Введение

В данном документе приведены задачи для решения их на зачёте по Информатике и программированию.

Темы: условные операторы, циклы, массивы (одномерные, двумерные), строки как массивы символов, алгоритмы, операции над массивами, структуры, функции

Содержание по задачам

Блок задач VERY EASY ☹

Задача 1. *Конвертировать часы и минуты в секунды*

Написать функцию, которая принимает на вход положительное количество часов и минут и конвертирует в секунды.

Примеры

convert(1, 3) ➞ 3780

convert(2, 0) ➞ 7200

convert(0, 0) ➞ 0

Задача 2. *Баскетбольные очки*
Вы подсчитываете очки в баскетбольном матче, учитывая количество набранных 3-х и 2-х очковых, находите окончательные очки для команды и возвращаете это значение.

Примеры

points(1, 1) ➞ 5

points(7, 5) ➞ 29

points(38, 8) ➞ 100

Блок задач EASY

Задача 3. *Сколько решений есть у этого квадратичного уравнения?*
Квадратное уравнение $a x^2 + b x + c = 0$ имеет 0, 1 или 2 различных решения для действительных значений x. Учитывая a, b и c, вы должны вернуть количество решений уравнения.

Примеры

solutions(1, 0, -1) ➞ 2 // $x^2 - 1 = 0$ имеет 2 решения (x = 1 и x = -1).

solutions(1, 0, 0) ➞ 1 // $x^2 = 0$ имеет 1 решение (x = 0).

solutions(1, 0, 1) ➞ 0 // $x^2 + 1 = 0$ не имеет решения в некомплексных числах

Задача 4. *Оператор AND*
Написать функцию, которая работает по аналогии оператором &&.

```
bool AND(bool a, bool b){
   // put your code here
}
```

Пример строчки из main():

```
if AND(x == 3, y == 8) printf("O, yes, its true");
```

Задача 5. *Сортировка чисел в порядке возрастания*
Создайте функцию, которая принимает массив чисел и возвращает новый массив, отсортированный в порядке возрастания (от наименьшего к наибольшему).

Если аргумент функции равен NULL, пустому массиву [] или неопределенному – вернуть пустой массив.

Примеры

sortNumsAscending ([1, 2, 10, 50, 5]) ➞ [1, 2, 5, 10, 50]

sortNumsAscending ([80, 29, 4, -95, -24, 85]) ➞ [-95, -24, 4, 29, 80, 85]

sortNumsAscending ([]) ➞ []

Задача 6. *Массив кратных*
Создайте функцию, которая принимает в качестве аргументов два числа (основание массива, размер массива) и возвращает массив, кратных числу значений.

Примеры

arrayOfMultiples (7, 5) ➞ [7, 14, 21, 28, 35]

arrayOfMultiples (12, 10) ➞ [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

arrayOfMultiples (17, 6) ➞ [17, 34, 51, 68, 85, 102]

Задача 7. *Найдите недостающий номер*
Создайте функцию, которая принимает массив чисел от 1 до 10 (исключая одно число) и возвращает отсутствующее число.

Примеры

missingNum([1, 2, 3, 4, 6, 7, 8, 9, 10]) ➞ 5

missingNum([7, 2, 3, 6, 5, 9, 1, 4, 8]) ➞ 10

missingNum([10, 5, 1, 2, 4, 6, 8, 3, 9]) ➞ 7

Указания.

Массив чисел будет несортирован (не по порядку).

Только один номер отсутствует.

Задача 8. *Строка в порядке?*
Создайте функцию, которая принимает строку и возвращает истину или ложь, в зависимости от того, в порядке ли символы или нет.

Примеры

isInOrder("abc") ➞ true

isInOrder("edabit") ➞ false

isInOrder("123") ➞ true

isInOrder("xyzz") ➞ true

Задача 9. *Размер строки*
Создайте функцию, которая принимает строку (массив символов) в качестве аргумента и возвращает длину строки.

Примеры

reverse("Hello World") ➞ 11

reverse("The quick brown fox.") ➞ 20

reverse("Edabit is really helpful!") ➞ 25

Задача 10. *Обратный порядок строки*
Создайте функцию, которая принимает строку (массив символов) в качестве аргумента и возвращает строку в обратном порядке.

Примеры

reverse("Hello World") ➞ "dlroW olleH"

reverse("The quick brown fox.") ➞ ".xof nworb kciuq ehT"

reverse("Edabit is really helpful!") ➞ "!lufpleh yllaer si tibadE"

Задача 11. *Последняя цифра*
Ваша задача - создать функцию, которая принимает 3 числа (a, b, c) и возвращает истину, если последняя цифра в произведении последних цифр чисел a и b - это последняя цифра числа c. См. пример.

Примеры

lastDig(25, 21, 125) ➞ true

// Последняя цифра 25 – это 5, последняя цифра 21 – это 1, а последняя цифра

// цифра 125 равна 5, а последняя цифра 5 * 1 = 5 – это 5, что равно

// последней цифре 125 (5).

lastDig(55, 226, 5190) ➞ true

lastDig(12, 215, 2142) ➞ false

Задача 12. *Подсчет единиц в двумерном массиве*
Создайте функцию для подсчета количества единиц в 2D-массиве.

Примеры

count_ones([

  [1, 0],

  [0, 0]

]) ➞ 1

count_ones([

  [1, 1, 1],

  [0, 0, 1],

  [1, 1, 1]

]) ➞ 7

count_ones([

[1, 2, 3],

[0, 2, 1],

[5, 7, 33]

]) ➞ 2

Блок задач MEDIUM & NORMAL

Задача 13. Сумма чётных и нечётных значений

Напишите функцию, которая принимает массив чисел и возвращает структуру Answer с двумя полями:

1. Первый элемент должен быть суммой всех четных чисел в массиве.
2. Второй элемент должен быть суммой всех нечетных чисел в массиве.

Примеры

sumOddAndEven([1, 2, 3, 4, 5, 6]) ➞ [12, 9]

// 2 + 4 + 6 = 12 и 1 + 3 + 5 = 9

sumOddAndEven([-1, -2, -3, -4, -5, -6]) ➞ [-12, -9])

sumOddAndEven([0, 0]) ➞ [0, 0])

Задача 14. Саймон говорит

Создайте функцию, которая принимает два массива и возвращает true, если второй массив следует за первым массивом на один элемент, и false в противном случае. Другими словами, определите, является ли второй массив первым, сдвинутым вправо на 1.

Примеры

simonSays([1, 2], [5, 1]) ➞ true

simonSays([1, 2], [5, 5]) ➞ false

simonSays([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) ➞ true

simonSays([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) ➞ false

Указания. Оба входных массива имеют одинаковую длину и минимальную длину 2.

Задача 15. Локальные максимумы

Напишите функцию, которая возвращает все элементы в массиве, которые строго превышают их левых и правых соседей.

Примеры

miniPeaks ([4, 5, 2, 1, 4, 9, 7, 2]) ➞ [5, 9]

// 5 имеет соседей 4 и 2, оба меньше 5

miniPeaks ([1, 2, 1, 1, 3, 2, 5, 4, 4]) ➞ [2, 3, 5]

miniPeaks ([1, 2, 3, 4, 5, 6]) ➞ []

Задача 16. *Расстояние между числами*
Расстояние между двумя числами - это общее значение разницы между каждой парой цифр.

Создайте функцию, которая возвращает расстояние между двумя целыми числами.

Примеры

digitDistance (234, 489) ➞ 12

// Поскольку | 2 - 4 | + | 3 - 8 | + | 4 - 9 | = 2 + 5 + 5

digitDistance (121, 599) ➞ 19

digitDistance (12, 12) ➞ 0

digitDistance (10, 20) ➞ 1

Указания. Оба целых числа имеют одинаковую длину.

Задача 17. *Вращайся, коснись земли*
Учитывая список направлений вращения, «влево» или «вправо», верните целое число, показывающее, сколько полных вращений на 360 ° было сделано. Обратите внимание, что каждое слово в массиве считается поворотом на 90 ° в этом направлении.

Примеры

spinAround (["право", "право", "право", "право", "лево", "право"]) ➞ 1

// Вы повернулись вправо 4 раза (90 * 4 = 360)

// Вы один раз повернулись влево (360 - 90 = 270)

// Но вы снова повернулись вправо, чтобы сделать полный оборот (270 + 90 = 360)

spinAround ([«влево», «вправо», «влево», «вправо»]) ➞ 0

spinAround ([«вправо», «вправо», «вправо», «вправо», «вправо», «вправо», «вправо», «вправо»]) ➞ 2

spinAround ([«влево», «влево», «влево», «влево»]) ➞ 1

Задача 18. *Накопительный массив*
Создайте функцию, которая принимает массив и возвращает массив накопительной суммы.

Примеры

accumulatingArray([1, 2, 3, 4]) ➞ [1, 3, 6, 10]

// [1, 3, 6, 10] можно записать как [1, 1 + 2, 1 + 2 + 3, 1 + 2 + 3 + 4]

accumulatingArray([1, 5, 7]) ➞ [1, 6, 13]

accumulatingArray([1, 0, 1, 0, 1]) ➞ [1, 1, 2, 2, 3]

accumulatingArray([]) ➞ []

Задача 19. *Удаление дубликатов из массива*
Создайте функцию, которая принимает массив элементов, удаляет все повторяющиеся элементы и возвращает новый массив в том же последовательном порядке, что и старый массив (за вычетом дубликатов).

Примеры

removeDups([1, 5, 9]) ➞ [1, 5, 9]

removeDups([9, 5, 9]) ➞ [9, 5]

removeDups([1, 7, 7, 3, 1, 6, 3]) ➞ [1, 7, 3, 6]

Задача 20. *Округлить до ближайшего N*
Создайте функцию, которая принимает два целых числа, num и n, и возвращает целое число, которое делится на n и является ближайшим к num. Если есть два числа, равноотстоящих от num и делящихся на n, выберите большее из них.

Примеры

roundNumber (33, 25) ➞ 25

roundNumber (46, 7) ➞ 49

roundNumber (133, 14) ➞ 140

Блок задач HARD

Задача 21. *CAPS LOCK DAY is over!*
22 октября - ДЕНЬ CAPS LOCK. Кроме этого дня, каждое предложение должно быть в нижнем регистре, поэтому напишите функцию для нормализации предложения, полученного 22 октября.

Создайте функцию, которая принимает строку. Если строка состоит только из символов верхнего регистра, преобразуйте ее в нижний регистр и добавьте в конце восклицательный знак.

Примеры

normalize("CAPS LOCK DAY IS OVER") ➞ "Caps lock day is over!"

normalize("Today is not caps lock day.") ➞ "Today is not caps lock day."

normalize("Let us stay calm, no need to panic.") ➞ "Let us stay calm, no need to panic."

Указание. Каждая строка представляет собой предложение и должна начинаться с символа верхнего регистра.

Задача 22. *УНО, частица геймплея*
Задача навеяна популярной карточной игрой Uno.

Напишите функцию, которая принимает два аргумента: (1) текущую руку игрока и (2) текущую открытую карту на столе. Функция вернет true, если игрок может сыграть, или false, если игрок должен взять из колоды.

Цвета: к- красный, ж – жёлтый, з – зелёный, с – синий.

Значения карт: 0 - 9.

Игрок может сыграть, если:

- У него есть карта того же цвета, что и открытая карта.
- У него есть карта, номер которой совпадает с номером открытой карты.

Используйте **структуру** Карта с полями char color и int value.

Примеры

canPlay (["ж 3", "ж 7", "с 8", "к 9", "к 2"], "к 1") => true

// Так как у игрока две красные карты, а открытая карта красная.

canPlay (["ж 3", "ж 7"], "с 7") => true

// Так как у игрока 7, а открытая карта - 7.

canPlay ([«ж 3», «ж 5», «к 8»], «к 2») ➞ true

canPlay ([«ж 3», «ж 5», «к 8»], «с 5») ➞ true

canPlay ([«ж 3», «с 5», «к 8», «к 9»], «з 4») ➞ false

canPlay (["ж 3", "к 8"], "з 2") ➞ false