**מדינת ישראל**
**משרד החינוך**

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

# מדעי המחשב

## הוראות

א. משך הבחינה: שעתיים וחמישים דקות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.
יש לבחור בארבע שאלות סך הכול.
לכל שאלה — 25 נקודות; 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
כתבו בשפה אחת בלבד — Java או C#.

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

**دولة إسرائيل**
**وزارة التّربية والتّعليم**

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

# علم الحاسوب

## تعليمات

أ. مدّة الامتحان: ساعتان وخمسون دقيقة.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج فصلان.
يجب اختيار ما مجموعه أربعة أسئلة.
لكلّ سؤال – 25 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 عمّا مجموعه أربعة أسئلة من الفصلين؛ الأوّل والثاني.** (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 المصفوفتان arr وَ flag بِكِبَر متساوٍ، وقطعة برنامج مكتوبة باللغتين Java وَ C#.

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| arr | 4 | 3 | 0 | 5 |

| | | | | |
|---|---|---|---|---|
| flag | 1 | 0 | 1 | 1 |

| **Java** | **C#** |
|---|---|
| int sum = 0;<br>int d = 1;<br>for (int i = 0; i < arr.length; i++)<br>{<br>  if (flag[i] == 1)<br>  {<br>  sum = sum + arr[i] * d;<br>  }<br>  d = d * 10;<br>}<br>System.out.println(sum); | int sum = 0;<br>int d = 1;<br>for (int i = 0; i < arr.Length; i++)<br>{<br>  if (flag[i] == 1)<br>  {<br>  sum = sum + arr[i] * d;<br>  }<br>  d = d * 10;<br>}<br>Console.WriteLine(sum); |

**أ.** تتبَّعوا بواسطة جدول متابعة، تنفيذ قطعة البرنامج، واكتبوا ماذا تطبع العمليّة.
يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من المتغيّرات: sum, d, i.

**ب.** اكتبوا مثالًا لمصفوفتين arr وَ flag بِكِبَر 3، تطبع قطعة البرنامج بالنسبة لهما العدد 301.

/ يتبع في صفحة 3 /

**2.** طبِّقوا العمليّة التي أمامكم:

Java - public static int longestIncrease(int[] arr)

C# - public static int LongestIncrease(int[] arr)

تتلقّى العمليّة المصفوفة arr لدرجات حرارة في فترة معيّنة ( ترتيب الأيّام من اليسار إلى اليمين ) . معلوم أنّ درجة الحرارة تختلف بين كلّ يومين متجاورين.

تُعيد العمليّة طول التسلسُل الأطول **للارتفاع** في درجة الحرارة يوميًّا.

مثال:
بالنسبة للمصفوفة التالية، تُعاد القيمة 4:

arr [20,22,25,24,23,26,27,28,25]

الشرح:
- تسلسُل الارتفاع الأوّل هو – 20,22,25 ، وطوله هو 3 .
- تسلسُل الارتفاع الثاني هو – 24 ، وطوله هو 1.
- تسلسُل الارتفاع الثالث هو – 23,26,27,28 ، وطوله هو **4** .
- تسلسُل الارتفاع الرابع هو 25 ، وطوله هو 1.

افترِضوا أنّه توجد في المصفوفة 3 خلايا على الأقلّ.

**3.** عدد موجب num يُسمّى "**عددًا متدرِّجًا**" إذا كانت أرقامه ( من اليسار إلى اليمين ) تُكوِّن متوالية فيها كلّ رقم هو أكبر بِـ 1 بالضبط من الرقم الذي عن يساره.

مثال لعدد متدرِّج: العدد 345 هو عدد متدرِّج 5 ← 4 ← 3 ( كلّ رقم أكبر بِـ 1 من الرقم الذي عن يساره ).
أمثلة إضافيّة لأعداد متدرِّجة: 78, 678, 1234
أمثلة لأعداد ليست أعدادًا متدرِّجة: 543, 343, 25

**أ.** اكتبوا عمليّة خارجيّة باسم isStep بلغة Java أو IsStep بلغة C# ، تتلقّى عددًا من نمط صحيح num أكبر أو يساوي 10 . تُعيد العمليّة true إذا كان num هو عدد متدرِّج، خلاف ذلك، تُعيد العمليّة false .

**ب.** اكتبوا عمليّة خارجيّة باسم theStep بلغة Java أو TheStep بلغة C# ، تتلقّى عددًا من نمط صحيح n أكبر أو يساوي 10 . تطبع العمليّة جميع الأعداد المتدرِّجة الموجودة من 10 وحتّى n ( بما في ذلك 10 وَ n ).
ملاحظة: يجب استعمال العمليّة التي كُتبت في البند "أ" .

/ يتبع في صفحة 4 /

## الفصل الثاني

**4.** معطاة الفئة **SmartDevice** ، التي تمثّل جهازًا كهربائيًّا في منظومة "البيت الذكيّ" . لهذه الفئة الصفات التالية :

- deviceName – اسم الجهاز، من نمط نصّ .
- wattage – استهلاك الجهاز للكهرباء ( واط )، من نمط صحيح .
- isActive – تشير هل الجهاز يعمل في الوقت الحاليّ، من نمط بوليانيّ ( إذا كان الجهاز يعمل في الوقت الحاليّ true ، وإذا لم يكن يعمل false ) .

افترِضوا أنّه توجد عمليّات Get/get وَ Set/set بالنسبة لجميع صفات الفئة .

أ. **(1)** اكتبوا عمليّة بنائيّة للفئة SmartDevice تتلقّى اسم الجهاز – name واستهلاك الكهرباء – watts . تبتدئ العمليّة قيَم الصفات حسب القيَم المتلقّاة وتحدِّد أنّ الجهاز يعمل في الوقت الحاليّ .
افترِضوا أنّ البارامترات المتلقّاة سليمة .

**(2)** جهاز كهربائيّ هو "High Energy" إذا كان استهلاكه للكهرباء أعلى من – 1,000 واط .
اكتبوا في الفئة SmartDevice عمليّة داخليّة باسم isHighEnergy بلغة Java أو IsHighEnergy بلغة C# .
تُعيد العمليّة true إذا كان الجهاز هو "High Energy" ، خلاف ذلك تُعيد العمليّة false .

**ب.** معطاة مصفوفة باسم devices من نمط SmartDevice تحوي معلومات عن جميع الأجهزة الكهربائيّة الموصولة بالبيت .
المصفوفة تحوي فقط كائنات ( لا توجد فيها قيَم null ) وهي ليست مرتّبة .

**(1)** اكتبوا عمليّة خارجيّة باسم amount بلغة Java أو Amount بلغة C# ، تتلقّى المصفوفة devices .
تُعيد العمليّة عدد الأجهزة في المصفوفة التي تعمل في الوقت الحاليّ وأيضًا تُعتبَر أجهزة "HighEnergy" ( حسب التعريف في البند الفرعيّ "أ (2)" )، وتطبع معدّل الاستهلاك الكهربائيّ لهذه الأجهزة .

**(2)** اكتبوا عمليّة خارجيّة باسم getNamesDevices بلغة Java أو GetNamesDevices بلغة C# ، تتلقّى المصفوفة devices . تُعيد العمليّة مصفوفة مملوءة ( بدون خلايا فارغة ) من نمط نصّ، تحوي أسماء (deviceName) الأجهزة في المصفوفة التي تعمل في الوقت الحاليّ، وأيضًا تُعتبَر أجهزة "High Energy" .
ملاحظة : يجب استعمال العمليّة التي كتبتموها في البند الفرعيّ "ب (1)" .

**5.** معطاة الفئة **Student** التي تُمثِّل طالبًا في مؤسَّسة أكاديميّة، ولها ثلاث صفات:

- name – اسم الطالب، من نمط صحيح.
- id – رقم هويّة الطالب، من نمط صحيح (عدد خاصّ).
- courseIDs – مصفوفة من نمط صحيح، فيها أرقام تعريفيّة للدورات التي سجّل فيها الطالب، بدون ترتيب معيّن.
  كِبَر المصفوفة هو بالتلاؤم مع عدد الدورات التي سجّل فيها الطالب (كلّ رقم تعريفيّ لدورة يظهر مرّة واحدة فقط في المصفوفة).

**أ.** **(1)** اكتبوا في الفئة Student عمليّة داخليّة باسم mutual بلغة Java أو Mutual بلغة C# ، تتلقّى طالبًا آخر – other من نمط Student. تُعيد العمليّة عدد الدورات المشترَكة بين الطالب الحاليّ والطالب الآخر (أي، عدد الدورات التي أرقامها التعريفيّة تظهر في المصفوفتين courseIDs).

**(2)** اكتبوا في الفئة Student عمليّة داخليّة باسم sameCourses بلغة Java أو SameCourses بلغة C# ، تتلقّى طالبًا آخر – other من نمط Student. تُعيد العمليّة true إذا كان الطالب الحاليّ والطالب الآخر مسجَّلَيْن بالضبط في نفس الدورات (بدون أهمّيّة للترتيب في المصفوفة). خلاف ذلك، تُعيد العمليّة false.

**ب.** معطاة الفئة **University** – جامعة، لها صفة واحدة:

المصفوفة – arr من نمط Student. المصفوفة ليست مرتّبة حسب أيّ ترتيب، كلّ طالب يظهر مرّة واحدة فقط، ولا توجد فيها قيَم null.

اكتبوا في الفئة University عمليّة داخليّة باسم common بلغة Java أو Common بلغة C# ، تتلقّى طالبًا – student من نمط Student لا يظهر في المصفوفة. تُعيد العمليّة true إذا كان في المصفوفة طالب توجد له وَلِـ student بالضبط نفس الدورات (لا يجب أن تكون بنفس الترتيب). خلاف ذلك، تُعيد العمليّة false .

ملاحظة: يجب استعمال العمليّة التي كتبتموهما في البند الفرعيّ "أ (2)".

/ يتبع في صفحة 6 /

**6.** **أ.** نصّ هو نصّ متناسق، إذا كان النصّ يُقرأ بالطريقة نفسها من اليسار إلى اليمين، ومن اليمين إلى اليسار.
اكتبوا عمليّة خارجيّة باسم isPalindrom بلغة Java أو IsPalindrom بلغة C# ، تتلقّى نصًّا – str ليس فارغًا.
تُعيد العمليّة true إذا كان النصّ هو نصّ متناسق، خلاف ذلك، تُعيد العمليّة false.

مثال:

- بالنسبة لِـ str = "12c21" ، يُعاد true .
- بالنسبة لِـ str ="a"= ، يُعاد true .
- بالنسبة لِـ str = "aa345" ، يُعاد false .
- بالنسبة لِـ str = "abab" ، يُعاد false .

**ب.** يسمّى النصّ "**نصّ زيبرا**" إذا تحقّقت فيه ثلاثة الشروط التالية:

1. النصّ هو نصّ متناسق.
2. النصّ مركّب بالضبط من **رمزين مختلفين**.
3. كلّ رمز في النصّ **يختلف عن الرمز الذي يُجاوره** (أي، لا يوجد رمزان متطابقان متجاوران).

أمثلة:

- النصّ "12121" هو زيبرا – لأنّه نصّ متناسق، ويوجد فيه بالضبط رمزان مختلفان، وكلّ رمز يختلف عن الرمز الذي يجاوره. كذلك النصّ "a1a1a" والنصّ "hihihih" هما زيبرا.
- النصّ "abab" ليس زيبرا لأنّه ليس نصًّا متناسقًا.
- النصّ "7" ليس زيبرا لأنّه يوجد فيه رمز واحد فقط.
- النصّ "aabaa" ليس زيبرا، لأنّه يوجد فيه رمزان متجاوران متطابقان.

اكتبوا عمليّة خارجيّة باسم isZebra بلغة Java أو IsZebra بلغة C# ، تتلقّى نصًّا – str ليس فارغًا. تُعيد العمليّة true إذا كان النصّ هو زيبرا، خلاف ذلك، تُعيد العمليّة false.

ملاحظة: يجب استعمال العمليّة التي كتبتموها في البند "أ".

**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.