Государство Израиль
Министерство просвещения
Тип экзамена: на аттестат зрелости
Время проведения экзамена: лето 2025 года
Номер вопросника: 899371

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ה, 2025
מספר השאלון: 899371

Компьютеры

מדעי המחשב

Указания

הוראות

а. Продолжительность экзамена: 2 часа 30 минут.

א. משך הבחינה: שעתיים וחצי.

б. Строение вопросника и ключ к оценке:
в этом вопроснике два раздела.

Раздел первый:	(2x20)	–	40 баллов
Раздел второй:	(2x30)	–	60 баллов
	Всего	–	100 баллов

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.

פרק ראשון	(20×2)	– 40	נקודות
פרק שני	(30×2)	– 60	נקודות
סך הכול		– 100	נקודות

в. Разрешенный вспомогательный материал:
любой вспомогательный материал, за исключением калькулятора с возможностью программирования.

ג. חומר עזר מותר בשימוש:
כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

г. Особое указание:
Все программы, которые вы должны написать на языке программирования, пишите только на одном языке – Java или C# .

Примечание: баллы сниматься не будут, если в написанных вами программах вы напишете большую букву вместо маленькой или наоборот.

ד. הוראה מיוחדת:
את כל התוכניות שיש לכתוב בשפת מחשב כתבו בשפה אחת בלבד – Java או C# .

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.
Пишите только в экзаменационной тетради. Напишите слово «טיוטה» в начале каждой страницы, отведенной вами под черновик. Выполнение черновых записей на листах, не относящихся к экзаменационной тетради, может привести к тому, что экзамен будет аннулирован.

Желаем успеха!

בהצלחה!

В о п р о с ы

В этом вопроснике два раздела.

Вы должны ответить на вопросы двух разделов в соответствии с указаниями каждого раздела.

Примечание: в каждом вопросе, в котором требуется ввод, нет необходимости проверять корректность вводимых данных.

Для решающих на языке Java: в каждом вопросе, в котором требуется ввод, считайте, что в программе написана команда:

```
Scanner scan = new Scanner (System.in);
```

РАЗДЕЛ ПЕРВЫЙ (40 баллов)

Ответьте на два из вопросов 1–3 (за каждый вопрос – 20 баллов).

1. Ниже приведен метод what на языке Java или What на языке C# :

Java	C#

public static int what (int[] arr)
{
int y = 0;
int x = arr[0];
for (int i = 0; i < arr.length; i++)
{
y += arr[i];
x = Math.max (x, y);
y = Math.max (y, 0);
}
return x;
}
	public static int What (int[] arr)
	{
	int y = 0;
	int x = arr[0];
	for (int i = 0 ;i < arr.Length; i++)
	{
	y += arr[i];
	x = Math.Max (x, y);
	y = Math.Max (y, 0);
	}
	return x;
	}

Дан массив arr :

	0	1	2	3	4	5
arr	3	4	− 8	10	− 6	8

При помощи таблицы трассировки [טבלת המעקב] выполните трассировку метода what(arr) на языке Java или What(arr) на языке C# и напишите, что возвращает этот метод.

Таблица трассировки должна содержать столбец для каждой из переменных: x , y , i .

/продолжение на странице 3/

2. Реализуйте метод arrange на языке Java или Arrange на языке C# :

Java – public static void arrange (int[] arr)

C# – public static void Arrange (int[] arr)

В принимаемом массиве arr есть положительные числа и отрицательные числа.
Метод переносит все положительные числа (включая 0) в начало массива arr , а после них размещает все отрицательные числа.
Порядок расположения положительных чисел (относительно друг друга) останется таким же, каким он был до работы метода, и порядок расположения отрицательных чисел (относительно друг друга) останется таким же, каким он был до работы метода.
Пример:
для представленного ниже массива arr :

	0	1	2	3	4	5	6	7
arr	7	− 2	− 5	61	4	− 1	0	33

массив arr по окончании работы метода будет выглядеть следующим образом:

	0	1	2	3	4	5	6	7
arr	7	61	4	0	33	− 2	− 5	− 1

Объяснение: метод перенес положительные числа в начало массива и сохранил изначальный порядок расположения положительных чисел и отрицательных чисел.

/продолжение на странице 4/

3. Положительное число num будет называться "нарциссическим числом", если выполняется следующее правило:

каждую из цифр числа num возводят в степень количества цифр, которые есть в num (длина числа num), складывают все результаты и получают сумму, равную num .

Пример "нарциссического числа": число 407 , потому что, когда каждую его цифру возводят в степень 3 (количество цифр в этом числе), получают 407 : $4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$

Пример числа, которое **не является** "нарциссическим числом": число 58 , потому что когда каждую его цифру возводят в степень 2 (количество цифр в этом числе), получают 89 : $5^2 + 8^2 = 25 + 64 = 89$.

(א) Напишите внешний метод с названием isNarc на языке Java или IsNarc на языке C# , принимающий целое число num , большее 0 . Метод возвращает true , если это число нарциссическое, иначе он возвращает false .

(ב) Напишите внешний метод с названием theNarc на языке Java или TheNarc на языке C# , принимающий целое число n , большее 0 . Метод выводит на печать все нарциссические числа, которые существуют, от 1 до n (включительно).

Примечание: следует воспользоваться методом, который вы написали в пункте (א).

/продолжение на странице 5/

РАЗДЕЛ ВТОРОЙ (60 баллов)

Ответьте на два из вопросов 4–6 (за каждый вопрос – 30 баллов).

4. К местному совету Оз относится несколько поселков. Местный совет обеспечивает живущих в этих поселках детей подвозкой из дома в школу и обратно. Подвозка из дома в школу называется "подвозка туда", а подвозка из школы домой называется "подвозка обратно".
Дан класс **Transport** – подвозка, в котором четыре атрибута:

- name – название поселка, для которого производится подвозка, строкового типа
- toSchool – вид подвозки. Для "подвозки туда" значение атрибута true , для "подвозки обратно" значение атрибута false .
- num – число учеников, пользующихся подвозкой, целочисленного типа, от 1 до 50 (включительно)
- day – номер дня недели, в который производится подвозка (от первого дня недели до шестого, соответственно), целочисленного типа, от 1 до 6 (включительно).

Считайте, что определены методы get/Get для атрибутов этого класса.

(א) **(1)** Напишите конструктор класса **Transport ,** который получает название поселка, для которого производится подвозка – name , число учеников, пользующихся подвозкой – num и день недели, в который производится подвозка – day .
Конструктор инициализирует атрибуты класса. Конструктор создаст вид подвозки (toSchool) как "подвозка туда".
Предположите, что получаемые параметры корректны.

(2) Напишите конструктор класса **Transport ,** который получает только число учеников, пользующихся подвозкой – num .
Конструктор инициализирует атрибуты класса так, что получится "подвозка обратно" в поселок Aviv в четвертый день недели для num учеников.
Примечание: если параметр числа учеников (num) некорректен, атрибуту будет присвоено значение 1 .

(ב) Дан массив arr , типа **Transport** , который содержит подвозки, произведенные в течение недели. Массив не отсортирован в каком-либо порядке, и в нем нет значений null .

(1) Напишите внешний метод с названием dayReport на языке Java или DayReport на языке C# , который получает массив arr , целое число day , обозначающее день недели, и булев параметр forward .
Если forward равен true , метод вернет суммарное число учеников, которые воспользовались "подвозкой туда" в день day . Иначе (если forward равен false) метод вернет суммарное число учеников, которые воспользовались "подвозкой обратно" в день day .
Считайте, что полученный параметр day корректен (от 1 до 6).

(2) Напишите внешний метод с названием moreForward на языке Java или MoreForward на языке C# , который получает массив arr .
Метод выведет на экран номера дней недели, в которые число учеников, которые воспользовались "подвозкой туда" было больше числа учеников, которые воспользовались "подвозкой обратно".
Примечание: можно воспользоваться методом, который вы написали в подпункте (ב)(1).

/продолжение на странице 6/

5. Дан класс **User** – пользователь социальной сети, в котором три атрибута:

- name – имя пользователя, строкового типа (возможно, есть несколько человек с одинаковыми именами)
- id – идентификационный номер, целочисленного типа (уникальный номер)
- friends – массив, целочисленного типа, в котором содержатся идентификационные номера друзей пользователя (каждый номер появляется только один раз). Размер массива соответствует количеству друзей пользователя.

Считайте, что определены методы get/Get и set/Set для атрибутов этого класса.

(א) Напишите в классе **User** внутренний метод с названием mutual на языке Java или Mutual на языке C# , который получает другого пользователя (User) – other . Метод возвращает количество общих друзей двух пользователей.

(ב) Дан класс **SocialNetwork** – социальная сеть, в котором один атрибут:

- users – массив типа **User**

Этот массив не отсортирован в каком-либо порядке, каждый пользователь появляется в нем только один раз, и в нем нет значений null .

Напишите в классе **SocialNetwork** внутренний метод с названием exactOne на языке Java или ExactOne на языке C# , получающий пользователя (User) – other , который не появляется в массиве. Метод возвращает true , если в массиве существует по меньшей мере один пользователь, у которого есть в точности один общий друг с other . Иначе метод возвращает false .

Примечание: можно воспользоваться методом, который вы написали в пункте (א).

/продолжение на странице 7/

6. Положительное число num будет называться "идеальным числом", если выполняется следующее условие:

складывают все положительные и целые делители числа num (такие числа, на которые num делится без остатка), кроме него самого, и полученная сумма равна исходному числу.

Пример идеального числа: число 6 , потому что его делители: 1 , 2 , 3, и их сумма 6 .

Дополнительный пример идеального числа: число 28 , потому что его делители: 1 , 2 , 4 , 7 , 14 , и их сумма 28 .

Пример числа, которое **не является** идеальным числом: число 8 , потому что его делители: 1 , 2 , 4 , и их сумма 7 .

(א) **(1)** Напишите внешний метод с названием isPerfect на языке Java или IsPerfect на языке C# , который получает целое число – num , большее чем 2 . Метод возвращает true , если это идеальное число, а иначе возвращает false.

(2) Напишите внешний метод с названием thePerfects на языке Java или ThePerfects на языке C# , который получает два целых числа, больших чем 2 : low , high . Этот метод выведет на экран все идеальные числа между этими двумя числами (включительно). Считайте, что low меньше high или равно ему.

Примечание: можно воспользоваться методом, который вы написали в подпункте (1).

Пример: для low=4 и high=10 метод выведет на экран 6 .

Объяснение: от числа 4 и до 10 только число 6 – идеальное число (число 4 – не идеальное число, потому что сумма его делителей 1 , 2 – это 3 , число 5 – не идеальное число, потому что делится только на число 1 , и так далее).

(ב) Есть предположение, что не существует нечетного числа, которое является идеальным числом.

Напишите внешний метод с названием noOdd на языке Java или NoOdd на языке C# , который возвращает true , если на самом деле между всеми числами от 3 до 999999 (включительно) нет идеального числа, которое является нечетным. Иначе метод возвращает false .

Примечание: можно воспользоваться методом, который вы написали в подпункте (א)(1).

Желаем успеха!