



البرمجة الإجرائية **Procedural Programming**

IPG202

الفصل السابع **تمارين عملية على الطرائق، الاستثناءات والملفات** **Practice On Methods, Exceptions, and Files**

الكلمات المفتاحية

طريقة، استدعاء، بارامتر، معامل، مصفوفة، عودية، مرجع، مصفوفة، استثناء، ملف.

ملخص الفصل

يتضمن هذا الفصل مجموعة من التمارين التطبيقية على الطرائق والاستثناءات والملفات وذلك بهدف تعزيز المفاهيم النظرية التي تم عرضها خلال الفصول الثلاثة السابقة واكتساب المهارات العملية في التعامل أنماط المسائل التي تتضمن تعاملًا مع هذه المفاهيم.

أهداف الفصل

بنهاية هذا الفصل سيكون الطالب قادراً على:

- تعزيز مهاراته في تعريف الطرائق والاستثناءات والملفات.
- تصميم حلول للمسائل التي تتضمن عدة وظائف عن طريق التجزئة إلى برامج جزئية.
- كتابة الشيفرة البرمجية بلغة C# للمسائل التي تتضمن استثناءات وملفات.
- تنفيذ البرامج واكتشاف الأخطاء وتصحيحها.

محتويات الفصل

1. التمرين الأول : حساب المجاميع
2. التمرين الثاني : رسم الأشكال الهندسية
3. التمرين الثالث : فرز مصفوفة أحادية.
4. التمرين الرابع : عمليات على مصفوفة أحادية.
5. التمرين الخامس : معالجة استثناءات الدخل.
6. التمرين السادس : معالجة قيم مخزنة في ملف

1- التمرين الأول: حساب المجاميع

ليكن لدينا المجموع التالي:

$$y = \sum_{k=0}^n \frac{x^k}{k!}$$

المطلوب استخدام الطرائق في حساب هذا المجموع.

تمهيد للحل:

بالنظر إلى صيغة المجموع المحدد نلاحظ أن هذه المسألة هي مسألة مركبة ويمكن تجزئتها إلى ثلاث وظائف:

- وظيفة لحساب عاملي عدد.
- وظيفة لحساب رفع عدد إلى قوة.
- وظيفة لحساب المجموع.

```
//A Method to calculate factorial
public static int factorial(int x)
{
    if (x == 0)
        return 1;
    else
        return x * factorial(x - 1);
}

//A Method to calculate power
public static double power(double x,int y)
{
    if (y == 0)
        return 1;
    else
        return x * power(x,y - 1);
}

//A Method to calculate sum
public static double sum(double x, int n)
{
    double result = 0;
    for (int i = 0; i <= n; i++)
        result = result + (power(x,i)/factorial(i));
    return result;
}
```

```

static void Main(string[] args)
{
    double A;
    int B;
    Console.Write("Enter a double value :");
    A = Double.Parse(Console.ReadLine());
    Console.Write("Enter a positive integer value :");
    B = Int32.Parse(Console.ReadLine());

    Console.WriteLine("The Sum is: "+sum(A, B));
}

```

فيما يلي عينة عن تنفيذ البرنامج:

```

C:\WINDOWS\system32\cmd.exe
Enter a double value :2.3
Enter a positive integer value :10
The Sum is: 9.97388834337326
Press any key to continue . . .

```

مابعد الحل:

قم بتوظيف ماتعلمته في هذا التمرين في حساب مجاميع وجداءات مختلفة من مثل:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

$$\ln(x) = -\sum_{n=1}^{\infty} \frac{(1-x)^n}{n} = -(1-x) - \frac{(1-x)^2}{2} - \frac{(1-x)^3}{3} - \dots$$

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \text{ for all } x$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \text{ for all } x$$

2- التمرين الثاني: رسم الأشكال الهندسية

المطلوب كتابة طريقة لطباعة مستطيل من تكرار محرف معين وبأبعاد يتم تمريرها للطريقة، أي يقوم بطباعة شكل من النوع (في المثال مستطيل بطول 8 وعرض 4 من محرف *) :

* * * * *
 * * * * *
 * * * * *
 * * * * *

```
//A Method to draw a rectangle
public static void draw(int length, int width = 3, char ch= '*')
{
    for (int i = 1; i <= width; i++)
    {
        for (int j = 1; j <= length; j++)
            Console.Write(ch);
        Console.WriteLine();
    }
}

static void Main(string[] args)
{
    for(int i=6;i<=36;i+=6)
        draw(i);
}
```

فيما يلي عينة عن تنفيذ البرنامج (قمنا برسم شكل درج):

[illegible]

مابعد الحل:

في هذا المثال قمنا باستخدام مفهوم الوسطاء الافتراضية لإعطاء قيم افتراضية للمعاملين width و char. حاول استخدام الأسلوب ذاته في كتابة طرائق لرسم مختلف الأشكال الهندسية الممكنة:

- مثلث (قائم الزاوية، متساوي الساقين، متساوي الأضلاع،).
- مربع.
- معين.
- متوازي أضلاع.

3- التمرين الثالث: فرز مصفوفة أحادية

المطلوب استخدام مفهوم الطرائق لإدخال قيم مصفوفة أحادية وترتيب هذه القيم تصاعدياً ومن ثم طباعتها.

```
//A Method to input array elements
public static void input_array(int [] marks)
{
    for (int i = 0; i < marks.Length; i++)
        marks[i] = Int32.Parse(Console.ReadLine());
}

//A Method to output array elements
public static void output_array(int[] marks)
{
    for (int i = 0; i < marks.Length; i++)
        Console.Write(marks[i] + " ");
    Console.WriteLine();
}

//A Method to swap 2 values
public static void swap(ref int a,ref int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}

//A Method to sort array elements
public static void sort_array(int[] marks)
{
    for(int pass=1;pass<marks.Length;pass++)
```

```

        for (int i = 0; i < marks.Length-1; i++)
            if (marks[i]>marks[i+1])
                swap(ref marks[i],ref marks[i+1]);
    }
    static void Main(string[] args)
    {
        int[] students_marks = new int[10];
        input_array(students_marks);
        Console.WriteLine("Students Marks before Sort:");
        output_array(students_marks);
        sort_array(students_marks);
        Console.WriteLine("Students Marks after Sort:");
        output_array(students_marks);
    }

```

فيما يلي عينة عن تنفيذ البرنامج:

```

C:\WINDOWS\system32\cmd.exe
76
66
74
46
83
72
37
62
87
96
Students Marks before Sort:
76 66 74 46 83 72 37 62 87 96
Students Marks after Sort:
37 46 62 66 72 74 76 83 87 96
Press any key to continue . . .

```

ما بعد الحل:

في هذا المثال قمنا باستخدام مفهوم التمرير بالمرجع لمبادلة قيمتي متحولين. كما قمنا بتنفيذ عملية الفرز باستخدام مفهوم الفرز الفقاعي ... قم بالتفتيش عن خوارزميات فرز أخرى وكتابة طرائق لتنفيذها، كمثال:

- خوارزمية الفرز بالتبديل.
- خوارزمية الفرز بالحثر.

4- التمرين الرابع: عمليات على مصفوفة أحادية

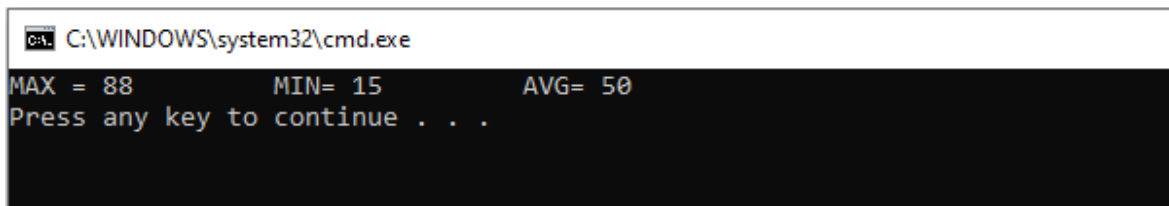
المطلوب كتابة طريقة واحدة لإيجاد القيمة العظمى والقيمة الصغرى والمتوسط الحسابي لقيم مصفوفة أحادية البعد ومن ثم طباعتها.

```
public static void MaxMinAvg_array(int [] a,out int max,out int min,out int avg)
{
    max = a[0];
    min = a[0];
    int sum = a[0];

    for (int i = 1; i < a.Length; i++)
    {
        if (a[i] > max)
            max = a[i];
        if (a[i]<min)
            min=a[i];
        sum+=a[i];
    }
    avg = sum / a.Length;
}

static void Main(string[] args)
{
    int[] b = {20,65,22,15,88,66,76};
    int mx,mn,av;
    MaxMinAvg_array(b, out mx, out mn, out av);
    Console.WriteLine("MAX = {0} \t MIN= {1} \t AVG= {2}", mx, mn, av);
}
```

فيما يلي عينة عن تنفيذ البرنامج:



```
C:\WINDOWS\system32\cmd.exe
MAX = 88      MIN= 15      AVG= 50
Press any key to continue . . .
```

مابعد الحل:

في هذا المثال قمنا باستخدام مفهوم التمرير كمعامل خرج لإعادة القيم المراد حسابها.

5- التمرين الخامس: معالجة استثناءات الإدخال

المطلوب كتابة برنامج لإدخال قيم مبيعات تاجر يومياً حساب مجموع المبيعات خلال شهر.
يطلب في هذا البرنامج معالجة حالة استثناءات إدخال القيم السالبة أو القيم غير الرقمية.

```
static void Main(string[] args)
{
    int daily_sales;
    int sum_sales = 0;

    try
    {
        int i = 1;
        while (i <= 30)
        {
            Console.WriteLine("Enter sales for day No: {0}", i);
            daily_sales = Int32.Parse(Console.ReadLine());
            if (daily_sales < 0)
                throw new Exception("sales must be positive");
            sum_sales = sum_sales + daily_sales;
            i++;
        }
    }

    catch (FormatException ex)
    {
        Console.WriteLine("An Error Occured");
        Console.WriteLine(ex.Message);
    }

    catch (Exception ex)
    {
        Console.WriteLine("An Error Occured");
        Console.WriteLine(ex.Message);
    }
}
```

فيما يلي عينة عن تنفيذ البرنامج:

```
C:\WINDOWS\system32\cmd.exe
Enter sales for day No1 :77
Enter sales for day No2 :65
Enter sales for day No3 :test
An Error Occured
Input string was not in a correct format.
Press any key to continue . . .
```

```
C:\WINDOWS\system32\cmd.exe
Enter sales for day No1 :87
Enter sales for day No2 :98
Enter sales for day No3 :-877
An Error Occured
sales must be positive
Press any key to continue . . .
```

6- التمرين السادس: معالجة قيم مخزنة في ملف

في هذا البرنامج لدينا ملف يدعى input.txt يحوي قائمة من القيم الصحيحة المخزنة كل قيمة على سطر منفصل. يطلب كتابة برنامج يقوم بإنشاء ملف يدعى output.txt بحيث يقوم البرنامج بقراءة الأرقام المخزنة في الملف input.txt رقماً رقماً ومن ثم اختبار كل رقم تمت قراءته فيما إذا كان تاماً أو لا، بحيث يتم تخزين الأعداد التامة في الملف output.txt.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace FileInOut
{
```

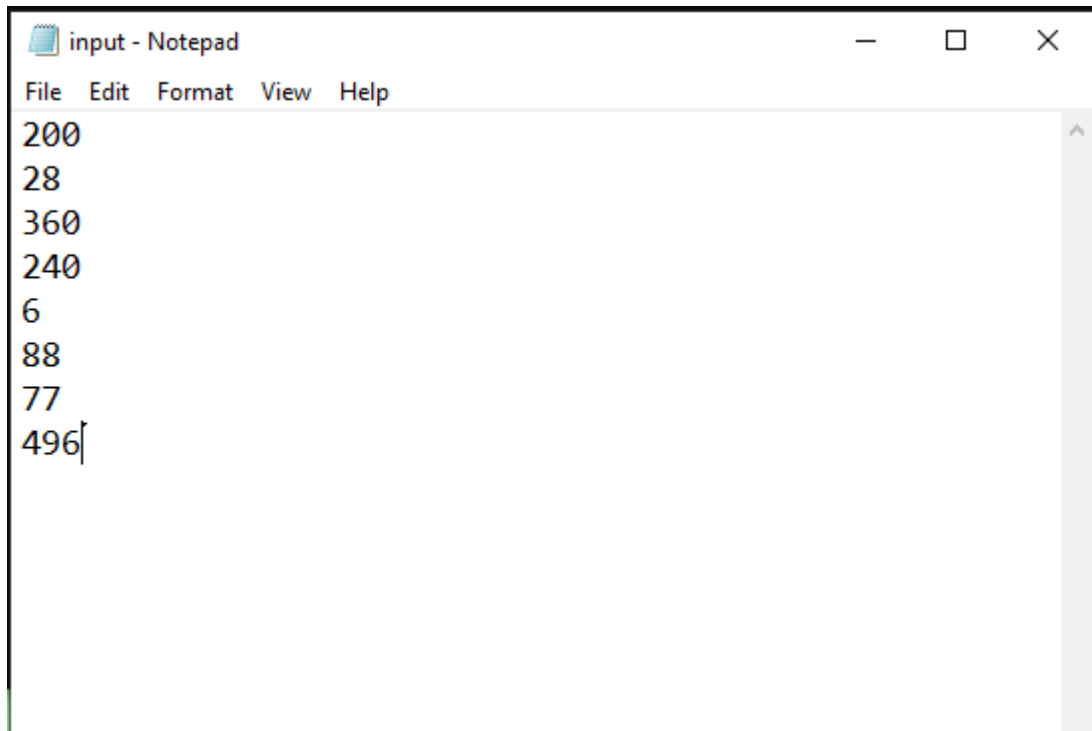
```

class Program
{
    public static bool perfect(int m)
    {
        int sum = 0;
        for (int i = 1; i < m; i++)
            if (m % i == 0)
                sum += i;
        if (sum == m)
            return true;
        else
            return false;
    }

    static void Main(string[] args)
    {
        FileStream fsr;
        StreamReader fr;
        FileStream fsw;
        StreamWriter fw;
        fsr = new FileStream("d:\\input.txt", FileMode.Open, FileAccess.Read);
        fsw = new FileStream("d:\\output.txt", FileMode.OpenOrCreate, FileAccess.Write);
        fr = new StreamReader(fsr);
        fw = new StreamWriter(fsw);
        int content;
        while (!fr.EndOfStream)
        {
            content = Int32.Parse(fr.ReadLine());
            if (perfect(content))
                fw.WriteLine(content);
        }
        fr.Close();
        fsr.Close();
        fw.Close();
        fsw.Close();
    }
}

```

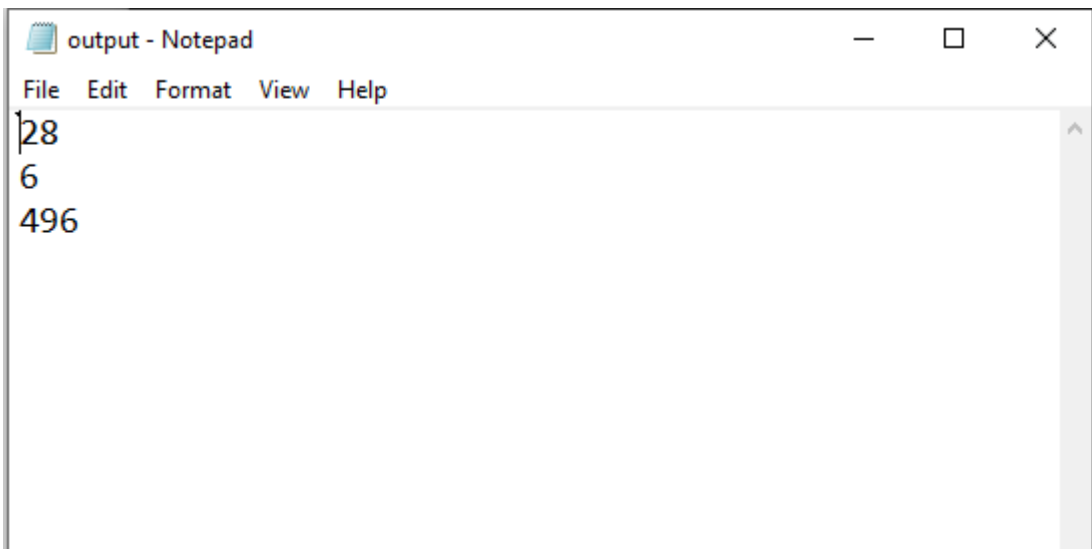
إذا قمنا بتجربة هذا البرنامج وكان لدينا ملف input.txt من الشكل التالي:



A screenshot of a Notepad window titled "input - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following numbers, each on a new line: 200, 28, 360, 240, 6, 88, 77, and 496. The cursor is positioned at the end of the last line.

```
200
28
360
240
6
88
77
496
```

فإن البرنامج سيقوم بإنشاء ملف output.txt ويكون محتواه على الشكل:



A screenshot of a Notepad window titled "output - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the following numbers, each on a new line: 28, 6, and 496. The cursor is positioned at the end of the first line.

```
28
6
496
```