מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ג, 2023
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. הוראות מיוחדות:
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: صيف 2023
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 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 عن أسئلة من الفصلين، حسب التعليمات في كلّ فصل.

ملاحظة: في كلّ سؤال يُطلب فيه استقبال، لا حاجة لفحص سلامة المدخَلات.

للّذين يحلّون 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. معطاة مصفوفة arr، تحوي أعدادًا صحيحة:

	0	1	2	3	4	5	6
arr	**3**	**4**	**9**	**6**	**8**	**4**	**2**

معطاة قطعة برنامج، مكتوبة بلغتَي Java وَ C#:

Java	C#

int sum = arr[0];
for (int i=1; i< arr.length; i++)
{
  arr[i] = arr[i] - sum;
  sum += arr[i];
}
	int sum = arr[0];
	for (int i=1; i< arr.Length; i++)
	{
	  arr[i] = arr[i] - sum;
	  sum += arr[i];
	}

أ. تتبَّعوا بواسطة جدول متابعة تنفيذ قطعة البرنامج، واكتبوا محتوى المصفوفة arr بعد تنفيذ قطعة البرنامج.

يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من المتغيّرات التالية: sum , arr[i] , i .

ب. اكتبوا مثالًا لمصفوفة arr بكِبَر 5 التي بعد تنفيذ قطعة البرنامج تحوي جميع خلاياها أعدادًا موجبة (أي: أعدادًا أكبر من صفر).

/ يتبع في صفحة 3 /

2. في لعبة رَمْي إلى الهدف، يرمي الشخص عددًا من الأسهم نحو الهدف. الدرجات بالنسبة لكلّ رمية هي عدد صحيح من 0 درجات (رمية ضائعة بشكل كامل) وحتّى 10 درجات (إصابة في مركز الهدف).
ينجح الشخص في اللعبة إذا تحقّقت على الأقلّ واحدة من الحالات التالية:

- في معدّل العلامات، حصل على أعلى من 5.0 درجات.
- في أغلب الرميات حاز على 6 درجات فصاعدًا.
- في رمية واحدة أو أكثر حاز على 10 درجات.

مثال: رمى شخص خمسة أسهم وحاز على الدرجات: 5, 7, 9, 3, 3. في المعدّل حاز على 5.4 درجات ((5 + 7 + 9 + 3 + 3) / 5)، لذا فقد نجح في اللعبة.

مثال إضافيّ: رمى شخص ستّة أسهم وحاز على الدرجات: **7**, **6**, 1, **7**, **8**, 0. في أربع رميات من ضمن الستّ (أي في أغلب رمياته) حاز على 6 درجات فصاعدًا، لذا فقد نجح في اللعبة.

مثال إضافيّ: رمى شخص ثلاثة أسهم وحاز على الدرجات: 2, **10**, 0. في إحدى الرميات حاز على 10 درجات، لذا فقد نجح في اللعبة.

اكتبوا عمليّة خارجيّة باسم success بلغة Java أو Success بلغة C# ، تتلقّى عددًا صحيحًا – num .
num يشير إلى عدد الأسهم التي رماها الشخص نحو الهدف.
تستقبل العمليّة عدد الدرجات التي حاز عليها الشخص بالنسبة لكلّ سهم رماه.
تُعيد العمليّة true إذا نجح الشخص في اللعبة، خلاف ذلك تُعيد العمليّة false .

/ يتبع في صفحة 4 /

3. قرّر مجلس البلديّة في مدينة معيّنة أن يمنح سكّان المدينة تخفيضات من مدفوعات مختلفة حسب المعايير التالية:

- الساكن في المدينة الذي عمره 70 فصاعدًا، يستحقّ تخفيضًا من الدفع للأرنونا.
- الساكن في المدينة الذي لديه 3 أولاد فصاعدًا، يستحقّ تخفيضًا من الدفع لرياض الأطفال.
- الساكن في المدينة الذي مكان عمله في المدينة، يستحقّ تخفيضًا من الدفع مقابل صفّ (رَكْن) السيّارة في المدينة.

المعلومات عن مدى استحقاق هذه التخفيضات ممثَّلة بواسطة عدد مكوَّن من ثلاثة أرقام، مركَّب من الرقمين 1 وَ 2 فقط.
الرقم 1 يشير إلى استحقاق التخفيض، والرقم 2 يشير إلى عدم استحقاق التخفيض.
الرقم الأوّل من اليسار (رقم المئات) يشير إلى الاستحقاق أو عدم الاستحقاق للتخفيض من الدفع للأرنونا،
والرقم الأوسط (رقم العشرات) يشير إلى الاستحقاق أو عدم الاستحقاق للتخفيض من الدفع لرياض الأطفال،
والرقم الأخير (رقم الآحاد) يشير إلى الاستحقاق أو عدم الاستحقاق للتخفيض من الدفع لصفّ السيّارة في المدينة.

مثال: الرقم 122 يشير إلى الاستحقاق للتخفيض في الدفع للأرنونا وعدم الاستحقاق للتخفيض في الدفع لرياض الأطفال وفي الدفع لصفّ السيّارة.

اكتبوا عمليّة خارجيّة باسم discounts بلغة Java أو Discounts بلغة C# ، تتلقّى ثلاثة متغيّرات: age, num, city .
age هو عدد صحيح يشير إلى عُمْر الساكن في المدينة، وَ num هو عدد صحيح يشير إلى عدد الأولاد للساكن،
وَ city هو قيمة بوليانيّة تشير إذا كان مكان عمل الساكن هو في المدينة (true) أو ليس في المدينة (false).
تُعيد العمليّة عددًا صحيحًا مكوَّنًا من ثلاثة أرقام يشير إلى استحقاق الساكن للتخفيضات أعلاه.

/ يتبع في صفحة 5 /

الفصل الثاني (60 درجة)

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

4. معطاة الفئة **Results** ، التي تمثّل أزمنة سباحة 50 مترًا لسبّاح في فترة معيّنة .
للفئة صفتان :

- arr – مصفوفة من نمط صحيح، تحوي عدد الثواني التي استغرقتها سباحة 50 مترًا للسبّاح في كلّ يوم في فترة معيّنة .
- name – اسم السبّاح، من نمط نصّ .

افترِضوا أنّه توجد عمليّتا get/Get وَ set/Set بالنسبة لصفتَي الفئة .

زمن سباحة 50 مترًا للسبّاح يمكن أن يتغيّر من يوم إلى آخر . "تَراجُع" هو اليوم الذي استغرقت فيه السباحة عددًا أكبر من الثواني ممّا في اليوم الذي قبله، وَ "تحسُّن" هو اليوم الذي استغرقت فيه السباحة عددًا أقلّ من الثواني ممّا في اليوم الذي قبله .
"فرق" هو الفرق في عدد الثواني بالقيمة المطلقة بين يوم معيّن واليوم الذي قبله .

أ. اكتبوا عمليّة داخليّة باسم deltas بلغة Java أو Deltas بلغة #C ، تتلقّى پارامترًا – isLonger من نمط بولينيّ، وتُعيد عددًا صحيحًا حسب الوصف الذي أمامكم :

- إذا كان الپارامتر isLonger الذي تمّ تلقّيه هو true ، تُعيد العمليّة المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تَراجُعًا" .
- إذا كان الپارامتر isLonger الذي تمّ تلقّيه هو false ، تُعيد العمليّة المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تحسُّنًا" .

مثال : بالنسبة للمصفوفة arr التي أمامكم، إذا كان isLonger هو true تُعيد العمليّة 14 ، وإذا كان isLonger هو false تُعيد العمليّة 17 .

	0	1	2	3	4	5	6
arr	**45**	**52**	**50**	**53**	**38**	**38**	**42**

شرح : الخلايا في المؤشّرات 6 وَ 3 وَ 1 تمثّل أيّامًا تُعتبَر "تَراجُعًا" ، والمجموع الكلّيّ للفروق فيها (4 + 3 + 7) هو 14 ثانية .
الخليّتان في المؤشّرين 4 وَ 2 تمثّلان يومين يُعتبَران "تحسُّنًا" ، والمجموع الكلّيّ للفرقين فيهما (15 + 2) هو 17 ثانية .

ب. اكتبوا عمليّة داخليّة باسم improving بلغة Java أو Improving بلغة #C ، تُعيد true إذا كان المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تحسُّنًا" **أكبر** من المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تَراجُعًا" .
خلاف ذلك تُعيد العمليّة false .
مثلًا، في المثال أعلاه تُعيد العمليّة true ، لأنّ المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تحسُّنًا" أكبر من المجموع الكلّيّ للفروق في الأيّام التي تُعتبَر "تَراجُعًا" (14 > 17) .
يجب الاستعانة بالعمليّة التي كتبتموها في البند "أ" .

/ يتبع في صفحة 6 /

5. الفئة **MyTime** تمثّل زمنًا، ولها صفتان:

- myMinute – عدد صحيح موجب بين 0 وَ 59 (بما في ذلك 0 وَ 59)، يمثّل دقيقة.
- myHour – عدد صحيح موجب بين 7 وَ 23 (بما في ذلك 7 وَ 23)، يمثّل ساعة.

افترِضوا أنّه توجد عمليّتا get/Get وَ set/Set بالنسبة لصفتَي الفئة، وعمليّة بنائيّة تتلقّى قِيَمًا بالنسبة لصفتَي الفئة.

أمامكم واجهة تطبيق الفئة. يمكن استعمال عمليّتَي الفئة بدون تطبيقهما.

عنوان العمليّة	وصف العمليّة
Java – public boolean before (MyTime other) **C#** – public bool Before (MyTime other)	تُعيد العمليّة true إذا كان زمن الكائن الحاليّ يسبق زمن الكائن other، خلاف ذلك (أي أنّه إذا كان الزمن مساويًا له أو بعده) تُعيد العمليّة false.
Java – public int diff (MyTime other) **C#** – public int Diff (MyTime other)	تُعيد العمليّة الفرق بالدقائق بالقيمة المطلقة بين زمن الكائن الحاليّ وزمن الكائن other.

الفئة **Parking** تعرض معلومات عن سيّارة رَكَنَت في موقف السيّارات، ولها ثلاث صفات:

- id – رقم لوحة ترخيص السيّارة، من نمط نصّ.
- in – زمن دخول السيّارة إلى موقف السيّارات، من نمط MyTime.
- out – زمن خروج السيّارة من موقف السيّارات، من نمط MyTime.

افترِضوا أنّه توجد عمليّتا get/Get وَ set/Set بالنسبة لصفات الفئة، وأنّ صفات الفئة ليست null.

معطاة مصفوفة أحاديّة الأبعاد cars من نمط **Parking** تحوي معلومات عن جميع السيّارات التي رَكَنَت في موقف السيّارات في يوم معيّن (موقف السيّارات يُفتَح في الساعة 07:00 ويُغلَق في الساعة 23:59). المصفوفة ليست مرتَّبة ولا تحوي قِيَم null.

أ. اكتبوا عمليّة خارجيّة باسم first بلغة Java أو First بلغة C#، تتلقّى المصفوفة cars، وتطبع رقم لوحة ترخيص السيّارة الأولى التي دخلت إلى موقف السيّارات.
افترِضوا أنّه لا توجد سيّارتان دخلتا إلى الموقف في نفس الوقت بالضبط.

يُحدَّد مبلغ الدفع في موقف السيّارات حسب فترة وقوف السيّارة. في الساعتين الأوليين (120 دقيقة) وقوف السيّارة هو مجّانيّ، وابتداءً من الدقيقة 121، تكون التسعيرة شيكلًا واحدًا لكلّ دقيقة وقوف.

ب. (1) اكتبوا عمليّة داخليّة في الفئة Parking باسم total بلغة Java أو Total بلغة C#، تُعيد المجموع الكلّيّ لعدد الدقائق التي رَكَنَت فيها السيّارة في موقف السيّارات.

(2) اكتبوا عمليّة خارجيّة باسم sumMoney بلغة Java أو SumMoney بلغة C#، تتلقّى المصفوفة cars، وتُعيد المبلغ الكلّيّ الذي دفعه جميع أصحاب السيّارات.
يمكن استعمال العمليّة التي كتبتموها في البند "ب (1)".

/ يتبع في صفحة 7 /

6. بهدف تشفير رسالة ما، يخلطون ترتيب الرموز التي تُركِّبها (الرموز ذاتها تُحفَظ لكنّ مكانها يختلف) .
مثال: بعد خلط ترتيب رموز الكلمة World يمكن الحصول على الرسالة المشفَّرة rdoWl .

كي يتمكّن مُستقبِل الرسالة المشفَّرة من فكّ تشفير رسالة المُرسِل الأصليّة، تُرفَق بالرسالة المشفَّرة "مصفوفة فكّ تشفير" .
كِبَر مصفوفة فكّ التشفير هو كَكِبَر الرسالة المشفَّرة، وفي كلّ خليّة يظهر رقم المؤشّر الأصليّ لكلّ رمز.
مثال: بالنسبة للرسالة المشفَّرة rdoWl (للرسالة الأصليّة World)، مصفوفة فكّ التشفير هي :

0	1	2	3	4
2	**4**	**1**	**0**	**3**

شرح: الرمز الأوّل في الرسالة المشفَّرة (r) يظهر في الرسالة الأصليّة في المؤشّر 2، والرمز الثاني في الرسالة المشفَّرة (d) يظهر في الرسالة الأصليّة في المؤشّر 4، والرمز الثالث في الرسالة المشفَّرة (o) يظهر في الرسالة الأصليّة في المؤشّر 1، وهكذا.

أ. اكتبوا مصفوفة فكّ التشفير بالنسبة للرسالة Study ، التي شُفِّرَت إلى الرسالة dyutS .

ب. اكتبوا عمليّة خارجيّة باسم originalText بلغة Java أو OriginalText بلغة C# ، تتلقّى نصًّا مشفَّرًا – str ومصفوفة فكّ تشفير – arr . تُعيد العمليّة النّصّ الأصليّ.

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.