מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ו, 2026
מספר השאלון: 899371
תרגום לערבית (2)

מדעי המחשב

הוראות

א. משך הבחינה: שעתיים וחצי.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.
פרק ראשון — (20x2) — 40 נק'
פרק שני — (30x2) — 60 נק'
סך הכול — 100 נק'

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

ד. הוראה מיוחדת:
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.

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بچروت
موعد الامتحان: صيف 2026
رقم النّموذج: 899371
ترجمة إلى العربيّة (2)

علم الحاسوب

تعليمات

أ. مدّة الامتحان: ساعتان ونصف.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج فصلان.
الفصل الأوّل – (20x2) – 40 درجة
الفصل الثّاني – (30x2) – 60 درجة
المجموع – 100 درجة

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

د. تعليمة خاصّة:
اكتبوا جميع البرامج الّتي 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 عن أسئلة من الفصلين، حسب التعليمات في كلّ فصل.

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 (40 درجة)

أجيبوا عن اثنين من الأسئلة 1-3. (لكلّ سؤال – 20 درجة.)

1. أمامكم قطعة برنامج، مكتوبة باللُّغتين Java وَ C#.

Java	C#

int n = 23;
int steps = 0;
while (n > 0) {
  int root = (int)(Math.sqrt(n));
  int square = (int)(Math.pow(root, 2));
  n = n – square;
  steps ++;
}
System.out.println(steps);
	int n = 23;
	int steps = 0;
	while (n > 0) {
	  int root = (int)(Math.Sqrt(n));
	  int square = (int)(Math.Pow(root, 2));
	  n = n – square;
	  steps ++;
	}
	Console.WriteLine(steps);

أ. تتبَّعوا بواسطة جدول متابعة، تنفيذ قطعة البرنامج، واكتبوا العدد الذي تطبعه قطعة البرنامج.
يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من المتغيّرات التالية: square، root، steps، n.

ب. اكتبوا مثالًا لعدد أكبر من 1 موضوع في n (في بداية القطعة)، بالنسبة له تطبع قطعة البرنامج 1 (لا حاجة لعرض متابعة).

/ يتبع في صفحة 3 /

2. البارامتر max هو عدد صحيح موجب يُمثِّل سرعة السفر القصوى بالكيلومترات/الساعة المسموحة للمركبات في قطعة شارع معيّنة.
البارامتر min هو عدد صحيح موجب يُمثِّل سرعة السفر الصغرى بالكيلومترات/الساعة المسموحة للمركبات في نفس قطعة الشارع.

أ. اكتبوا عمليّة خارجيّة باسم getStatus بلغة Java أو GetStatus بلغة C# ، تتلقّى عددًا صحيحًا موجبًا – speed ، يُمثِّل سرعة السفر بالكيلومترات/الساعة لمركبة معيّنة في نفس قطعة الشارع، والبارامترَيْن max وَ min .
تُعيد العمليّة عددًا صحيحًا حسب الشروط التالية:
- إذا كانت سرعة سفر المركبة أعلى بِـ 10 كم/الساعة على الأقلّ، من سرعة السفر القصوى المسموحة، تُعيد العمليّة 1 .
- إذا كانت سرعة سفر المركبة أقلّ بِـ 10 كم/الساعة على الأقلّ، من سرعة السفر الصغرى المسموحة، تُعيد العمليّة 1– .
- إذا لم يتحقّق الشرطان أعلاه، تُعيد العمليّة 0 .

المصفوفة speeds هي مصفوفة من نمط صحيح تحوي معلومات عن سرعات سفر مركبات مختلفة في نفس قطعة الشارع المعيّنة (كلّ خليّة في المصفوفة تعرض سرعة سفر مركبة أخرى). المصفوفة ليست مرتّبة بترتيب معيّن.

ب. اكتبوا عمليّة خارجيّة باسم getAllStatus بلغة Java أو GetAllStatus بلغة C# تتلقّى المصفوفة speeds والبارامترَيْن max وَ min .
تُعيد العمليّة مصفوفة جديدة من نمط صحيح بِكِبَر مطابق للمصفوفة speeds ، تعرض في كلّ خليّة العدد 1 أو 1– أو 0 حسب المركبة الموجودة في نفس المؤشِّر في المصفوفة speeds (حسب الشروط التي في البند "أ")، في المؤشِّر المطابق لمؤشِّر المركبة التي في المصفوفة speeds .
مثال: بالنسبة لِـ max = 70 ، min = 30 ، والمصفوفة speeds :

	0	1	2	3	4	5	6
speeds	80	50	79	90	20	100	25

تُعاد المصفوفة التالية :

0	1	2	3	4	5	6
1	0	0	1	1–	1	0

ملاحظتان: – يجب استعمال العمليّة التي كتبتموها في البند "أ".
– لا تُغيِّروا المصفوفة speeds .

جـ. تحصل المركبة على مخالفة إذا كانت سرعة سفرها في قطعة الشارع هذه أعلى بِـ 10 كم/الساعة على الأقلّ، من السرعة القصوى المسموحة، أو أقلّ بِـ 10 كم/الساعة على الأقلّ، من السرعة الصغرى المسموحة.
اكتبوا عمليّة خارجيّة باسم mostFined بلغة Java أو MostFined بلغة C# ، تتلقّى المصفوفة speeds والبارامترَيْن max وَ min .
تُعيد العمليّة true إذا حصل على الأقلّ، نصف عدد المركبات في المصفوفة speeds على مخالفة، خلاف ذلك تُعيد العمليّة false .
مثال: بالنسبة للمصفوفة المعطاة في البند "ب"، تُعيد العمليّة true ، لأنّه من بين المركبات السبع في المصفوفة، أربع مركبات تحصل على مخالفة (المركبات في المؤشِّرات 0 ,3 ,4 ,5).
ملاحظة: يجب استعمال العمليّة التي كتبتموها في البند "ب".

/ يتبع في صفحة 4 /

3. في لعبة "**إصابة دقيقة**" يختارون مسبَقًا نصًّا سرّيًّا، مكوّنًا من الأحرف A–Z بدون ترتيب معيّن. كلّ حرف يمكن أن يظهر في النصّ السرّيّ مرّةً واحدةً على الأكثر (بحيث طول النصّ هو بين 1 وَ 26).

تُدار اللعبة على النحو التالي:

يُخمِّن اللاعب نصًّا معيّنًا (يعتقد اللاعب أنّه النصّ السرّيّ). بالنسبة لكلّ حرف في نصّ التخمين، يحصل اللاعب على درجة حسب مدى نجاح تخمينه:

- درجة "**دقيق**" تُعطى للتخمين الصحيح للحرف ولمكانه في النصّ السرّيّ.
- درجة "**إصابة**" تُعطى للتخمين الصحيح للحرف في النصّ السرّيّ، لكن في المكان غير الصحيح.
- درجة "**تضييع**" تُعطى لتخمين حرف ليس موجودًا بتاتًا في النصّ السرّيّ.

مثال: بالنسبة للتخمين "AGCF" للنصّ السرّيّ "FGAV" يحصل اللاعب على الدرجات التالية:

درجة "دقيق" واحدة للتخمين الصحيح للحرف G ولمكانه في النصّ السرّيّ (في المؤشِّر 1).

درجتَي "إصابة" بالنسبة للتخمين الصحيح للحرفين A وَ F في النصّ السرّيّ، لكن في مكانَيْن غير صحيحَيْن.

درجة "تضييع" واحدة بالنسبة لتخمين غير صحيح للحرف C، غير الموجود بتاتًا في النصّ السرّيّ.

اكتبوا عمليّة خارجيّة باسم guess بلغة Java أو Guess بلغة C#، تتلقّى البارامترَيْن:

نصًّا سرّيًّا – secretStr، ونصّ تخمين – playerStr.

افترِضوا أنّ النصَّيْن سليمان (أي تظهر فيهما فقط الأحرف A–Z، وكلّ حرف يظهر في كلّ نصّ مرّة واحدة على الأكثر).

إذا كان طول نصّ التخمين مختلفًا عن طول النصّ السرّيّ، تطبع العمليّة رسالة خطأ "Size Error".

إذا كان طول نصّ التخمين مطابقًا لطول النصّ السرّيّ، تطبع العمليّة عدد درجات الـ "دقيق"، وعدد درجات الـ "الإصابة" وعدد درجات الـ "تضييع" بالنسبة لتخمين اللاعب (كلّ طباعة في سطر جديد).

/ يتبع في صفحة 5 /

الفصل الثاني (60 درجة)

أجيبوا عن اثنين من الأسئلة 4-6 . (لكلّ سؤال – 30 درجة .)

4. معطاة الفئة **Order** ، التي تمثّل تفاصيل طلبيّة لمُنتَج معيّن، ولها أربع صفات :

- productID – الكود المميِّز للمُنتَج، من نمط صحيح بين 0 وَ 99 (بما في ذلك 0 وَ 99) .
- price – سعر المُنتَج، من نمط عدد حقيقيّ .
- quantity – كمّيّة وحدات المُنتَج، من نمط صحيح .
- delivery – هل يشمل شحنًا إلى البيت، من نمط بوليانيّ (true – بالنسبة للطلبيّة التي تشمل شحنًا، false – بالنسبة للطلبيّة التي لا تشمل شحنًا) .

الزبون الذي يطلب شحنًا إلى البيت يجب أن يدفع **مبلغًا إضافيًّا** قدره 100 شيكل. إذا طُلِبت 10 وحدات على الأقلّ، في نفس الطلبيّة، وبالإضافة إلى ذلك، مبلغ الطلبيّة (لا يشمل الشحن) هو أعلى من 170.00 شيكل – الشحن هو **مجّانيّ** .

افترِضوا أنّه توجد عمليّات get/Get وَ set/Set لصفات الفئة .

أ. اكتبوا في الفئة **Order** عمليّة بنائيّة تتلقّى الكود المميِّز للمُنتَج – productID وسعر المُنتَج – price .
تبتدئ العمليّة صفات الفئة بحيث يتمّ تلقّي طلبيّة للمُنتَج – productID ، بسعر المُنتَج – price ، وبكمّيّة وحدة واحدة مع شحن إلى البيت .

ب. (1) اكتبوا في الفئة **Order** عمليّة داخليّة باسم totalPrice بلغة Java أو TotalPrice بلغة C# ، تُعيد عددًا حقيقيًّا للمبلغ الكلّيّ للدّفع لجميع المنتَجات في الطلبيّة **بدون رسوم الشحن** .

(2) اكتبوا في الفئة **Order** عمليّة داخليّة باسم totalCost بلغة Java أو TotalCost بلغة C# ، تُعيد عددًا حقيقيًّا للمبلغ **يشمل رسوم الشحن**، الذي يجب على الزبون أن يدفعه (إذا كانت الطلبيّة لا تشمل شحنًا إلى البيت أو كان الشحن مجّانيًّا، يُعاد مبلغ الطلبيّة بدون رسوم شحن) .
ملاحظة : يجب استعمال العمليّة التي كتبتموها في البند الفرعيّ "ب (1)" .

جـ. معطاة مصفوفة – allOrders من نمط **Order** ، تحوي الطلبيّات التي كانت في أسبوع معيّن. المصفوفة ليست مرتَّبة حسب ترتيب معيّن، وليس فيها قيَم null .
اكتبوا عمليّة خارجيّة باسم sumQuantity بلغة Java أو SumQuantity بلغة C# ، تتلقّى المصفوفة allOrders .
تُعيد العمليّة مصفوفة جديدة من نمط صحيح بِكِبَر 100 ، تظهر في كلّ خليّة كمّيّة الوحدات الكلّيّة التي طُلِبَت من المُنتَج، الذي الكود المميِّز له مطابق لمؤشِّر الخليّة .
مثلًا، إذا طُلِبَت من مُنتَج الكود المميِّز له هو 0 – في طلبيّة معيّنة 15 وحدة، وفي طلبيّة أخرى 20 وحدة، وفي طلبيّة إضافيّة 3 وحدات – يظهر في الخليّة التي مؤشِّرها 0 العدد 38 .

/ يتبع في صفحة 6 /

5. معطاة الفئة **Birthdate** – تاريخ الولادة، ولها ثلاث صفات:

- day – اليوم في الشهر (عدد بين 1 وَ 31)، من نمط صحيح.
- month – الشهر (عدد بين 1 وَ 12)، من نمط صحيح.
- year – السنة، من نمط صحيح (عدد موجب).

افترِضوا أنّه توجد عمليّات get/Get وَ set/Set لصفات الفئة.

أ. اكتبوا في الفئة **Birthdate** عمليّة داخليّة باسم sameDate بلغة Java أو SameDate بلغة C# ، تتلقّى كائنًا آخر – other من نمط **Birthdate** .
تُعيد العمليّة true إذا كان تاريخا الولادة متطابقَيْن، خلاف ذلك تُعيد false .

في شركة الهايتك "شبكة اجتماعيّة"، طوَّروا منظومة تبحث عن تشابُه بين زبائن الشركة. لهذا الغرض، بَنَت الشركة الفئة **Client** – زبون، ولها ثلاث صفات:

- firstNameInitial – الحرف الأوّل في الاسم الشخصيّ (حرف كبير باللّغة الإنجليزيّة)، من نمط char
- lastNameInitial – الحرف الأوّل في اسم العائلة (حرف كبير باللّغة الإنجليزيّة)، من نمط char
- birthdate – تاريخ الولادة، من نمط **Birthdate** .

افترِضوا أنّه توجد عمليّات get/Get وَ set/Set لصفات الفئة.

ب. (1) الأحرف الأولى لاسمَي زبونَيْن تكون متطابقة، إذا كان الحرف الأوّل في اسمَيْهما الشخصيّين وكذلك الحرف الأوّل في اسمَي عائلتَيْهما متطابقة.
اكتبوا في الفئة **Client** عمليّة داخليّة باسم sameInitials بلغة Java أو SameInitials بلغة C# ، تتلقّى كائنًا آخر – other من نمط **Client** . تُعيد العمليّة true إذا كانت الأحرف الأولى لاسمَي الزبونَيْن متطابقة. خلاف ذلك تُعيد العمليّة false .

(2) في هذه المنظومة، يُعَدُّ الزبونان "متشابهَيْن"، إذ كانت الأحرف الأولى في اسمَيْهما متطابقة، **وكذلك** إذا كان تاريخا ولادتهما متطابقَيْن (إذا تحقّق شرط واحد فقط، الزبونان ليسا "متشابهَيْن").
اكتبوا في الفئة **Client** عمليّة داخليّة باسم areSame بلغة Java أو AreSame بلغة C# ، تتلقّى كائنًا آخر – other من نمط **Client** . تُعيد العمليّة true إذا كان الزبونان "متشابهَيْن"، خلاف ذلك تُعيد العمليّة false .
يجب استعمال العمليّتَيْن اللتين كتبتموهما في البند "أ" وفي البند الفرعيّ "ب (1)".

جـ. في هذه المنظومة، يُعَدُّ الزبون "مميّزًا"، إذا لم يكن أيّ زبون آخر "مشابهًا" له (كما عُرِّفَ في البند الفرعيّ "ب (2)").
اكتبوا عمليّة خارجيّة باسم uniques بلغة Java أو Uniques بلغة C# ، تتلقّى مصفوفة – clients من نمط **Client** لجميع الزبائن في المنظومة. المصفوفة ليست مرتّبة بحسب ترتيب معيّن، ولا توجد فيها قِيَم null . تطبع العمليّة الحرف الأوّل في الاسم الشخصيّ للزبائن "المميّزين" في المصفوفة (كلّ طباعة في سطر جديد).
يجب استعمال العمليّة التي كتبتموهما في البند الفرعيّ "ب (2)".

/ يتبع في صفحة 7 /

6. أ. طبِّقوا العمليّة التي أمامكم:

Java – public static int leftDigits (int num, int k)

C# – public static int LeftDigits (int num, int k)

num هو عدد موجب، وَ k هو عدد موجب أصغر من عدد أرقام num (أي طول العدد num).

تُعيد العمليّة العدد الممثَّل بواسطة k الأرقام التي من الجهة اليُسرى للعدد num .

مثال: بالنسبة لِـ k = 5 وَ num = **26004**3100 ، تُعيد العمليّة العدد 26004 .

ب. طبِّقوا العمليّة التي أمامكم:

Java – public static int rightDigits (int num, int k)

C# – public static int RightDigits (int num, int k)

num هو عدد موجب، وَ k هو عدد موجب أصغر من عدد أرقام num (أي طول العدد num).

تُعيد العمليّة العدد الممثَّل بواسطة k الأرقام التي من الجهة اليمنى للعدد num (افترِضوا أنّ العدد المُعاد سليم، ولا يبدأ بِـ 0).

مثال: بالنسبة لِـ k = 5 وَ num = 2600**43100** ، تُعيد العمليّة العدد 43100 .

جـ. عدد صحيح موجب يُسمّى "كابِركار متوازن"، عندما يُحقِّق الشّرط التالي:
نُربِّع العدد، ونُقسِّم النتيجة في الوسط إلى قسمَيْن طولُهما متساوٍ (أي عدد الأرقام في النتيجة هو زوجيّ، وعدد الأرقام في كلّ قسم متطابق)، ومجموع القسمين مساوٍ للعدد الأصليّ.

مثالان: العددان **9** وَ **4950** هما "كابِركار متوازن".

الشرح:

العدد	تربيع العدد	القسم الأيمن	القسم الأيسر	المجموع
9	$9^2 = 81$	1	8	$8 + 1 = \mathbf{9}$
4950	$4950^2 = 24502500$	2500	2450	$2450 + 2500 = \mathbf{4950}$

اكتبوا عمليّة خارجيّة باسم isKaprekar بلغة Java أو IsKaprekar بلغة C# ، تتلقّى عددًا صحيحًا موجبًا – num .

تُعيد العمليّة true إذا كان num هو عدد "كابِركار متوازن"، خلاف ذلك تُعيد العمليّة false .

يجب استعمال العمليّتَيْن اللتين كتبتموهما في البندَيْن "أ – ب".

افترِضوا أنّه إذا رَبَّعْنا num ، سيكون عدد الأرقام في النتيجة زوجيًّا، وسيكون قسما النتيجة سليمَيْن.

בהצלחה!
نتمنّى لكم النّجاح!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.
حقوق الطّبع محفوظة لدولة إسرائيل.
النّسخ أو النّشر ممنوعان إلّا بإذن من وزارة التّربية والتّعليم.