به نام خدای یکتا

گزارش تمرین اول درس پایگاه داده پیشرفته

نام و نام خانوادگی :

مبین رمضانی خرم آبادی 40311415016

نام استاد :

آقای دکتر آرمین رشنو

مهر ۱۴۰۳

سوال 1 :

1- Write a program to get a number and print all its **even digits** separated by *.
Input: 822145635
Output: 6*4*2*2*8

حل سوال 1 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```
1    def evenDigits(number):
2        even_digits = [digit for digit in str(number) if int(digit) % 2 == 0]
3        return '*'.join(even_digits)
4
5
6    num = 822145635
7    print(evenDigits(num))
```

در این برنامه، خط اول یک تابع به نام evenDigits تعریف می‌کند که یک پارامتر به نام number به عنوان ورودی می‌گیرد. در خط دوم، یک لیست تعریف شده است که از number به صورت رشته (str) استفاده می‌کند تا بتواند هر رقم را جداگانه بررسی کند. حلقه for درون لیست، هر رقم از number را به صورت یک رشته جدا می‌کند. شرط if درون حلقه بررسی می‌کند که آیا رقم مورد نظر زوج است یا خیر (یعنی باقیمانده تقسیم بر ۲ صفر باشد). نتیجه‌ی نهایی لیستی از ارقام زوج به نام even_digits خواهد بود. در خط سوم، با استفاده از متد join، لیست ارقام زوج را با کاراکتر * به یک رشته تبدیل می‌کند. ارقام زوج توسط * از هم جدا می‌شوند. در خط ششم، یک متغیر به نام num تعریف شده و مقدار آن برابر با عدد 822145635 است. درنهایت خط ۷ ام، تابع evenDigits را با ورودی num فراخوانی می‌کند و نتیجه آن را چاپ می‌کند.

خروجی آن به صورت زیر است :

```
Python Console
8*2*2*4*6
```

سوال 2 :

2- Write a program to compute the following expression for **500 sentences**:

$$\frac{3!}{2+9} - \frac{5!}{3+7} + \frac{7!}{4+5} - \frac{9!}{5+3} + \frac{11!}{6+1} - \frac{13!}{7-1} \dots$$

حل سوال 2 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```python
import math


def compute_expression(n):
    change_sign = True
    sign = 1
    fact_start = 3
    first_num = 2
    second_num = 9
    result = 0
    for _ in range(n):
        fact = math.factorial(fact_start)
        result += sign * (fact / (first_num + second_num))
        fact_start += 2
        first_num += 1
        second_num -= 2
        if change_sign:
            sign = -1
            change_sign = False
        else:
            sign = 1
            change_sign = True

    return result


print(compute_expression(500))
```

در این برنامه، خط اول کتابخانه‌ی math برای استفاده از تابع factorial (محاسبه فاکتوریل) در برنامه وارد می‌شود. در خط ۴، تابعی به نام compute_expression تعریف شده که یک پارامتر n دریافت می‌کند، این پارامتر تعداد دفعاتی است که حلقه در تابع اجرا خواهد شد. در خط ۵، change_sign متغیری برای کنترل تغییر علامت در هر مرحله از محاسبه هست که در ابتدا مقدار True دارد. در خط ۶، sign متغیری که در هر مرحله مشخص می‌کند که مقدار مثبت یک یا منفی یک به محاسبات اضافه شود و در ابتدا مقدار آن ۱ است. در خط ۷، fact_start شروع فاکتوریل از ۳ خواهد بود و به تدریج در هر مرحله افزایش می‌یابد. در خط ۸، first_num عدد اولی است که در مخرج کسر استفاده می‌شود و به تدریج اضافه می‌شود و در ابتدا مقدار آن ۲ میباشد. در خط ۹، second_num عدد دومی است که در مخرج کسر استفاده می‌شود و به تدریج کاهش میابد و در ابتدا مقدار آن ۹ میباشد. در خط ۱۰، result متغیری است که نتیجه نهایی محاسبات را ذخیره می‌کند.

از خط ۱۱ تا ۲۲، ساختار حلقه for تعریف شده است. در خط ۱۱، حلقه به تعداد n بار تکرار می‌شود. در خط ۱۲، فاکتوریل عدد fact_start محاسبه می‌شود و درون متغیر با نام fact ذخیره می‌شود. در خط ۱۳، محاسبه اصلی برنامه طبق الگویی که مشخص است، در هر بار اجرای حلقه انجام می‌شود و مقدار آن با متغیر result جمع می‌شود و درون این متغیر ذخیره می‌شود. در خط ۱۴، متغیر fact_start طبق الگو ۲ واحد افزایش می‌یابد و درون همان متغیر ذخیره می‌شود. در خط ۱۵، متغیر first_num طبق الگو ۱ واحد افزایش می‌یابد و درون همان متغیر ذخیره می‌شود. در خط ۱۶، متغیر second_num طبق الگو ۲ واحد کاهش می‌یابد و درون همان متغیر ذخیره می‌شود. در خط ۱۷، مقدار change_sign بررسی می‌شود. اگر True باشد، یعنی باید علامت به ۱- تغییر کند و این کار در دستور خط ۱۸ انجام می‌شود و بعد از آن در خط ۱۹، مقدار change_sign برابر با False می‌شود. اما اگر False باشد، یعنی باید علامت به ۱ تغییر کند و این کار در دستور خط ۲۱ انجام می‌شود و بعد از آن در خط ۲۲، مقدار change_sign برابر با True می‌شود. در خط ۲۴، نتیجه کلی محاسبه شده توسط تابع بازگردانده می‌شود. درنهایت در خط ۲۷، تابع compute_expression با مقدار ۵۰۰ برای n فراخوانی می‌شود و نتیجه محاسبات چاپ می‌شود.

نکته : خروجی این برنامه خطا می‌دهد. زیرا زمانی که به جمله ۱۲ ام میرسد، مقدار first_num و second_num به ترتیب برابر با ۱۳ و ۱۳- می‌شود و اگر این دو مقدار در مخرج کسر باهم جمع شوند، مقدار آن صفر خواهد شد که خطای تقسیم بر صفر رخ می‌دهد و ادامه اجرای برنامه با خطا مواجه می‌شود.

سوال 3 :

3- Write a program to **print all 4 digits numbers** (between 1000 and 9999) that the **sum** of the **first** and **second** digits is equal with the **product** of the **third** and **forth** digits.

3466: $6 + 6 = 3 \times 4$

Output: 1110, 1101, …, 2999 , … , 3466 , ….

حل سوال 3 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```
for num in range(1000, 10000):
    number = str(num)
    digit1 = int(number[0])
    digit2 = int(number[1])
    digit3 = int(number[2])
    digit4 = int(number[3])
    if digit1 + digit2 == digit3 * digit4:
        print(num)
```

در این برنامه، خط اول یک حلقه for تعریف شده که متغیر num در بازه‌ی اعداد ۴ رقمی از ۱۰۰۰ تا ۹۹۹۹ تکرار می‌شود. به عبارتی، این حلقه همه‌ی اعداد چهار رقمی را بررسی می‌کند. در خط دوم، عدد num که یک عدد صحیح است، به رشته (string) تبدیل می‌شود و درون متغیر number ذخیره می‌شود. این کار به ما این امکان را می‌دهد که هر رقم آن را به‌صورت جداگانه استخراج کنیم. در خط سوم، رقم اول عدد، یعنی number[0] که اولین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit1 ذخیره می‌شود. در خط چهارم، رقم دوم عدد، یعنی number[1] که دومین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit2 ذخیره می‌شود. در خط پنجم، رقم سوم عدد، یعنی number[2] که سومین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit3 ذخیره می‌شود. در خط ششم، رقم چهارم عدد، یعنی number[3] که چهارمین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit4 ذخیره می‌شود. در خط هفتم، شرط if بررسی می‌کند که آیا جمع دو رقم اول (digit1 + digit2) برابر با ضرب دو رقم آخر (digit3 * digit4) است یا خیر. اگر این شرط برقرار باشد، دستور درون شرط اجرا می‌شود که همان عدد چهار رقمی جاری در حلقه است را چاپ می کند.

خروجی آن به صورت زیر است :

```
1011
1112
1121
1213
1231
1314
1322
1341
```

و همینطور الی آخر ...

سوال 4 :

4- Write a program to print **all 3 digits numbers** (between a 100 and 999) that does **not** have **odd** digits. Consider 0 as even digit.
Output: 200, 202,204, 206,208,**220**,222,…

حل سوال 4 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```
1    for num in range(100, 1000):
2        number = str(num)
3        digit1 = int(number[0])
4        digit2 = int(number[1])
5        digit3 = int(number[2])
6        if digit1 % 2 == 0 and digit2 % 2 == 0 and digit3 % 2 == 0:
7            print(num)
```

در این برنامه، خط اول یک حلقه for ایجاد می‌کند که متغیر num را به ترتیب در بازه‌ی اعداد ۳ رقمی (از ۱۰۰ تا ۹۹۹) قرار می‌دهد. به عبارت دیگر، این حلقه تمام اعداد سه‌رقمی را بررسی می‌کند. در خط دوم، عدد num که یک عدد صحیح است به یک رشته (string) تبدیل می‌شود و مقدار آن درون number ذخیره می‌کند. این تبدیل به ما اجازه می‌دهد که هر یک از ارقام عدد را به صورت جداگانه و با استفاده از اندیس دسترسی پیدا کنیم. دقیقا مشابه با سوال قبل. در خط سوم، رقم اول عدد، یعنی number[0] که اولین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit1 ذخیره می‌شود. در خط چهارم، رقم دوم عدد، یعنی number[1] که دومین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit2 ذخیره می‌شود. در خط پنجم، رقم سوم عدد، یعنی number[2] که سومین کاراکتر رشته است، به عدد صحیح (integer) تبدیل و در متغیر digit3 ذخیره می‌شود. در خط ۶، این شرط بررسی می کند که آیا سه تا شرط زوج بودن برای ارقام آن به طور همزمان صدق می کند یا خیر. اگه شرط درست باشد، یعنی همه ارقام زوج باشند، در خط ۷، عدد num که همان عدد سه‌رقمی جاری است، چاپ می‌شود.

خروجی آن به صورت زیر است :

```
200
202
204
206
208
220
222
224
226
```

و همینطور الی آخر ...

سوال 5 :

5- Write a program to get **n** from user and print the following pattern:
Input: n=8
Output:

```
1
2   4
3   6   9
4   8   12   16
5   10   15   20   25
6   12   18   24   30   36
7   14   21   28   35   42   49
8   16   24   32   40   48   56   64
```

حل سوال 5 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```
1    def pattern(n):
2        for i in range(1, n + 1):
3            for j in range(1, i + 1):
4                print(i * j, end=" ")
5            print()
6
7
8    pattern(8)
```

در این برنامه، خط اول یک تابع به نام pattern تعریف می‌کند که یک پارامتر n می‌گیرد. این پارامتر تعیین می‌کند که الگو چند ردیف داشته باشد. خط دوم، یک حلقه for تعریف می‌کند که متغیر i را از 1 تا n+1 (شامل n) افزایش می‌دهد. در هر تکرار، مقدار i نشان‌دهنده‌ی ردیف جاری در الگو است. در خط ۳، یک حلقه داخلی for تعریف شده که متغیر j از 1 تا مقدار i+1 (شامل i) حرکت می‌کند. این حلقه برای هر ردیف به اندازه مقدار i (تعداد ستون‌ها) اجرا می‌شود. در خط ۴، مقدار ضرب i و j را محاسبه کرده و چاپ می‌کند. end=" " باعث می‌شود که بعد از هر چاپ، به جای رفتن به خط بعدی، یک فاصله (Space) درج شود تا اعداد در یک ردیف نمایش داده شوند. در خط ۵، بعد از اتمام هر ردیف از اعداد (پس از اتمام حلقه داخلی for)، این دستور باعث می‌شود که نشانگر به خط بعدی منتقل شود تا ردیف بعدی چاپ شود. در نهایت خط ۸، تابع pattern را با مقدار n = 8 فراخوانی می‌کند که به معنی چاپ یک الگوی ۸ ردیفی است.

خروجی آن به صورت زیر است :

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
```

سوال 6 :

6- Write a program to get **n** and then **n numbers** from user and compute **maximum**, **minimum**, **average** and **standard deviation**. Test your program for **5 different inputs**.

Input

```
Enter the n: 5

Enter number 1: 12
Enter number 2: 15
Enter number 3: 9
Enter number 4: 11
Enter number 5: 84
```

Output

```
Maximum is: 84
Minimum is: 9
Average is: 26.2
Standard Deviation is : 29.06
```

حل سوال 6 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```python
import numpy as np

numbers = []
n = int(input("Enter the n:"))
for i in range(n):
    number = int(input(f"Enter number {i+1}:"))
    numbers.append(number)

numbers = np.array(numbers)
maximum = np.max(numbers)
minimum = np.min(numbers)
average = np.mean(numbers)
std = np.std(numbers)  # standard deviation

print("Maximum is:", maximum)
print("Minimum is:", minimum)
print("Average is:", average)
print(f"Standard Deviation is: {std:.2f}")
```

در این برنامه، خط اول کتابخانه‌ی numpy وارد می‌شود و به‌طور اختصاری به نام np استفاده می‌شود. این کتابخانه برای انجام محاسبات عددی کارآمد بر روی آرایه‌ها و ماتریس‌ها استفاده می‌شود. در خط ۳، یک لیست خالی به نام numbers ایجاد می‌شود تا اعداد وارد شده توسط کاربر در آن ذخیره شوند. در خط ۴، از کاربر درخواست می‌شود که تعداد n (تعداد اعدادی که می‌خواهد) را وارد کند. ورودی به عدد صحیح تبدیل شده و در متغیر n ذخیره می‌شود. در خط ۵، یک حلقه for تعریف شده که از 0 تا n-1 تکرار می‌شود. این حلقه برای دریافت n عدد از کاربر استفاده می‌شود. در خط۶، در هر تکرار حلقه، از کاربر درخواست می‌شود که یک عدد وارد کند. عدد وارد شده به عدد صحیح تبدیل شده و در متغیر number ذخیره می‌شود و در خط ۷، این مقدار به لیست numbers اضافه خواهد شد.

در خط ۹، لیست numbers به یک آرایه‌ی numpy تبدیل می‌شود تا بتوانیم عملیات عددی مختلف را بر روی آن انجام دهیم. در خط ۱۰، از تابع np.max استفاده می‌شود تا بیشترین مقدار (بزرگترین عدد) در آرایه‌ی numbers پیدا شود و در متغیر maximum ذخیره شود. در خط ۱۱، از تابع np.min استفاده می‌شود تا کمترین مقدار (کوچکترین عدد) در آرایه‌ی numbers پیدا شود و در متغیر minimum ذخیره شود. در خط ۱۲، از تابع np.mean استفاده می‌شود تا میانگین اعداد موجود در آرایه محاسبه شود و در متغیر average ذخیره شود. در خط ۱۳، از تابع np.std استفاده می‌شود تا انحراف معیار اعداد موجود در آرایه محاسبه شود و در متغیر std ذخیره شود.

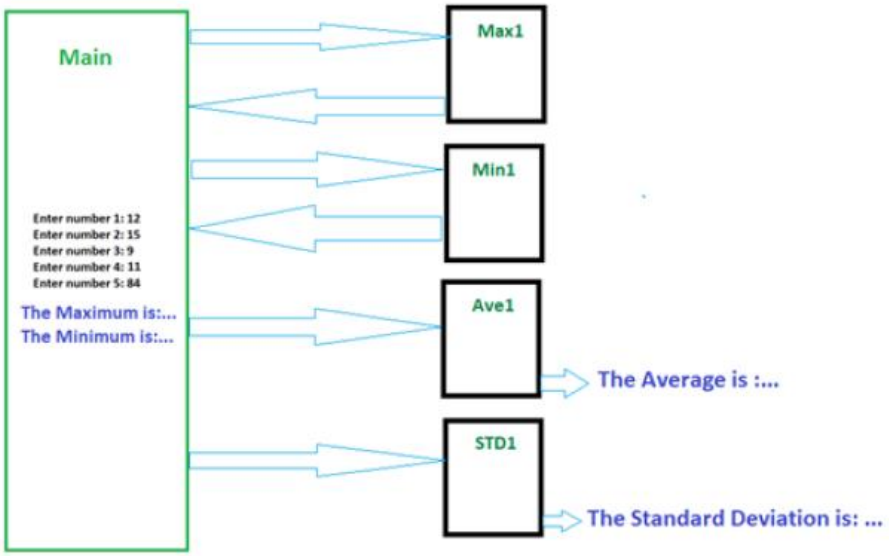
در خط ۱۵، مقدار بیشترین عدد (maximum) چاپ می‌شود. در خط ۱۶، مقدار کمترین عدد (minimum) چاپ می‌شود. در خط ۱۷، مقدار میانگین (average) چاپ می‌شود. در خط ۱۸، مقدار انحراف معیار (std) با دقت دو رقم اعشار چاپ می‌شود.

خروجی آن به صورت زیر است :

```
Enter the n:5
Enter number 1:12
Enter number 2:15
Enter number 3:9
Enter number 4:11
Enter number 5:84
Maximum is: 84
Minimum is: 9
Average is: 26.2
Standard Deviation is: 28.96
```

سوال 7 :

7- Write 4 functions for question 6. Get **5 numbers** from user in main. Send numbers for Max1, Min1, Ave1 and STD1 functions. Max1 gives n numbers and return maximum. Min1 gives n number and return minimum. Ave1 and STD1 give n number and print average and standard deviation in their own body and do not return any value. Test your program for 5 different inputs.



حل سوال 7 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```python
from statistics import mean
import math


def Max1(numbers):
    maximum = numbers[0]
    for num in numbers:
        if num > maximum:
            maximum = num
    print("The Maximum is: ", maximum)


def Min1(numbers):
    minimum = numbers[0]
    for num in numbers:
        if num < minimum:
            minimum = num
    print("The Minimum is: ", minimum)

```

```
20
21  def Ave1(numbers):
22      total = 0
23      for num in numbers:
24          total += num
25      average = total / len(numbers)
26      print("The Average is: ", average)
27
28
29  def STD1(numbers):
30      ave = mean(numbers)
31      total = 0
32      for num in numbers:
33          total += ((num - ave) ** 2)
34
35      std = math.sqrt(total / len(numbers))
36      print(f"The Standard Deviation is: {std:.2f}")
37
38
39  nums = []
40  n = int(input("Enter the n:"))
41  for i in range(n):
42      number = int(input(f"Enter number {i + 1}:"))
43      nums.append(number)
44
45  Max1(nums)
46  Min1(nums)
47  Ave1(nums)
48  STD1(nums)
```

در این برنامه، خط اول از ماژول statistics، تابع mean (میانگین) وارد می‌شود که برای محاسبه میانگین اعداد در تابع STD1 استفاده خواهد شد. خط دوم کتابخانه math وارد می‌شود که برای استفاده از تابع sqrt (محاسبه جذر) در محاسبه انحراف معیار استفاده خواهد شد. در خط ۵، تابعی به نام Max1 تعریف می‌شود که یک آرایه از اعداد (numbers) دریافت می‌کند و بیشترین مقدار را در آن پیدا می‌کند. در خط ۶، مقدار اولین عنصر از لیست numbers در متغیر maximum ذخیره می‌شود و این متغیر به عنوان مقدار اولیه‌ی ماکزیمم استفاده می‌شود. در خط ۷، یک حلقه for تعریف شده که هر عنصر از لیست numbers را بررسی می‌کند. در خط ۸، شرط if بررسی می‌کند آیا عدد فعلی (num) از مقدار فعلی maximum بزرگتر است یا نه. اگر این شرط برقرار باشد، در خط ۹، عدد بزرگتر به عنوان ماکزیمم جدید در نظر گرفته می‌شود. در خط ۱۰، بعد از اتمام حلقه، مقدار نهایی maximum که بیشترین عدد است چاپ می‌شود.

در خطوط ۱۳ تا ۱۸، تابع Min1 مشابه تابع Max1 است، ولی کمترین عدد را پیدا می‌کند.در خط ۲۱، این تابع میانگین اعداد را محاسبه می‌کند. در خط ۲۲، متغیر total برای جمع کردن اعداد تنظیم شده و مقدار اولیه آن 0 است. در خط ۲۳ و ۲۴، در هر تکرار حلقه، عدد فعلی به متغیر total اضافه می‌شود. بعد از اتمام حلقه، در خط ۲۵، مقدار میانگین با تقسیم مجموع کل اعداد بر تعداد اعداد محاسبه می‌شود. در خط ۲۶ هم مقدار نهایی میانگین چاپ می‌شود.

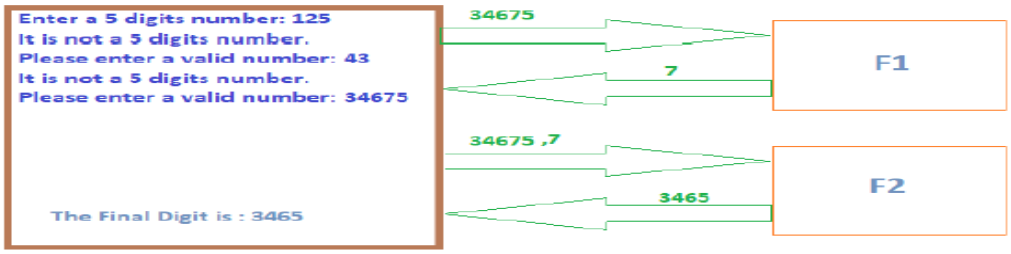
خط ۲۹، تابع انحراف معیار اعداد را محاسبه می‌کند. در خط ۳۰، از تابع mean استفاده می‌شود تا میانگین اعداد محاسبه و در متغیر ave ذخیره شود. در خط های ۳۲ و ۳۳، در هر تکرار حلقه، مربع تفاوت هر عدد با میانگین محاسبه و به متغیر total اضافه می‌شود. در خط ۳۵، مقدار انحراف معیار با استفاده از فرمول جذر مجموع مربع تفاوت‌ها تقسیم بر تعداد اعداد محاسبه می‌شود. در خط ۳۶ هم مقدار نهایی انحراف معیار با دقت دو رقم اعشار چاپ می‌شود. در خط ۳۹، یک لیست خالی به نام nums برای ذخیره‌ی اعداد ورودی کاربر ایجاد می‌شود. در خط ۴۰، تعداد اعدادی که کاربر می‌خواهد وارد کند از ورودی دریافت شده و به عدد صحیح تبدیل می‌شود. در خط های ۴۱ تا ۴۳، یک حلقه for که از 0 تا n-1 اجرا می‌شود تا کاربر اعداد خود را وارد کند. سپس از کاربر خواسته می‌شود که عدد خود را وارد کند و این عدد به عدد صحیح تبدیل می‌شود. بعد عدد وارد شده به انتهای لیست nums اضافه می‌شود. در خط ۴۵، تابع Max1 برای محاسبه و چاپ بیشترین عدد فراخوانی می‌شود. در خط ۴۶، تابع Min1 برای محاسبه و چاپ کمترین عدد فراخوانی می‌شود. در خط ۴۷، تابع Ave1 برای محاسبه و چاپ میانگین اعداد فراخوانی می‌شود. در خط ۴۸ هم تابع STD1 برای محاسبه و چاپ انحراف معیار فراخوانی می‌شود.

خروجی آن به صورت زیر است :

```
Enter the n:5
Enter number 1:12
Enter number 2:15
Enter number 3:9
Enter number 4:11
Enter number 5:84
The Maximum is:  84
The Minimum is:  9
The Average is:  26.2
The Standard Deviation is: 28.96
```

سوال 8 :

8- Give a number with 5 digits in main. If the number is not a 5 digits number, ask **repeatedly** from user to enter a valid number. Send the number for F1 function to find and return the maximum digit of the number. In the next step, send the maximum digit and the input number for F2 function to delete the maximum digit from number and return it. Finally, print the final output in main as shown in figure below.



حل سوال 8 :

روالی که برای حل این سوال درنظر گرفته شد بصورت زیر است :

```python
def F1(number):
    digits = []
    while number > 0:
        digit = number % 10
        digits.append(digit)
        number //= 10
    maxDigit = max(digits)
    return maxDigit


def F2(number, digit):
    number = str(number).replace(f"{digit}", "").strip()
    return number


num = input("Enter a 5 digits number:")
```

```python
while True:
    if len(num) == 5 and num.isdigit():
        break
    else:
        print("It is not a 5 digits number.")
        num = input("Please enter a valid number:")

num = int(num)
max_digit = F1(num)
print("The Maximum digit is:", max_digit)
print(f"The Final Digit without {max_digit} is:", F2(num, max_digit))
```

در این برنامه، در خط اول، تابعی به نام F1 تعریف می‌شود که یک عدد صحیح (number) را دریافت می‌کند و بزرگترین رقم آن را پیدا و برمی‌گرداند. در خط دوم، یک لیست خالی به نام digits تعریف می‌شود تا ارقام عدد به ترتیب در آن ذخیره شوند. در خط ۳، این حلقه while اجرا می‌شود تا زمانی که عدد number بزرگتر از صفر باشد. در خط ۴، با استفاده از عملگر درصد (٪)، رقم آخر عدد (number) به دست می‌آید و در متغیر digit ذخیره می‌شود. سپس در خط ۵، رقم آخر به لیست digits اضافه می‌شود. در خط ۶، عدد number به وسیله‌ی عملگر تقسیم صحیح (//) بر ۱۰ تقسیم می‌شود تا رقم آخر حذف شود و حلقه برای رقم بعدی تکرار شود. در خط ۷، بعد از اتمام حلقه، بیشترین مقدار در لیست digits (بزرگترین رقم عدد) با استفاده از تابع max پیدا شده و در متغیر maxDigit ذخیره می‌شود. در خط ۸، بزرگترین رقم(maxDigit) برگردانده می‌شود.

در خط ۱۱، تابع دیگری به نام F2 تعریف می‌شود که یک عدد (number) و یک رقم (digit) را دریافت کرده و تمام تکرارهای آن رقم را از عدد حذف می‌کند. در خط ۱۲، عدد number به رشته تبدیل می‌شود و تمام تکرارهای رقم digit از آن حذف می‌شود. از replace برای حذف استفاده شده و strip برای حذف فضاهای اضافی احتمالی. در خط ۱۳، رشته‌ی اصلاح شده عدد (بدون رقم مشخص‌شده) برگردانده می‌شود.

در خط ۱۶، از کاربر خواسته می‌شود که یک عدد پنج رقمی وارد کند و این مقدار به صورت رشته در متغیر num ذخیره می‌شود. در خط ۱۸، یک حلقه while بی‌نهایت تعریف می‌شود که تا زمانی که شرط معتبر بودن عدد برقرار نباشد، از کاربر دوباره درخواست عدد می‌کند. در خط ۱۹، این شرط بررسی می‌کند که آیا عدد ورودی دقیقا پنج رقمی است و فقط شامل ارقام عددی (نه کاراکترهای دیگر) است یا خیر. اگر شرط معتبر بود، حلقه شکسته می‌شود و از آن خارج می‌شود. در صورتی که عدد ورودی نامعتبر باشد، بخش else اجرا می‌شود. در خط ۲۲، پیامی چاپ می‌شود که کاربر را مطلع می‌کند عدد وارد شده پنج رقمی نیست. در خط ۲۳، دوباره از کاربر درخواست می‌شود که یک عدد معتبر وارد کند. در خط ۲۵، عدد وارد شده (که یک رشته بود) به عدد صحیح تبدیل می‌شود. در خط ۲۶، تابع F1 فراخوانی می‌شود تا بزرگترین رقم عدد پیدا شود و نتیجه در متغیر max_digit ذخیره می‌شود. در خط ۲۷، بزرگترین رقم عدد چاپ می‌شود. در خط ۲۸، تابع F2 فراخوانی می‌شود تا تمام تکرارهای رقم بزرگترین حذف شده و عدد نهایی چاپ می‌شود.

خروجی آن به صورت زیر است :

```
Enter a 5 digits number:125
It is not a 5 digits number.
Please enter a valid number:43
It is not a 5 digits number.
Please enter a valid number:34675
The Maximum digit is: 7
The Final Digit without 7 is: 3465
```

سوال 9 :

9-**Global** and **Local** variables: Run codes P1, P2, P3 and P4 and report outputs. Explain reasons for their outputs in details.

P1

```
def f():
    # local variable
    s = "I live in Khorramabad"
    print(s)

# Main
f()
```

P2

```
def f():

    # local variable
    s = "I live in Khorramabad"
    print("Inside Function:", s)

# Main
f()
print(s)
```

P3

```
def f():
    global s
    s ='I live in Khorramabad.'
    print(s)

# Main: Global Scope
s = 'I live in Iran.'
f()
print(s)
```

P4

```
# This function uses global variable s
def f():
    s="I live in Khorramabad"
    print(s)

# Main
s = "I live in Iran"
f()
print(s)
```

حل سوال 9 :

خروجی **P1** بصورت زیر است :

```
I live in khorramabad
```

در کد P1، متغیر s به عنوان یک متغیر محلی درون تابع f تعریف شده و مقدار رشته ای I live in khorramabad به آن اختصاص داده شده است. چون s یک متغیر محلی است، فقط در محدوده تابع f معتبر است و زمانی که تابع فراخوانی می شود، مقدار آن چاپ می شود. به همین دلیل، خروجی تابع این رشته را نمایش می دهد. پس از اتمام اجرای تابع، متغیر s از حافظه حذف می شود و دیگر قابل دسترسی نیست.

خروجی **P2** بصورت زیر است :

```
Traceback (most recent call last):
  File "E:\pythonProject\Advanced-Database-Homework-1\main.py", line 254, in <module>
    print(s)
          ^
NameError: name 's' is not defined
Inside Function: I live in khorramabad
```

در کد P2، خطای NameError: name 's' is not defined به این دلیل رخ می‌دهد که متغیر s به صورت محلی درون تابع f تعریف شده است و خارج از آن قابل دسترسی نیست. وقتی تابع f اجرا می‌شود، رشته I live in khorramabad درون تابع چاپ می‌شود، اما پس از آن،

در خط بعدی که قصد داریم متغیر s را در فضای اصلی برنامه (بیرون از تابع) چاپ کنیم، این متغیر وجود ندارد، چون به عنوان یک متغیر محلی فقط در محدوده تابع تعریف شده است. به همین دلیل، برنامه خطای نام s تعریف نشده را می‌دهد.

خروجی **P3** بصورت زیر است :

```
I live in khorramabad.
I live in khorramabad.
```

در کد P3، متغیر s ابتدا به عنوان یک متغیر سراسری با مقدار I live in iran. تعریف شده است. سپس درون تابع f با استفاده از کلمه کلیدی global، متغیر s سراسری تغییر داده می‌شود و مقدار جدید I live in khorramabad. به آن اختصاص می‌یابد. هنگام چاپ داخل تابع و پس از آن در فضای اصلی برنامه، مقدار تغییر یافته‌ی s، یعنی I live in khorramabad. ، نمایش داده می‌شود. به همین دلیل هر دو خروجی یکسان هستند.

خروجی **P4** بصورت زیر است :

```
I live in khorramabad.
I live in iran.
```

در کد P4، متغیر s درون تابع به صورت محلی تعریف شده و مقدار "I live in khorramabad." چاپ می‌شود. اما متغیر سراسری s با مقدار "I live in iran." خارج از تابع بدون تغییر باقی می‌ماند و پس از فراخوانی تابع، مقدار سراسری آن چاپ می‌شود. بنابراین خروجی دو مقدار متفاوت دارد.

سوال 10 :

10-Study about **Recursive Functions** in python. Explain in details that how does the following recursive function work?

```
def factorial(x):
    if x == 1: # This is the base case
        return 1

    else: # This is the recursive case
        return(x * factorial(x-1))

print(factorial(4))
```

حل سوال 10 :

در این کد، تابع بازگشتی factorial برای محاسبه فاکتوریل یک عدد استفاده می‌شود. فاکتوریل عدد x برابر است با حاصل‌ضرب تمام اعداد از 1 تا x. ساختار تابع به گونه‌ای است که از شرط پایه و فراخوانی بازگشتی تشکیل شده است.

شرط پایه (Base Case) :

شرط if x == 1 بررسی می‌کند که آیا عدد x برابر 1 است. در این حالت، تابع 1 را برمی‌گرداند و فرآیند بازگشتی متوقف می‌شود. زیرا !1 همان 1 است.

فراخوانی بازگشتی (Recursive Case) :

اگر x بزرگتر از 1 باشد، تابع factorial(x - 1) فراخوانی می‌شود و مقدار x در نتیجه‌ی فاکتوریل عدد قبلی ضرب می‌شود.

مثال برای factorial(4) :

- **فراخوانی اول** factorial(4) : تابع بررسی می‌کند که x برابر 1 نیست، پس بازگشتی انجام می‌شود 4 * factorial(3)
- **فراخوانی دوم** factorial(3) : تابع دوباره بازگشتی انجام می‌دهد 3 * factorial(2)
- **فراخوانی سوم** factorial(2) : بازگشتی دیگر 2 * factorial(1)
- **فراخوانی چهارم** factorial(1) : چون x == 1 است، مقدار 1 برگردانده می‌شود.
- سپس محاسبات به ترتیب انجام می‌شود : 2 = 1 * 2، 6 = 2 * 3، و در نهایت 24 = 6 * 4

نتیجه‌ی نهایی factorial(4) برابر با 24 است.