  
  
  
  
  
  
**Part01**

**Question: What is the default value assigned to array elements in C#?**

القيمة بتتحدد حسب نوع البيانات.

* لو أرقام، بتكون صفر.
* لو نصوص أو كائنات، بتكون null.
* لو Boolean، بتكون false.

**Question: What is the difference between Array.Clone() and Array.Copy()?**

**Clone():**

بيعمل نسخة جديدة من المصفوفة كلها، بنفس القيم.

**Copy():**

بينقل القيم من مصفوفة للتانية، وبتكون أنت اللي مجهز المصفوفة اللي هينقل لها.

**Question: What is the difference between GetLength() and Length for multi dimensional arrays?**

**Length**

بيرجع إجمالي عدد العناصر في المصفوفة كلها، يعني بيحسب كل الصفوف × كل الأعمدة.

**GetLength(dimension)**

بيرجع عدد العناصر في بُعد معين من أبعاد المصفوفة، زي عدد الصفوف أو عدد الأعمدة.

**Question: What is the difference between Array.Copy() and**

**Array.ConstrainedCopy()?**

* Array.Copy بتنقل البيانات من مصفوفة للتانية بشكل عادي، ولو حصل خطأ ممكن تكون نقلت جزء من البيانات.
* Array.ConstrainedCopy بتكون أكثر أمان، يا إما تنقل كل البيانات بنجاح، يا ما تنقلش خالص لو في مشكلة.

**Question: Why is foreach preferred for read-only operations on arrays?**

لأن:

* Foreach مصممة علشان تعدي على كل عنصر في المصفوفة من غير ما تغيّره.
* سهلة في الاستخدام
* بتحميك من إنك تعدل على العناصر بالغلط
* بتخلي الكود أوضح وأسهل في الفهم

**Question: Why is input validation important when working with user inputs?**

لأن:

* يسبب أخطاء في البرنامج
* يفتح ثغرات أمان زي (SQL Injection)
* يخلي النتائج غلط أو غير منطقية

التحقق من الإدخال بيضمن إن البيانات اللي داخلة صح وآمنة، وبيخلّي البرنامج:

* أكثر استقرارًا
* أكثر أمانًا
* بيشتغل بشكل متوقع

**Question: How can you format the output of a 2D array for better readability?**

لما:

* تتعرض العناصر على شكل صفوف وأعمدة
* يعني كل صف في سطر جديد، وكل عنصر متفصل بمسافة أو علامة.
* تستخدم مسافات متساوية أو علامات زي Tab أو (|)
* علشان العناصر تكون مصطفّة جنب بعض.
* تضيف عنوان أو تنسيق بسيط لو بتحب تنظم البيانات أكتر.

**Question: When should you prefer a switch statement over if-else?**

**تستخدم switch لما:**

* عندك متغير واحد وبتقارن قيمته مع حالات محددة زي أرقام أو حروف أو كلمات.
* الحالات كتير ومرتبطة بقيم ثابتة، مش شروط معقدة.
* عايز تخلي الكود أوضح وأسهل في القراءة من كذا if-else ورا بعض.

**Question: What is the time complexity of Array.Sort()?**

الوقت اللي بتاخده Array.Sort غالبًا بيكون:

**n log n**  لما البيانات عادية.  
وفي حالات نادرة جدًا ممكن تبقى أبطأ.

**Question: Which loop (for or foreach) is more efficient for calculating the sum of an array, and why?**

الـ for loop بتكون غالبًا أكتر كفاءة شوية من foreach، خصوصًا مع المصفوفات.

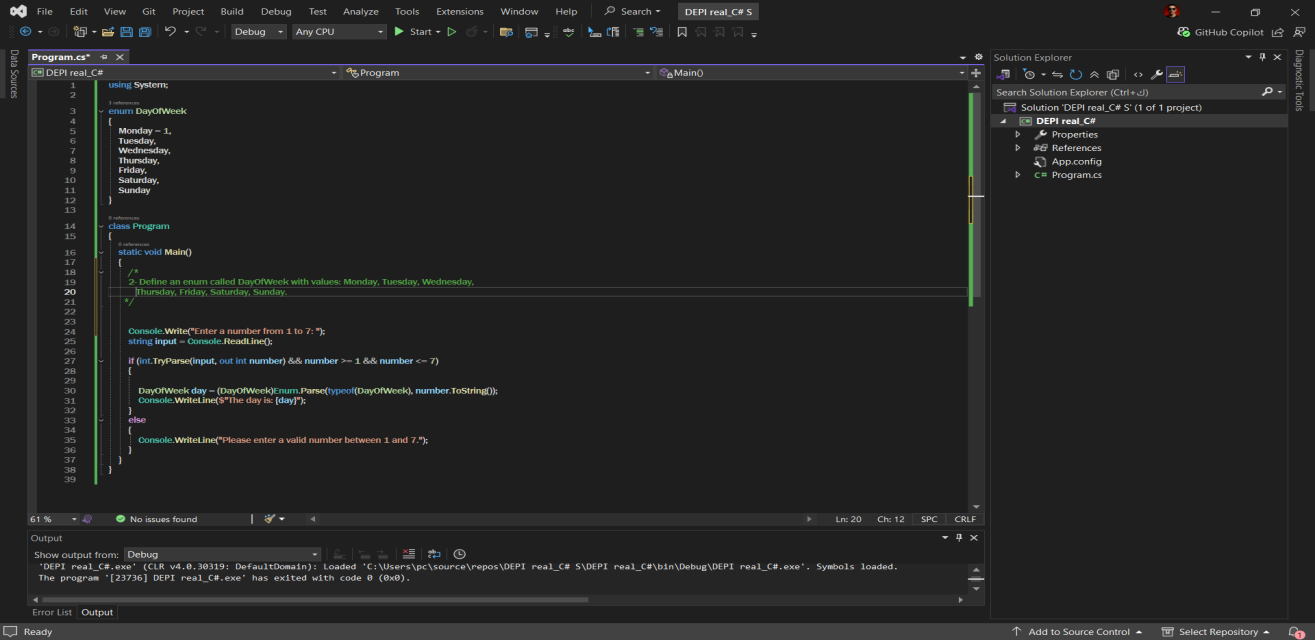
لأن:

For بتتعامل مع العناصر مباشرة باستخدام الفهرس (index).

foreach بيستخدم Enumerator داخلي، وده بيضيف خطوة زيادة في الخلفية.

**Part02**

**2- Define an enum called DayOfWeek with values: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.**



**3- What happens if the user enters a value outside the range of 1 to 7?**

البرنامج هيطبع رسالة تقوله إن الرقم غلط، ومش هيكمّل.

يعني مش هيحوّل الرقم، ومش هيحصل خطأ في الكود.

**Part03 Bonus**

**4- self study report**

**إيه هو الـ Jagged Array؟**

هو مصفوفة جواها مصفوفات، وكل مصفوفة جواها ممكن يكون ليها طول مختلف.

يعني مش زي المصفوفة الثنائية (2D array) اللي كل الصفوف فيها نفس الطول.

مثال بسيط:

صف أول فيه 3 عناصر، صف تاني فيه 5 عناصر، وصف ثالث فيه 2 بس.

**إيه هو "Array of Objects"؟**

يعني مصفوفة كل عنصر فيها كائن (object) من كلاس معين، مش مجرد رقم أو نص.

مثال:

لو عندك كلاس اسمه "طالب"، تقدر تعمل مصفوفة فيها طلاب.

كل عنصر في المصفوفة بيكون طالب مختلف، ليه بياناته الخاصة )زي الاسم، السن، الدرجات... إلخ(.

**5- what’s the default size of stack and heap and what are the consideration**

**Stack:**

* حوالي 1 ميجا لكل Thread
* للمتغيرات البسيطة
* سريع بس مساحته محدودة

**Heap:**

* مفيش حجم ثابت
* للكائنات والبيانات الكبيرة
* أبطأ شوية، بس مساحته أكبر

**6- what is time complexity**

* هي طريقة نعرف بيها الكود سريع ولا بطيء لما البيانات تكتر.
* لو الكود بياخد وقت ثابت مهما كان حجم البيانات، ده سريع.
* لو الوقت بيزيد كل ما البيانات زادت، ده أبطأ.