GLOBAL ROBOTICS CHALLENGE

دليل القواعد الفنية للمسابقة 2026

التحدي العالمي للروبوتات

GLOBAL ROBOTICS CHALLENGE – GRC

تحدى المبرمجين بالبايثون

PYTHON DEVELOPMENT

Smart Programmers CHALLENGE

لمزيد من المعلومات :

Scan QR Code

COPYRIGHT All Rights Reserved

GLOBAL ROBOTICS CHALLENGE

1. المقدمة الفنية :

تهدف مسابقة البرمجة بلغة Python إلى صقل مهارات المشاركين في التفكير المنطقي وحل المشكلات البرمجية، من خلال مجموعة من المهام (Missions) المتدرجة التي تراعي الفئات العمرية المختلفة.

تعتمد المسابقة على تدريب الطلاب على تحويل الأفكار إلى حلول عملية باستخدام البرمجة، مع التركيز على:

- التحليل المنطقي للمشكلات.
- تصميم الخوارزميات البسيطة والمتقدمة.
- تنفيذ الكود بوضوح وتنظيم يسهّل قراءته وتطويره.

وتتميز المسابقة بأنها لا تختبر فقط الجانب المعرفي، بل تعزز أيضًا الإبداع، الدقة، والانضباط في كتابة الكود، بما يجعلها تجربة تعليمية وتنافسية متكاملة تناسب المبتدئين والمتقدمين على حد سواء.

2. تكوين الفريق :

- عدد أعضاء الفريق: من 2 إلى 4 متسابقين، تحت إشراف مدرب.
- الفئة العمرية:

1. فئة الناشئين(Junior) :
   - ❖ الأعمار: من 10 إلى 12 سنة.
2. فئة الكبار(Senior) :
   - ❖ الأعمار: من 13 إلى 17 سنة.
3. فئة البالغين (المتقدمين - Adult) :
   - ❖ الأعمار: 18 سنة فما فوق.

3. القواعد العامة:

- على كل مشارك إحضار جهاز اللابتوب الخاص به.
- يجب تسليم الكود/المهمة قبل انتهاء الوقت المحدد (العد التنازلي).
- يمنع استخدام الإنترنت تمامًا خلال وقت المسابقة.
- بعد كل جولة، يقوم الفريق بعرض عمله أمام لجنة التحكيم.
- لجنة التحكيم تقوم بمراجعة الكود المقدم يدويًا وتقييمه وفق معايير محددة.
- قبل بداية كل جولة، سيتم تخصيص 15 دقيقة لشرح التحدي والإجابة على أسئلة الفرق.
- أي تواصل مع أشخاص من خارج الفريق أثناء وقت المسابقة ممنوع تمامًا.
- أي تدخل أو مساعدة من المدربين/المشرفين أثناء وقت الجولات يؤدي إلى إنذار أولًا، وتكراره قد يؤدي لاستبعاد الفريق من المسابقة.

4. القواعد الخاصة بمبرمجي البايثون :

- الفئة العمرية المسموح لها بالمشاركة: من 10 إلى +18 سنة.
- يسمح باستخدام أي محرر أكواد (IDE) مثل: VS Code
- المسابقة تتكون من 3 جولات، وكل جولة عبارة عن تحدٍ مختلف يشرحه المشرف خلال الـ 15 دقيقة المخصصة قبل البداية.
- مدة كل جولة: ساعة واحدة فقط.
- التحديات مقسمة حسب الفئة العمرية:
- كل مهمة (Mission) يجب تنفيذها في الوقت المحدد.
- الكود يجب أن يكون:
  - ❖ منظم وواضح (Clean Code).
  - ❖ مُعلق عليه (Comments) توضح الفكرة الأساسية.
  - ❖ مبني بشكل صحيح (Structured)باستخدام (Functions/Classes).
  - ❖ مقروء (Readable) بأسماء متغيرات ودوال واضحة.
- أي ناتج Output يجب أن يكون منسقًا واحترافيًا.
- إستخدام الإنترنت خلال الجولة غير مسموح منعًا للغش.

5. المتطلبات الفنية:

- **مستوى الناشئين:**
  - ❖ **الإلمام بالأساسيات: المتغيرات، العمليات الحسابية، الشروط، الحلقات.**
  - ❖ **القدرة على استخدام القوائم والنصوص.**
  - ❖ **كتابة دوال بسيطة تأخذ مدخلات وتُرجع مخرجات.**
  - ❖ **معرفة بالقواميس والمجموعات.**
  - ❖ **فهم خوارزميات البحث والفرز البسيطة.**
  - ❖ **يُسمح باستخدام مجموعة محدودة من الـ Built-in Functions مثل:**
    - **.range(), upper(), lower(), int(), str()**
- **مستوى الكبار:**
  - ❖ **الإلمام بكل ما سبق في مستوى الناشئين.**
  - ❖ **التعامل مع الملفات (فتح – قراءة – كتابة).**
  - ❖ **تطبيق مبادئ البرمجة الكائنية OOP (الفئات، الكائنات، الوراثة البسيطة).**
  - ❖ **كتابة دوال متقدمة (Recursive, Lambda).**
  - ❖ **الإلمام بهياكل بيانات إضافية: Stacks, Queues, Tuples.**
  - ❖ **معرفة بخوارزميات بحث (Binary Search) وفرز (Selection sort , insertion sort).**
  - ❖ **يُسمح باستخدام دوال إضافية مثل:**
    - **.max(), min(), sum(), map(), filter()**
- **مستوى البالغين:**
  - ❖ **الإلمام بكل متطلبات مستوى الكبار.**
  - ❖ **التوسع في OOP (بما في ذلك الوراثة المتعددة).**
  - ❖ **التعامل مع الأخطاء (Exception Handling).**
  - ❖ **إنشاء واستخدام Modules وPackages.**
  - ❖ **معرفة بخوارزميات بحث (DFS, BFS) وفرز متقدمة (Merge, Sort, Quick Sort).**
  - ❖ **معرفة هياكل بيانات متقدمة: Heap, Linked List, Graph.**
  - ❖ **يُسمح باستخدام دوال إضافية مثل:**
    - **.any(), all(), round(), reversed(), type(), isinstance(), hasattr()**

6. أمثلة استرشادية (Examples) :

- **مستوى الناشئين:**
- **المهمة 1 :String Reverser**

اكتب برنامج يطلب من المستخدم إدخال كلمة، ثم يقوم بعكسها ويطبع النتيجة.

الناتج المتوقع:(Expected Output)

Enter a word: hello
olleh

- **المهمة 2 :Even Numbers Finder**

اكتب برنامج يطلب من المستخدم إدخال رقم، ثم يطبع جميع الأعداد الزوجية من 1 إلى هذا الرقم.

الناتج المتوقع:(Expected Output)

Enter a number: 10
2 4 6 8 10

- **مستوى الكبار:**
- **المهمة 1 :Word Counter**

اكتب برنامج يقرأ محتوى ملف نصي (text file) ثم يعرض عدد الكلمات الموجودة فيه.

الناتج المتوقع:(Expected Output)

File Content: "Python is great and Python is powerful”
Word Count = 7

- **المهمة 2 :Queue Simulation**

اكتب برنامج يحاكي قائمة انتظار (Queue). أضف عدة عناصر(enqueue) ، ثم أزل عنصرين(dequeue) ، واعرض القائمة بعد كل عملية.

الناتج المتوقع:(Expected Output)

Enqueue: A, B, C, D
Queue = [A, B, C, D]

Dequeue → A
Queue = [B, C, D]

Dequeue → B
Queue = [C, D]

- مستوى البالغين:
- المهمة: **Coin Change Problem (Dynamic Programming)**

اكتب برنامج يحسب أقل عدد من العملات اللازمة لتكوين مبلغ محدد باستخدام فئات معينة من العملات.

الناتج المتوقع:(Expected Output)

Coins : [5 ,2 ,1]
Amount: 11
Minimum coins needed = 3

7. للمزيد من الأمثله استرشادية :

https://drive.google.com/drive/folders/1YTdiyGHJS-enQ_c9UAwV2sSYQZn7DGQX?usp=drive_link

GLOBAL ROBOTICS CHALLENGE

Good Luck.