מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ד, 2024
מספר השאלון: 899371
תרגום לערבית (2)

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بچروت
موعد الامتحان: صيف 2024
رقم النّموذج: 899371
ترجمة إلى العربيّة (2)

انتبهوا: في هذا الامتحان توجد توجيهات خاصّة.
يجب الإجابة عن الأسئلة حسب التّوجيهات.

מדעי המחשב

הוראות

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.
פרק ראשון (25x4)
פרק שני
סך הכול 100 נקודות

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

ד. הוראות מיוחדות:
את כל התוכניות שיש לכתוב בשפת מחשב כתבו בשפה אחת בלבד — Java או C#.

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

علم الحاسوب

تعليمات

أ. مدّة الامتحان: ثلاث ساعات.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج فصلان.
الفصل الأوّل (25x4)
الفصل الثّاني
المجموع 100 درجة

جـ. موادّ مساعدة يُسمح استعمالها: كلّ مادّة مساعدة، عدا الحاسبة التي توجد فيها إمكانيّة برمجة.

د. تعليمات خاصّة:
اكتبوا جميع البرامج التي يجب كتابتها بلغة حاسوب، بلغة واحدة فقط – Java أو C#.

ملاحظة: لن تُخصم درجات إذا كتبتم في البرامج حرفًا كبيرًا بدلًا من حرف صغير أو بالعكس.

يجب الكتابة في دفتر الامتحان فقط. يجب كتابة "مسوّدة" في بداية كلّ صفحة تُستعمَل مسوّدة.
كتابة أيّة مسوّدة على أوراق خارج دفتر الامتحان قد تسبّب إلغاء الامتحان.

الأسئلة في هذا النّموذج ترد بصيغة الجمع، ورغم ذلك يجب على كلّ طالبة وطالب الإجابة عنها بشكل فرديّ.

ב ה צ ל ח ה !

نتمنّى لكم النّجاح!

الأسئلة

في هذا النموذج فصلان.

يجب الإجابة عمّا مجموعه أربعة أسئلة من الفصلين الأوّل والثّاني (لكلّ سؤال – 25 درجة).

ملاحظة: في كلّ سؤال يُطلب فيه استقبال، لا حاجة لفحص سلامة المدخَلات.
للّذين يحلُّون بلغة Java: في كلّ سؤال يُطلب فيه استقبال، افترِضوا أنّ الأمر التالي مكتوب في البرنامج:

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

الفصل الأوّل

1. معطاة المصفوفة arr، التي تحوي أعدادًا صحيحة موجبة.

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

أمامكم قطعة برنامج، مكتوبة بلغة Java وبلغة 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++;
	}

أ. تتبَّعوا بواسطة جدول متابعة تنفيذ قطعة البرنامج بالنسبة للمصفوفة arr، واكتبوا قيمة المتغيِّر count بعد تنفيذ قطعة البرنامج.
يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من المتغيِّرات التالية: count ، i ، num ، digit .

ب. اكتبوا مثالًا لمصفوفة arr بِكِبَر 4، تحوي أعدادًا صحيحة موجبة (أكبر من 0) بحيث في نهاية قطعة البرنامج، قيمة المتغيِّر count تبقى 0 .

جـ. اكتبوا مثالًا لمصفوفة arr بِكِبَر 4 تحوي أعدادًا صحيحة موجبة (أكبر من 0) بحيث في نهاية قطعة البرنامج، تكون قيمة المتغيِّر count مساوية لِـ 4 .

/ يتبع في صفحة 3 /

2. "مصفوفة كلمات السرّ" في منظومة مُحَوْسَبَة لبنك معيَّن، هي مصفوفة من نمط صحيح بِكِبَر 10 ، يَحْفَظ فيها البنك عشر كلمات السرّ الأخيرة للزبون (ليتأكّد بأن لا يختارها الزبون مرّة أخرى). تُحفَظ كلمات السرّ حسب الترتيب من الأحدث إلى الأقدم. كلمة السرّ الأحدث (الحاليّة) موجودة في الخليّة 0، وكلمة السرّ الأقدم موجودة في الخليّة الأخيرة في المصفوفة.

عندما يتمّ تلقّي كلمة سرّ جديدة، تُمْحَى كلمة السرّ الأقدم من المصفوفة (كلمة السرّ التي في الخليّة الأخيرة في المصفوفة)، وبقيّة كلمات السرّ تتحرّك إلى اليمين، كلّ واحدة إلى الخليّة التي بجانبها، وكلمة السرّ الجديدة تظهر في الخليّة الأولى (المؤشّر 0).

اكتبوا عمليّة بوليانيّة خارجيّة باسم getPass بلغة Java أو GetPass بلغة C# ، تتلقّى مصفوفة كاملة لكلمات السرّ – arr، وكلمة سرّ – password ، من نمط صحيح.

إذا كانت كلمة السرّ التي تمّ تلقّيها هي جديدة (أي أنّها لا تُطابق أيّة كلمة سرّ تظهر في المصفوفة)، تُضيفها العمليّة إلى المصفوفة arr في الخليّة 0 (بعد تحريك بقيّة كلمات السرّ خليّةً واحدةً إلى اليمين) وتُعيد true.
خلاف ذلك، تبقى مصفوفة كلمات السرّ بدون تغيير، ويُعاد false .

مثال: بالنسبة للمصفوفة arr التي أمامكم وَ password = 300، تبقى المصفوفة بدون تغيير، وتُعيد العمليّة false .

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

الشرح: كلمة السرّ 300 سبق ظهورها في المصفوفة arr ، في المؤشّر 3.

مثال آخر: بالنسبة لنفس المصفوفة وَ password = 8888 ، تُعيد العمليّة true ، وتبدو المصفوفة على النحو التالي بعد نهاية العمليّة:

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

الشرح: كلمة السرّ 8888 ليست إحدى كلمات السرّ التي سبق ظهورها في المصفوفة.

/ يتبع في صفحة 4 /

3. "مصفوفة مخلوطة" هي مصفوفة من نمط صحيح، نُفِّذت فيها العمليّة الموصوفة أمامكم 30 مرّة:

i. يتمّ اقتراع رقمَي مؤشّرَيْن سليمَيْن (ضمن حدود كِبَر المصفوفة).

ii. القيمتان الموجودتان في الخليّتَيْن اللَّتَيْن اقتُرِعَ رقماهما تتبادلان فيما بينهما.

ملاحظة: إذا نَتَجَ في الاقتراع رقما مؤشّرَيْن متساويان، لا يتمّ احتساب القرعة (أي أنّها لا تُحصى كجزء من العمليّة المُنفَّذة 30 مرّة).

اكتبوا عمليّة خارجيّة باسم shuffle بلغة Java أو Shuffle بلغة C# ، تتلقّى مصفوفة من نمط صحيح – arr وتخلطها. تطبع العمليّة قيَم الخلايا حسب الترتيب بعد الخلط (كلّ قيمة في سطر جديد).

/ يتبع في صفحة 5 /

الفصل الثاني

4. معطاة الفئة **TourPackage** – رزمة خلويّة للمسافرين إلى خارج البلاد، ولها خمس صفات:

- id – رقم هويّة الزبون، من نمط صحيح.
- price – سعر الرزمة، من نمط صحيح.
- maxTime – عدد دقائق المكالمة المشمول في الرزمة، من نمط صحيح.
- maxData – حجم البيانات (بـ Giga) لتصفُّح الإنترنت، المشمول في الرزمة، من نمط صحيح.
- extra – مبلغ الدفع مقابل تجاوُز عدد الدقائق وحجم البيانات المشمولَيْن في الرزمة الأساسيّة، من نمط صحيح.
  تسعيرتا التجاوُز هما شيكل واحد مقابل كلّ دقيقة مكالمة إضافيّة، وشيكلان مقابل كلّ Giga إضافيّ في حجم تصفُّح الإنترنت.

افترِضوا أنّه توجد العمليّة get/Get لصفات الفئة.

أ. **(1)** اكتبوا عمليّة بنائيّة في الفئة TourPackage، تتلقّى رقم هويّة الزبون – id، وسعر الرزمة – price، وعدد دقائق المكالمة المشمولة في الرزمة – maxTime، وحجم البيانات – maxData.
تبتدئ العمليّة صفات الفئة. الصفة extra مُبتَدَأة لـ 0.

(2) اكتبوا عمليّة داخليّة في الفئة TourPackage باسم setExtra بلغة Java أو SetExtra بلغة C#، تتلقّى عدد دقائق المكالمة التي استعملها الزبون بالفعل خارج البلاد – minutes (من نمط صحيح)، وحجم التصفُّح الذي استعمله الزبون بالفعل خارج البلاد – usage (من نمط صحيح).
تُحدِّد العمليّة قيمة للصفة extra حسب تجاوُز الرزمة الأساسيّة (إذا كان هناك تجاوُز) ولتسعيرتَي التجاوُز.
ملاحظة: extra هي 0 قبل العمليّة.

معطاة مصفوفة كاملة – packages من نمط TourPackage، فيها معطيات عن زبائن مختلفين (بعد أن تمّ حساب وتحديث مبلغ الدفع مسبقًا مقابل تجاوُز الرزمة الأساسيّة في الصفة extra).

ب. **(1)** اكتبوا عمليّة خارجيّة باسم calculate بلغة Java أو Calculate بلغة C#، تتلقّى المصفوفة packages، وتُعيد عدد الزبائن الذين يجب أن يدفعوا مبلغًا إضافيًّا مقابل تجاوُز الرزمة الأساسيّة.

(2) اكتبوا عمليّة خارجيّة باسم customers بلغة Java أو Customers بلغة C#، تتلقّى المصفوفة packages.
تُعيد العمليّة مصفوفة من نمط صحيح، فيها أرقام هويّات الزبائن (id) الذين يجب أن يدفعوا مبلغًا إضافيًّا مقابل تجاوُز الرزمة الأساسيّة (كِبَر المصفوفة هو حسب عدد الأشخاص الذين تجاوزوا الرزمة). لا توجد أهمّيّة لترتيب الظهور في المصفوفة.
ملاحظة: يجب الاستعانة بالعمليّة التي كتبتموها في البند الفرعيّ "ب (1)".

/ يتبع في صفحة 6 /

5. معطاة الفئة **Lesson** – درس، ولها أربع صفات:

- id – كود الموضوع التعليميّ، من نمط صحيح.
- hh – الساعة التي يبدأ فيها الدرس (بين 8 وَ 17)، من نمط صحيح.
- mm – الدقيقة التي يبدأ فيها الدرس (بين 0 وَ 59)، من نمط صحيح.
- duration – مدّة الدرس بالدقائق، من نمط صحيح.

افترِضوا أنّ العمليّتين get/Get وَ set/Set موجودتان بالنسبة لصفات الفئة.

ملاحظة: يمكن أن يُجرى درس في نفس الموضوع (id) عدّة مرّات في نفس اليوم التعليميّ.

معطاة مصفوفة دروس كاملة – arr من نمط Lesson ليوم تعليميّ. المصفوفة **ليست مرتَّبة**.

أ. (1) اكتبوا في الفئة Lesson عمليّة داخليّة باسم isLater بلغة Java أو IsLater بلغة #C، تتلقّى درسًا آخر – other (من نمط Lesson). تُعيد العمليّة true إذا كان الدرس الحاليّ يبدأ في وقت متأخّر عن الزمن الذي يبدأ فيه الدرس الآخر (other)، خلاف ذلك تُعيد false.

(2) اكتبوا عمليّة خارجيّة باسم last تتلقّى المصفوفة arr، وتطبع كود الموضوع (id) الذي بدأ درسه آخِرًا في اليوم التعليميّ.

افترِضوا أنّه يوجد درس واحد فقط يُحقِّق هذا الشرط.

ملاحظة: يجب الاستعانة بالعمليّة التي كتبتموها في البند الفرعيّ "أ (1)".

ب. (1) اكتبوا عمليّة خارجيّة باسم sumDuration بلغة Java أو SumDuration بلغة #C، تتلقّى المصفوفة arr وكود موضوع – id. تُعيد العمليّة عدد الدقائق الكلّيّ لجميع الدروس في كود الموضوع id، التي تُجرى في نفس اليوم التعليميّ.

مثال: إذا استغرق درس في الموضوع id 40 دقيقة، وفي نفس اليوم استغرق درس آخر في نفس الموضوع 70 دقيقة، يُعاد 110.

(2) اكتبوا عمليّة خارجيّة باسم longest بلغة Java أو Longest بلغة #C، تتلقّى المصفوفة arr. تُعيد العمليّة كود الموضوع (id) الذي كان عدد الدقائق الكلّيّ لجميع دروسه في نفس اليوم التعليميّ هو الأكبر.

افترِضوا أنّه يوجد موضوع واحد فقط يُحقِّق هذا الشرط.

ملاحظة: يجب الاستعانة بالعمليّة التي كتبتموها في البند الفرعيّ "ب (1)".

/ يتبع في صفحة 7 /

6. معطاة عمليّة خارجيّة باسم digitSum بلغة Java أو DigitSum بلغة #C ، تتلقّى عددًا صحيحًا num1 – وتُعيد مجموع جميع الأرقام في العدد. يمكن استعمال العمليّة بدون تطبيقها.
مثال: بالنسبة لـ num1 = 961 ، تُعيد العمليّة 16 (1 + 6 + 9).

"مجموع الأرقام العميق" لِعَدد ما، هو عدد مكوَّن من **رقم واحد** يَنْتُج بالطريقة التالية:
يحسبون مجموع الأرقام مرّة تلو الأخرى، إلى أن يَنْتُج عدد مكوَّن من رقم واحد.
أمثلة:
"مجموع الأرقام العميق" للعدد 5 هو 5 .
"مجموع الأرقام العميق" للعدد 36 هو 9 (3+6) .
"مجموع الأرقام العميق" للعدد 942378 هو 6، كما هو مفصَّل فيما يلي:

9 + 4 + 2 + 3 + 7 + 8 = 33
3 + 3 = 6

أ. **(1)** اكتبوا عمليّة خارجيّة باسم deepSum بلغة Java أو DeepSum بلغة #C، تتلقّى عددًا صحيحًا num1 ليس سالبًا، وتُعيد "مجموع الأرقام العميق" له.

(2) "مجموع الأرقام العميق" يمكن أن يكون فرديًّا (مثلًا 5 كما في المثال الأوّل أعلاه) أو زوجيًّا (مثلًا 6 كما في المثال الثالث أعلاه). هناك مَن يدّعي أنّه في المجال الذي بين 1 وَ 999999 توجد أعداد لها "مجموع أرقام عميق" زوجيّ أكثر من الأعداد التي لها "مجموع أرقام عميق" فرديّ.
اكتبوا عمليّة خارجيّة باسم isCorrect بلغة Java أو IsCorrect بلغة #C، تُعيد true إذا كان الادّعاء صحيحًا، خلاف ذلك تُعيد false .
ملاحظة: يجب الاستعانة بالعمليّة التي كتبتموها في البند الفرعيّ "أ (1)".

معطاة عمليّة خارجيّة أخرى، باسم digitExists بلغة Java أو DigitExists بلغة #C ، تتلقّى عددًا صحيحًا num – ورقمًا – digit . تُعيد العمليّة true إذا كان الرقم digit يظهر في العدد num على الأقلّ مرّة واحدة، خلاف ذلك تُعيد false .
يمكن استعمال العمليّة بدون تطبيقها.

ب. اكتبوا عمليّة خارجيّة باسم inBoth بلغة Java أو InBoth بلغة #C، تتلقّى عددَيْن صحيحَيْن: num1 وَ num2 .
تُعيد العمليّة true إذا كان "مجموع الأرقام العميق" لِـ num1 يظهر في العدد num2 ، وكذلك "مجموع الأرقام العميق" لِـ num2 يظهر في العدد num1 ، خلاف ذلك تُعيد false .
مثلًا: بالنسبة لِـ num1 = 36 وَ num2 = 942378 ، تُعيد العمليّة true ، لأنّ "مجموع الأرقام العميق" لِـ num1 (9) يظهر في العدد num2 (**9**42378) ، وَ "مجموع الأرقام العميق" لِـ num2 (6) يظهر في العدد num1 (3**6**) .
ملاحظة: يجب الاستعانة بالعمليّة التي كتبتموها في البند الفرعيّ "أ (1)".

ב ה צ ל ח ה!
نتمنّى لكم النّجاح!