أقيم تعلمي

أُوظِّف في هذا الدرس ما تعلَّمْتُه من معارف في الإجابة عن الأسئل :المعرفة :ة الآتية

السؤال الأوَّل: ما أبرز أوجه التشابه وأوجه الاختلاف بين القوائم والسلاسل؟

أوجه التشابه بين القوائم والسلاسل	
	كلاهما يمثل تسلسلًا من العناصر بترتيب محدد أي أن كل عنصر له مؤشر (index) يحدد موقعه في التسلسل. • يمكن الوصول إلى عناصر القوائم والسلاسل باستخدام مؤشراتها. • يمكن تحديد طول القائمة والسلسلة باستخدام الدالة( ) len. • يمكن تكرار عناصر القائمة والسلسلة باستخدام الحلقة for. يمكن التحقق من وجود عنصر معين في القائمة والسلسلة باستخدام العامل in.

أوجه الإختلاف بين القوائم والسلاسل	القوائم	السلاسل
	يمكن أن تحتوي على أي نوع من البيانات، سواء كانت أعدادًا، سلاسل، قوائم أخرى، أو حتى دالات.	تحتوي فقط على سلاسل من الأحرف
	• قابلة للتغيير، أي يمكن إضافة عناصر جديدة، حذف عناصر، أو تعديل	غير قابلة للتغيير، أي لا يمكن تعديل عناصرها بعد إنشائها. يجب إنشاء سلسلة جديدة إذا أردت إجراء أي تغيير.

تدعم عمليات خاصة بالسلاسل مثل : تحويل الأحرف إلى حروف كبيرة أو صغيرة، البحث عن سلاسل فرعية، وغيرها	عناصر موجودة. تدعم مجموعة واسعة من العمليات مثل الفرز، العكس، الدمج، وغيرها.	

أكتب جملة يُمكِن استخدامها في حذف الحرف الأخير من سلس **السؤال الثاني:**
لة تُسمّى str1()

```
str1=" hello"
str1 =strl[:-1]
print(str1)
```

لتنفيذ كل مهمة من المهام ا)أو أكثر(أذكر طريقتين مختلفتين **السؤال الثالث:**
لآتية، ثمَّ أُحدِّد أيُّ
هذه الطرائق أسهل أو أفضل

1. عناصر 10 طباعة العنصر الأوَّل في قائمة تتألَّف من.

- باستخدام الدالة (index)

```
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(my_list[0])
```

- باستخدام دالة pop()

```
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
first_element = my_list.pop(0)
print(first_element)
```

- الطريقة الاولى هي الاسهل والافضل

طباعة العنصر الأخير في قائمة تتألَّف من2 . 10 عناصر.

- الطريقة الاولى : باستخدام (index) للوصول إلى العنصر الأخير، الفهرسة تبدأ من الصفر، لذلك للوصول إلى العنصر الأخير في قائمة من 10 عناصر، سنستخدم الفهرس 9 (10 - 1).

- الطريقة الثانية: باستخدام دالة len() تعطي طول القائمة (عدد العناصر). وبالتالي، يمكننا استخدام len(my_list) - 1 للحصول على فهرس العنصر الأخير.

```
y_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
last_index = len(my_list) - 1
last_element = my_list[last_index]
print(last_element())
```

التأكد من وجود حرف 'a' في سلسلة ما. 3 .

1.

```
def check_for_a(text):
```

```
 if 'a' in text:
   print("The text contains the letter 'a'")
 else:
   print("The text does not contain the letter 'a'")
```

2.

```
my_string = " The text contains the letter 'a'"
if any(char == 'a' for char in my_string):
   print("The text does not contain the letter 'a'")
```

الطريقة الثانية الاسهل

4 . ترتيب عناصر قائمة ما(الجاهزةِ الدوالِّ باستخدام).

1

```
my_list = [5, 2, 8, 1, 9]
my_list.sort()
print(my_list)
```

2

```
sorted_list = sorted(my_list, reverse=True)
print(sorted_list)
```

الطريقة الاولى أكثر استخدام

5 . إيجاد العنصر الأكبر في قائمة ما.

1

```
my_list = [3, 7, 2, 9, 5]
largest_number = max(my_list)
print("أكبر عدد هو:", largest_number)
```

2

```
my_list = [3, 7, 2, 9, 5]
```

```
sorted_list = sorted(my_list, reverse=True)
largest_number = sorted_list[0]
print("أكبر عدد هو:", largest_number)
```

3

```
my_list = [3, 7, 2, 9, 5]
largest_number = my_list[0]  # نفترض أن العنصر الأول هو الأكبر في البداية
for num in my_list:
    if num > largest_number:
        largest_number = num
print("أكبر عدد هو:", largest_number)
```

الطريقة الاولى هي الطريقة الاسهل

6 . حذف جميع عناصر قائمة ما.

1

```
my_list = [1, 2, 3, 4, 5]
my_list = []
print(my_list)
```

2

```
my_list = [1, 2, 3, 4, 5]
my_list[:] = []
print(my_list)
```

3

```
numbers = [10, 20, 30, 40, 50]
numbers.clear()
print(numbers)
```

4

```
my_list = [1, 2, 3, 4, 5]
for i in range(len(my_list)):
    del my_list[0]
print(my_list)
```

أُوظِّف مهارة التفكير الناقد ومهارة حَلّ المشكلات والمهارات ال :المهارات برمجية في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أتتبَّع كل جزء من البرنام(أتتبَّع البرنامج الآتي من دون تشغيله ، ثمَّ أذكر ناتج تشغيله)ج بصورة مُنفصِلة.

- الجزء الأول والثاني: سيطبعان أكبر عنصر في كل جزء متتالي من القائمة، مع اختلاف طفيف في نقطة البداية.
- الجزء الثالث: ستصبح القائمة الأصلية ضعف طولها وتحتوي على نفس العناصر مرتين بترتيب عكسي
- يعكس هذا الجزء القائمة ثم يضيف كل عنصر من القائمة المقلوبة إلى نهاية القائمة الأصلية في النهاية، تصبح القائمة الأصلية ضعف طولها وتحتوي على نفس العناصر مرتين، ولكن بترتيب عكسي.

السؤال الثاني: أُحدِّد الهدف الرئيس لكل برنامج من البرامج المُبيَّنة في الجدول الآتي، ولا أَصِف ما يقوم به البرنامج في كل سطر بصورة مُنفصِلة، وإنَّما أس تعمل بضع كلمات لتلخيص المهام التي يؤدّيها البرنامج بوجه عام.

السؤال الثالث: أكتب برنامجًا يُحقِّق كلًّا من المهام الآتية

```
result = 
for e in a
if e < 0
result = [e] + resu
else
result += [e
print(result
```

- هذا البرنامج يقوم بفرز العناصر السالبة في قائمة معينة بحيث تكون في بداية القائمة وبالترتيب العكسي لظهورها الأصلي، مع الحفاظ على ترتيب العناصر الموجبة والصفر.

```
found = False
for x in a:
for y in a:
if x + y == 0:
found = True
break
```

هذا البرنامج يفحص جميع الأزواج الممكنة ويحدد ما إذا كان a من الأعداد في القائمة هناك زوج مجموع عناصره يساوي صفرًا.

```
for row in a
temp = row[0
row[0] = row[-1
row[-1] = tem
```

- هذا البرنامج يقوم بتبديل العنصر الأول والأخير في كل صف داخل قائمة من القوائم.

```
j = len(a) - 1
i = 0
for row in a:
temp = row[i]
row[i] = row[j]
row[j] = temp
i += 1
j -= 1
```

- هذا المقطع البرمجي يقوم بعملية عكس عناصر كل صف بشكل كامل، وليس فقط تبادل العنصر الأول والأخ بسبب زيادة في كل تكرار j ونقصان قيمة i قيمة

بناء قائمة من أرقام يدخلها المستخدم ثم طباعة عدد الأرقام ال زوجية الموجودة : **المهمة الأولى**

للحلقة، مما يؤدي إلى تبادل جميع العناصر تباعًا.

في القائمة

```
umbers = []
n = int(input("Enter the number of elements: "))

for i in range(n):
  num = int(input("Enter number: "))
  numbers.append(num)

even_count = 0
for num in numbers:
  if num % 2 == 0:
    even_count += 1

print("Number of even numbers:", even_count)
```

المهمة الثانية : بناء قائمتين من أرقام يدخلها المستخدم ثم التحقق من وجود أ ي عنصر من عناصر القائمة الأولى في القائمة الثانية.

```
list1 = []
n1 = int(input("Enter the number of elements for the first list: "))
for i in range(n1):
  num = int(input("Enter number for the first list: "))
  list1.append(num)
```

```
list2 = []
n2 = int(input("Enter the number of elements for the second list: "))
for i in range(n2):
  num = int(input("Enter number for the second list: "))
  list2.append(num)

found = False
for element in list1:
  if element in list2:
    found = True
   break

if found:
  print("There is at least one element from the first list in the second list.")
else:
  print("There are no common elements between the two lists.")
```

المهمة الثالثة: قراءة سلسلة أحرف من المستخدم ثم التأكد من وجود 3 أحرف متتابعة ومتساوية داخل السلسلة.

```
text = input("Enter a string: ")

found = False
for i in range(len(text) - 2):
  if text[i] == text[i+1] == text[i+2]:
    found = True
```

```
    break

if found:
  print("The string contains 3 consecutive equal characters.")
else:
  print("The string does not contain 3 consecutive equal characters.")
```

السؤال الرابع: أكتب جزءاً من برنامج يتحقق من أن قائمة ثنائية الأبعاد اسمها board هي قائمة مربعة أي أن عدد الصفوف مساو لعدد الأعمدة

```
board = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
if is_square(board):
  print("The board is square.")
else:
  print("The board is not square.")
```