Государство Израиль
Министерство просвещения
Тип экзамена: на аттестат зрелости
Время проведения экзамена: лето 2024 года
Номер вопросника: 899371

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ד, 2024
מספר השאלון: 899371

Обратите внимание: в этом вопроснике есть специальные инструкции.
Отвечайте на вопросы, следуя этим инструкциям.

שימו לב: בבחינה זו יש הנחיות מיוחדות.
יש לענות על השאלות על פי הנחיות אלה.

Компьютеры

מדעי המחשב

Указания

а. Продолжительность экзамена: 3 часа.

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

Раздел первый:	} (25×4)	
Раздел второй:		
Всего	–	100 баллов

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

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.

הוראות

א. משך הבחינה: שלוש שעות.

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

פרק ראשון	} (25×4)	
פרק שני		
סך הכול	–	100 נקודות

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

ד. הוראה מיוחדת:
את כל התוכניות שיש לכתוב בשפת מחשב כְּתבו 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 в общей сложности на четыре вопроса из первого и второго разделов (за каждый вопрос – 25 баллов).

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

1. Дан массив arr , содержащий положительные целые числа:

	0	1	2	3	4	5	6
arr	30	141	324	623	8004	458	6

Ниже приведен фрагмент программы, написанной на языках Java и C# :

Java	C#

int count=0;
for (int i=0; i< arr.length; i++)
{
  int num=arr[i]/10;
  int digit=num%10;
  if (digit==i)
  count++;
}
	int count=0;
	for (int i=0; i< arr.Length; i++)
	{
	  int num=arr[i]/10;
	  int digit=num%10;
	  if (digit==i)
	  count++;
	}

(א) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением фрагмента программы для массива arr и запишите значение переменной count после выполнения этого фрагмента.

Таблица трассировки должна содержать столбец для каждой из переменных:
count , i , num , digit .

(ב) Напишите пример массива arr длиной 4 , все ячейки которого содержат положительные (больше нуля) целые числа, для которого после выполнения этого фрагмента программы значение переменной count останется равным 0 .

(ג) Напишите пример массива arr длиной 4 , все ячейки которого содержат положительные (больше нуля) целые числа, для которого после выполнения этого фрагмента программы значение переменной count будет равно 4 .

/продолжение на странице 3/

2. В компьютеризированной системе одного банка "массив паролей" представляет собой массив целочисленного типа, длиной 10 , в котором банк хранит 10 последних паролей клиента (чтобы иметь возможность убедиться, что клиент не выберет их снова). Пароли хранятся упорядоченно, от самого нового к самому старому. Самый новый пароль (текущий) находится в ячейке 0 , а самый старый пароль находится в последней ячейке массива.
Когда появляется новый пароль, то самый старый из паролей (находящийся в последней ячейке массива) удаляется из массива, остальные пароли сдвигаются вправо, каждый в соседнюю с ним ячейку, а новый пароль занимает первую ячейку (с индексом 0).

Напишите внешний метод-предикат [פעולה בולאנית] с названием getPass на языке Java или GetPass на языке C# , который получает в качестве параметров заполненный массив паролей – arr и пароль целочисленного типа – password .
В случае, если полученный пароль – новый (то есть не совпадает ни с каким паролем из массива), метод добавит его в ячейку 0 массива arr (предварительно сдвинув остальные пароли на одну ячейку вправо) и вернет true . Иначе массив паролей останется без изменений и метод вернет false .

Пример: для представленного ниже массива arr и password = 300 массив останется без изменений и метод вернет false .

	0	1	2	3	4	5	6	7	8	9
arr	1223	134	245	300	111	101	777	900	195	1234

Объяснение: пароль 300 уже находится в массиве arr по индексу 3 .

Дополнительный пример: для того же массива и password = 8888 метод вернет true , а массив по окончании работы метода будет выглядеть следующим образом:

	0	1	2	3	4	5	6	7	8	9
arr	8888	1223	134	245	300	111	101	777	900	195

Объяснение: пароль 8888 не является одним из паролей, уже находящихся в массиве.

/продолжение на странице 4/

3. "Перемешанный массив" – это массив целочисленного типа, над которым была выполнена 30 раз следующая последовательность действий:

i. Генерируют два случайных числа, которые являются корректными индексами (в пределах размера массива).

ii. Значения ячеек, номера которых были сгенерированы случайным образом, меняются местами.

Примечание: если при генерировании выпали два одинаковых числа, оно не будет засчитано (то есть не будет считаться одной из 30 итераций).

Напишите внешний метод с названием shuffle на языке Java или Shuffle на языке C# , который принимает целочисленный массив – arr и перемешивает его. Метод выводит на печать, по порядку, значения ячеек после перемешивания (каждое значение на новой строке).

/продолжение на странице 5/

РАЗДЕЛ ВТОРОЙ

4. Дан класс **TourPackage –** пакет мобильной связи для полетов за границу, в котором пять атрибутов:

- id – идентификационный номер клиента, целочисленного типа
- price – стоимость пакета, целочисленного типа
- maxTime – входящее в пакет число минут разговора, целочисленного типа
- maxData – входящий в пакет объем интернет трафика (в гигабайтах), целочисленного типа
- extra – сумма платежа за превышение входящего в основной пакет количества минут разговора и объема трафика, целочисленного типа. Тариф за превышение составляет один шекель за каждую дополнительную минуту разговора и 2 шекеля за каждый дополнительный гигабайт трафика.

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

(א) (1) Напишите конструктор класса TourPackage **,** который принимает идентификационный номер клиента – id , стоимость пакета – price , входящее в пакет число минут разговора – maxTime , входящий в пакет объем интернет трафика – maxData .
Конструктор инициализирует атрибуты класса. Атрибут extra по умолчанию принимает значение 0 .

(2) Напишите внутрений метод класса TourPackage с названием setExtra на языке Java или SetExtra на языке C# , который получает количество минут разговора, фактически использованных клиентом за рубежом – minutes (целочисленного типа), и объем интернет трафика, фактически использованный клиентом за границей – usage (целочисленного типа).
Метод определяет значение атрибута extra в зависимости от превышения использования основного пакета (если оно есть) и от тарифов за превышение.
Примечание: extra равен 0 перед началом работы метода.

Дан заполненный массив – packages , типа TourPackage **,** который содержит данные о разных клиентах (после того как сумма платежа за превышение использования основного пакета была вычислена и присвоена атрибуту extra).

(ב) (1) Напишите внешний метод с названием calculate на языке Java или Calculate на языке C# , который получает массив packages и возвращает число клиентов, которые должны заплатить дополнительно за превышение использования основного пакета.

(2) Напишите внешний метод с названием customers на языке Java или Customers на языке C# , который получает массив packages . Метод возвращает массив целочисленного типа, содержащий идентификационные номера клиентов (id), которые должны заплатить дополнительно за превышение использования основного пакета (размер массива равен числу клиентов, превысивших использование). Порядок элементов в массиве не имеет значения.

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

/продолжение на странице 6/

5. Дан класс **Lesson** – урок, в котором четыре атрибута:

- id – код изучаемого предмета, целочисленного типа
- hh – час начала урока (от 8 до 17), целочисленного типа
- mm – минута начала урока (от 0 до 59), целочисленного типа
- duration – длительность урока в минутах, целочисленного типа

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

Примечание: уроки по одному и тому же предмету (id) могут проходить несколько раз в течение дня.

Дан заполненный массив уроков – arr типа Lesson , который представляет учебный день. Это **неупорядоченный** массив.

(א) (1) Напишите в классе Lesson внутренний метод с названием isLater на языке Java или IsLater на языке C# , который получает другой урок – other (типа Lesson). Метод возвращает true , если данный урок начинается позже того времени, в которое начинается другой урок (other), а иначе возвращает false .

(2) Напишите внешний метод с названием last на языке Java или Last на языке C# , который получает массив arr и выводит на печать код предмета (id), урок по которому начинается последним в этот учебный день.

Предположите, что существует только один урок, который удовлетворяет этому условию.

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

(ב) (1) Напишите внешний метод с названием sumDuration на языке Java или SumDuration на языке C# , который получает массив arr и код предмета – id . Метод возвращает суммарное число минут всех уроков с кодом предмета id , которые проходят в этот учебный день.

Например: если урок по предмету id длится 40 минут и в этот день есть еще один урок по тому же предмету, который длится 70 минут, то метод вернет 110 .

(2) Напишите внешний метод с названием longest на языке Java или Longest на языке C# , который получает массив arr . Метод возвращает код предмета (id), для которого суммарное число минут всех уроков в этот учебный день является наибольшим.

Предположите, что существует только один предмет, который удовлетворяет этому условию.

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

/продолжение на странице 7/

6. Дан внешний метод с названием digitSum на языке Java или DigitSum на языке C# , который получает целое число – num1 и возвращает сумму всех его цифр. Можно использовать этот метод, не реализуя его.

Например: для num1 = 961 метод вернет 16 (9 + 6 + 1) .

"Глубокая сумма цифр" какого-либо числа – это **однозначное** число, получаемое следующим образом: вычисляют сумму цифр снова и снова, пока не будет получено однозначное число.

Примеры:

"Глубокая сумма цифр" числа 5 равна 5 .

"Глубокая сумма цифр" числа 36 равна 9 (3+6) .

"Глубокая сумма цифр" числа 942378 равна 6 , как объяснено ниже:

9 + 4 + 2 + 3 + 7 + 8 = 33

3 + 3 = 6

(א) (1) Напишите внешний метод с названием deepSum на языке Java или DeepSum на языке C# , который получает целое неотрицательное число num1 и возвращает его "глубокую сумму цифр".

(2) "Глубокая сумма цифр" может быть нечетной (например, 5 , как в первом из приведенных выше примеров) или четной (например, 6 , как в третьем из приведенных выше примеров). Существует мнение, что между 1 и 999999 естъ больше чисел с четной "глубокой суммой цифр", чем с нечетной "глубокой суммой цифр".

Напишите еще один внешний метод с названием isCorrect на языке Java или IsCorrect на языке C# , который возвращает true , если это мнение верно, а иначе возвращает false.

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

Дан дополнительный внешний метод с названием digitExists на языке Java или DigitExists на языке C# , который получает целое число – num и цифру – digit . Метод возвращает true , если цифра digit появляется в числе num по меньшей мере один раз, а иначе возвращает false . Можно использовать этот метод, не реализуя его.

(ב) Напишите внешний метод с названием inBoth на языке Java или InBoth на языке C# , который получает два целых числа: num1 и num2 . Метод возвращает true , если "глубокая сумма цифр" num1 появляется в num2 , а также "глубокая сумма цифр" num2 появляется в num1 , а иначе возвращает false .

Например, для num1 = 36 и num2 = 942378 метод вернет true , поскольку "глубокая сумма цифр" num1 (9) появляется в числе num2 (**9**42378) , а "глубокая сумма цифр" num2 (6) появляется в числе num1 (3**6**) .

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

Желаем успеха!