → true`, `altBits(43) → false`

Задача 5. Да се напише програма, която прочита от стандартния вход число **n** и извежда на стандартния изход триъгълник, съставен от числата от 1 до **n**, изписани последователно, като на първия ред има едно число, на втория две и т. н. Последният ред може да остане непълен. Числата да са подравнени вдясно както е показано на примерите. Числото **n** трябва да е по-малко от 100.
Примери:

Вход	**Изход**

`7`	`      1`
	`    2  3`
	`  4  5  6`
	`7`

Вход	**Изход**

`14`	`          1`
	`       2  3`
	`    4  5  6`
	`  7  8  9 10`
	`11 12 13 14`