

## BÀI THỰC HÀNH SỐ 1

### I. Mục tiêu.

Hiểu và nắm vững các đặc điểm cú pháp, câu lệnh, lập trình tổng quát trong lập trình C#.

### II. Thời gian

Thực hành: 5 tiết

### III. Hướng dẫn chung

**Câu 1:** Tạo project mới trong Visual Studio Community 2019 và thực thi chương trình Hello World với Console App

#### Hướng dẫn:

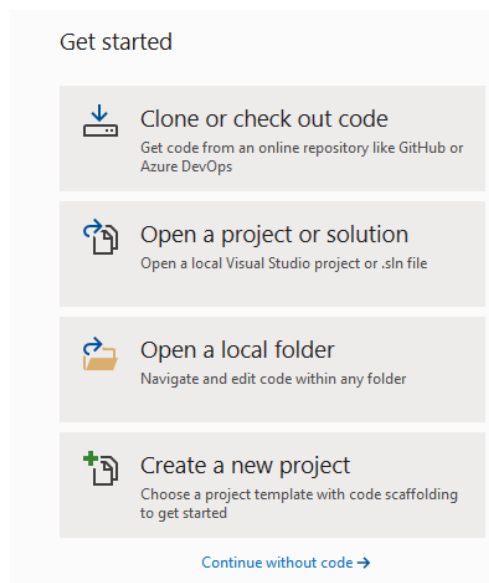
- **Bước 1:** Download Visual Studio Community 2019 tại:

<https://visualstudio.microsoft.com/free-developer-offers/>

- **Bước 2:** Cài đặt VS Community.

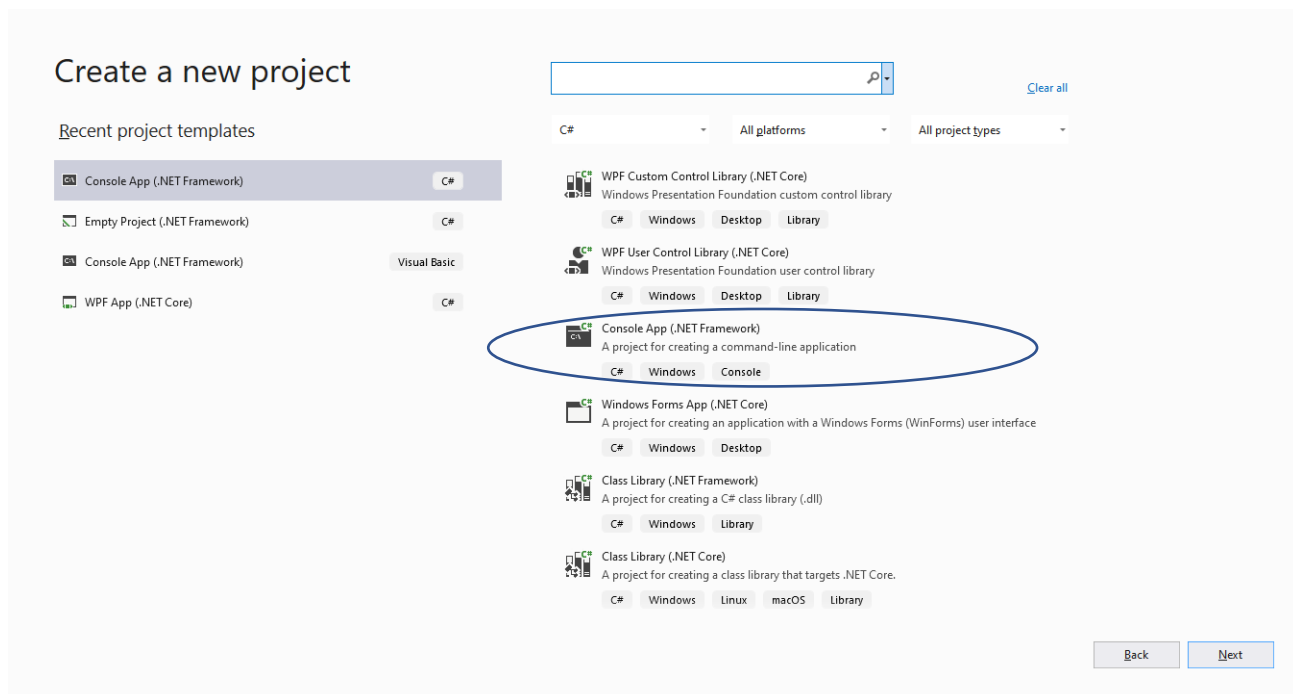
- **Bước 3:** Mở VS, ở màn hình chính của VS, chọn **Create a new project** hoặc

**Continue without code**

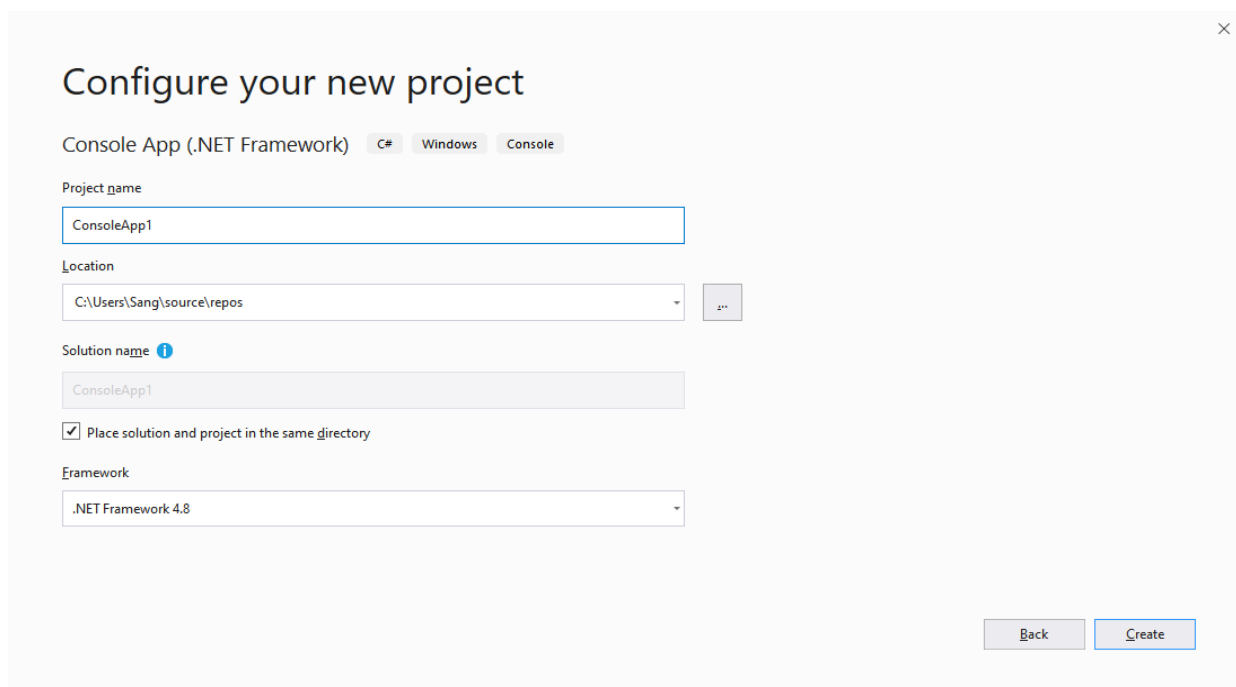


- Nếu chọn **Continue without code** sẽ xuất hiện cửa sổ mới và tiếp tục chọn **File=> New => Project...**

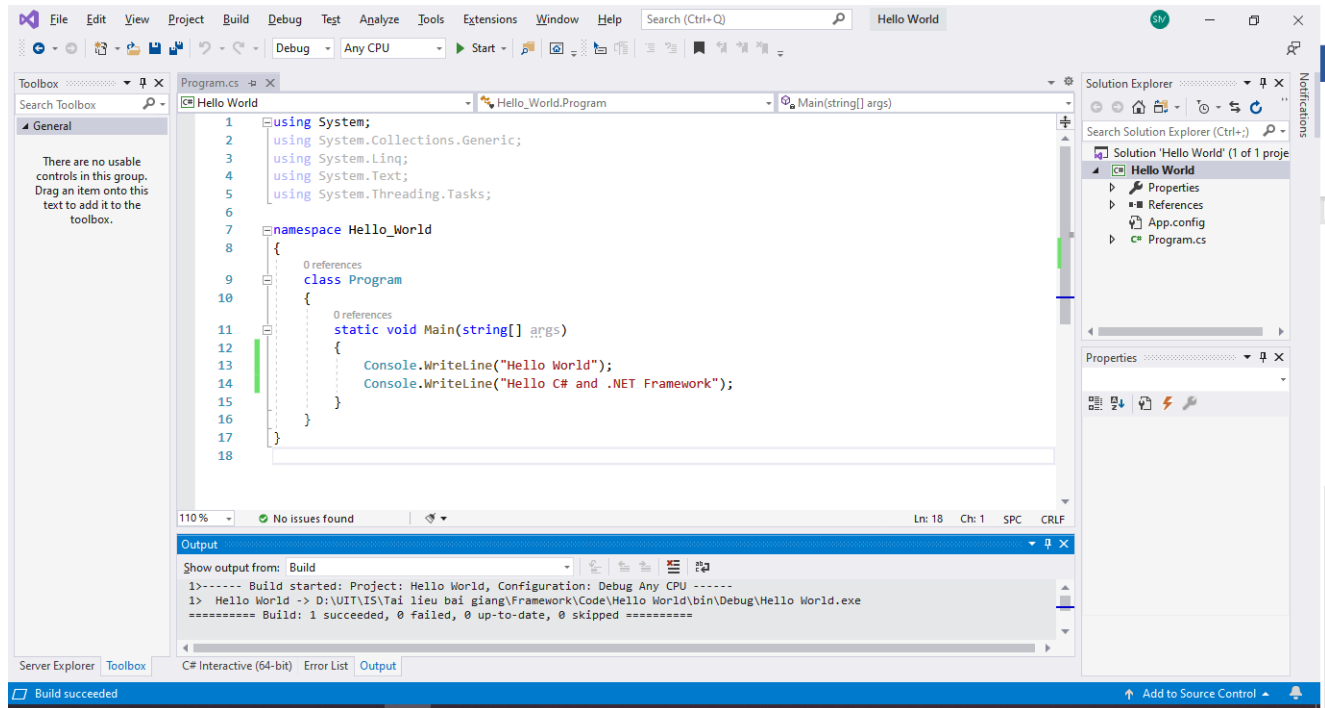
○ **Bước 4:** Xuất hiện cửa sổ **Create a new Project**, chọn **ConsoleApp (.NET Framework)** => **Next**



○ **Bước 5:** Điền tên **Project name** và chọn đường dẫn lưu ứng dụng trong cửa sổ **Location**. Chọn **check** ở **Place solution...** để tạo solution và project cùng thư mục (Trong một solution có thể có nhiều project). Có thể lựa chọn phiên bản .NET Framework ở khung **Framework**. Cuối cùng chọn **Create**.



○ **Bước 6:** Thêm các đoạn code vào trong hàm main (như hình). Sau đó trên thanh menu của Visual Studio chọn **Debug => Start Without Debugging** hoặc nhấn **Ctrl + F5** để thực thi ứng dụng.

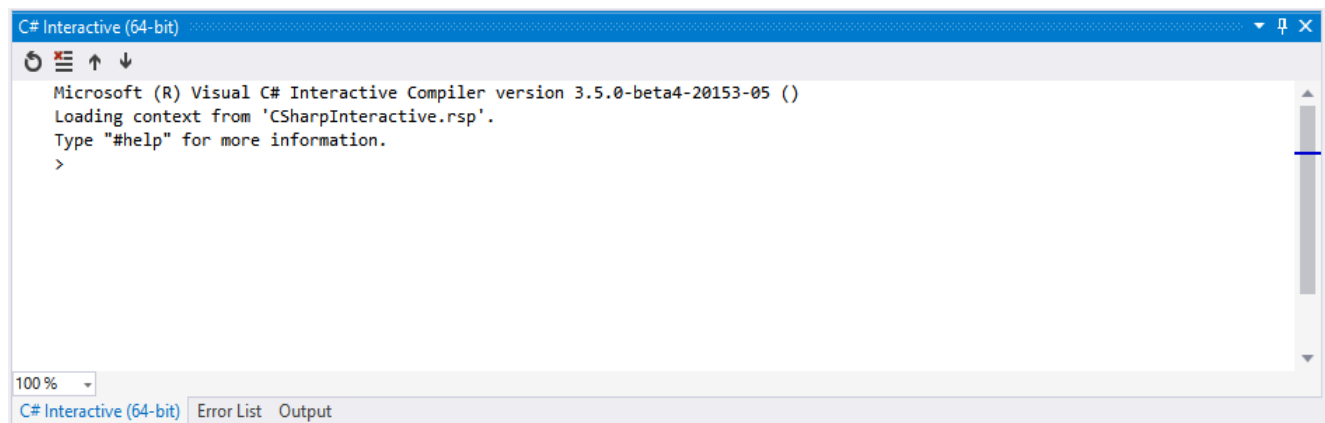


## Câu 2: Sử dụng C# interactive.

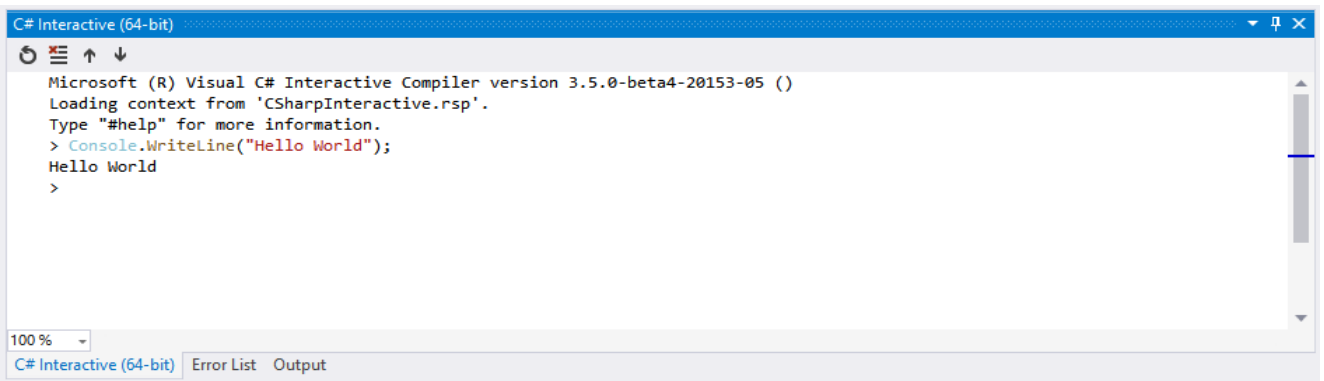
Visual Studio cung cấp một công cụ hỗ trợ lập trình cho lập trình viên trong việc thử nghiệm các cú pháp kiểu dữ liệu, cấu trúc điều khiển... cơ bản, hoặc khi thực thi một vài dòng code đơn giản; được gọi là C# interactive.

Đây là giao diện dòng lệnh đặc biệt cho phép viết và thực thi từng dòng hoặc một nhóm các câu lệnh C# mà không cần tạo project.

Trên thanh menu của Visual Studio chọn **View => Other windows => C# interactive**

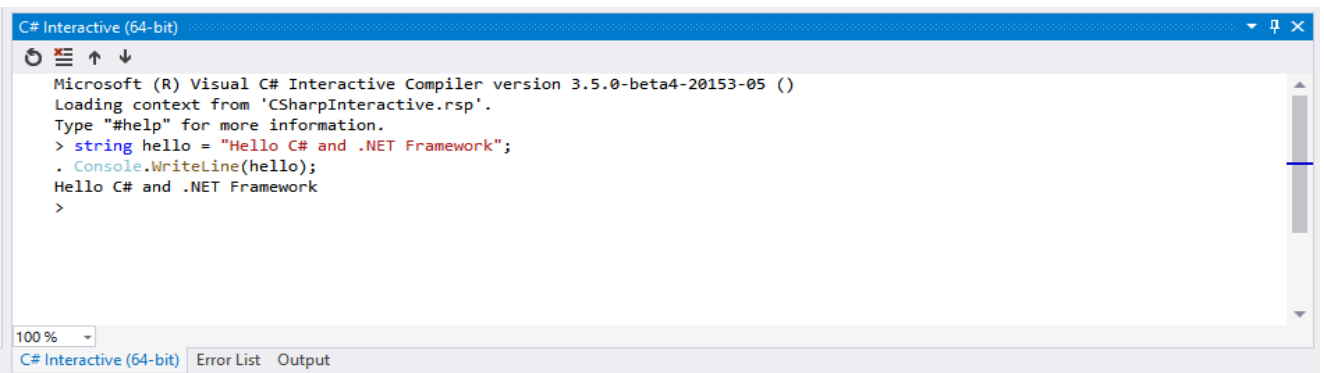


Nhập câu lệnh của C# vào và Enter thì lệnh sẽ được thực thi



```
C# Interactive (64-bit)
Microsoft (R) Visual C# Interactive Compiler version 3.5.0-beta4-20153-05 ()
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> Console.WriteLine("Hello World");
Hello World
>
```

Để thực thi một đoạn lệnh, kết thúc mỗi dòng ấn Shift + Enter; khi đó dòng lệnh sẽ không được thực thi ngay, và tiếp tục viết câu lệnh khác. Khi kết thúc câu lệnh cuối nhấn Enter để thực thi.



```
C# Interactive (64-bit)
Microsoft (R) Visual C# Interactive Compiler version 3.5.0-beta4-20153-05 ()
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> string hello = "Hello C# and .NET Framework";
. Console.WriteLine(hello);
Hello C# and .NET Framework
>
```

### Một số lưu ý khi dùng C# interactive:

- Chỉ xuất ra console bằng các câu lệnh `Console.WriteLine`, `Console.Write`.
- Không thể đọc dữ liệu từ các câu lệnh `Console.Read`, `Console.ReadKey`, `Console.ReadLine`.
- Hỗ trợ intellisense như giao diện viết code bình thường.
- Muốn in kết quả của một biến có thể dùng câu lệnh `Console.WriteLine`, `Console.Write` hoặc viết tên biến (không viết dấu chấm phẩy sau tên biến) và Enter.

**Câu 3:** Viết chương trình cho phép nhập họ tên sinh viên và địa chỉ từ bàn phím, sau đó xuất ra màn hình họ tên và địa chỉ vừa mới nhập vào.

### Hướng dẫn: Nhập xuất cơ bản trong C# với Console

Trong C# có 5 lệnh cơ bản dùng để nhập/xuất giá trị vào Console:

- `Console.Write(<giá trị cần in ra màn hình>)`: xuất giá trị ra màn hình console nhưng không đưa con trỏ xuống dòng.

- `Console.WriteLine(<giá trị cần in ra màn hình>)`: xuất xuất giá trị ra màn hình console nhưng đưa con trỏ xuống dòng.
- `Console.Read()`: đọc một ký tự từ bàn phím và trả về số nguyên là mã ASCII.
- `Console.ReadKey()`: đọc một ký tự và trả về kiểu `ConsoleKeyInfo`. Khi người dùng nhập một phím bất kỳ, `ReadKey` sẽ đọc và kết thúc nhập ngay. Được dùng để dừng màn hình console và chờ nhấn phím bất kỳ để tiếp tục.
- `Console.ReadLine()`: đọc một dòng (kết thúc bằng nhấn Enter) và trả về một chuỗi. Vì thế để nhập số, phải chuyển đổi chuỗi sang kiểu số.
- Một số thuộc tính và phương thức bổ sung cho console:
  - `Console.BackgroundColor`: đặt màu nền cho văn bản.
  - `Console.ForegroundColor`: đặt màu văn bản.
  - `Console.Title`: đặt tiêu đề cho console.
  - `Console.OutputEncoding = Encoding.UTF8`: để xuất chữ theo unicode.
- Biến đổi dữ liệu từ console:
  - Do dữ liệu nhập vào từ console sử dụng hàm `ReadLine` đều là kiểu `string` nên phải thực hiện đổi kiểu chuỗi về số gọi là **type conversion**.
  - Sử dụng 2 phương thức `Parse()` hoặc `bool TryParse()` để thực hiện.
  - VD:
 

```
string input = Console.ReadLine();
int number = int.Parse(input);

input = Console.ReadLine();
bool check = bool.Parse(input);
```
- In chuỗi có định dạng với placeholder:
  - Dùng phương thức `Write/WriteLine` để in ra chuỗi có kèm các biến vào các vị trí đã định sẵn trong chuỗi.
  - Vị trí đánh dấu được (placeholder) được biểu diễn bằng cụm `{0}`, `{1}` thay thế cho các biến xuất hiện theo thứ tự.
 

```
int x = 10, y = 20;
WriteLine("Tọa độ x= {0}, y= {1}", x, y);
KQ: Tọa độ x= 10, y= 20
```

Với thuộc tính màu:  
`ConsoleColor.<màu>`

- In chuỗi có định dạng với **string interpolation**:
  - o In chuỗi bắt đầu bằng ký tự \$ và ở những vị trí đặt biểu thức/biến vào sẽ được đặt trong cặp dấu {}.

```
WriteLine($"Sum: {x} + {y} = {x + y}");
```

**KQ:** Sum: 10 + 20 = 30

- o Có thể đặt những biểu thức phức tạp như: phép toán điều kiện, các đoạn code vào trong cặp dấu {}

### Viết chương trình nhập xuất cơ bản:

- o **Bước 1:** Tạo một project mới như đã hướng dẫn ở câu 1.
- o **Bước 2:** Viết mã cho hàm Main(); sử dụng hàm ReadLine() để nhập họ tên và địa chỉ.
- o **Bước 3:** Dùng hàm WriteLine() hoặc Write() để xuất họ tên và địa chỉ ra màn hình.

```
static void Main(string[] args)
{
    OutputEncoding = Encoding.UTF8;
    Write("Nhập họ tên sinh viên: ");
    string name = Console.ReadLine();
    Write("Nhập địa chỉ sinh viên: ");
    string address = Console.ReadLine();

    WriteLine("Thông tin của sinh viên:");
    WriteLine("Họ và tên: {0}\t Địa chỉ: {1}", name, address);
}
```

### Câu 4: Nhập vào 3 số nguyên và in ra số lớn nhất:

- o **Cách 1:** sử dụng **if - else**

```
public void TimMax()
{
    OutputEncoding = Encoding.UTF8;
    Write("Nhập số thứ 1: ");
    int a = int.Parse(ReadLine());
    Write("Nhập số thứ 2: ");
    int b = int.Parse(ReadLine());
    Write("Nhập số thứ 3: ");
    int c = int.Parse(ReadLine());

    int max = a;
    if (max < b)
    {
        max = b;
    }
    if (max < c)
    {
        max = c;
    }

    WriteLine($"Số lớn nhất: { max}");
}
```

○ Cách 2: sử dụng `switch - case`

```
public void TimMax()
{
    OutputEncoding = Encoding.UTF8;
    Write("Nhập số thứ 1: ");
    int a = int.Parse(ReadLine());
    Write("Nhập số thứ 2: ");
    int b = int.Parse(ReadLine());
    Write("Nhập số thứ 3: ");
    int c = int.Parse(ReadLine());

    int max = a;

    switch (max > b)
    {
        case true:
            switch (max > c)
            {
                case false:
                    max = c;
                    break;
            }
            break;
        case false:
            switch (b > c)
            {
                case true:
                    max = b;
                    break;
                case false:
                    max = c;
                    break;
            }
            break;
    }

    WriteLine($"Số lớn nhất: { max}");
}
```

- **Cách 3:** sử dụng toán tử điều kiện ?

```
public void TimMax()
{
    OutputEncoding = Encoding.UTF8;
    Write("Nhập số thứ 1: ");
    int a = int.Parse(ReadLine());
    Write("Nhập số thứ 2: ");
    int b = int.Parse(ReadLine());
    Write("Nhập số thứ 3: ");
    int c = int.Parse(ReadLine());

    int max = a > b ? a > c ? a : c : b > c ? b : c;
    WriteLine($"Số lớn nhất: {max}");
}
```

#### IV. Thực hành

**Mục tiêu:** Các bài tập trong phần này giúp sinh viên có khả năng:

- Cách viết code sử dụng Visual Studio.
- Hiểu rõ được các kiểu dữ liệu của C#.
- Sử dụng thành thạo các loại toán tử của C#.
- Sử dụng thành thạo lệnh rẽ nhánh **If**, lệnh lựa chọn **Switch**.
- Sử dụng thành thạo các cấu trúc lặp.
- Xử lý chuỗi ký tự.

**Câu 1:** Viết chương trình nhập vào bán kính của đường tròn, tính chu vi đường tròn đó.

Hướng dẫn: sử dụng hằng số Math.PI để tính chu vi.

**Câu 2:** Viết chương trình nhập 2 phân số. Tính tổng, hiệu, tích, thương của 2 phân số nói trên.

Hướng dẫn: Có thể sử dụng **struct** để tạo phân số.

```
struct PHAN_SO
{
    public int Tu_So, Mau_So;
}

//Khai báo và sử dụng phân số
PHAN_SO x, y;
x.Tu_So = 2;
y.Mau_So = 3;
```



**Câu 3:** Viết chương trình nhập vào họ tên, ngày sinh và giới tính của nhân viên, rồi tính ngày về hưu của nhân viên.

Hướng dẫn: Sử dụng hàm **DateTime** để xử lý ngày tháng. Tuổi nghỉ hưu của nam là 60 và nữ là 55.

**Câu 4:** Viết hàm nhập vào 2 chuỗi x và y. Thực hiện các thao tác sau:

- Cho biết tổng chiều dài của chuỗi x. (Hướng dẫn: sử dụng thuộc tính **Length**)
- Cho biết 3 ký tự đầu tiên của chuỗi x. (Hướng dẫn: sử dụng hàm **Substring**);
- Cho biết 3 ký tự cuối của chuỗi x.
- Cho biết ký tự thứ 6 của chuỗi x.
- Tạo chuỗi mới gồm 3 ký tự đầu tiên của chuỗi x và 3 ký tự cuối của chuỗi y.
- Kiểm tra 2 chuỗi x, y có bằng nhau hay không (Hướng dẫn: sử dụng hàm **Equals**)
- Cho biết y có xuất hiện trong x hay không? Nếu có, tại vị trí nào? (Hướng dẫn: sử dụng hàm **indexOf**)
- Cho biết tất cả các vị trí xuất hiện của y trong x.

Lưu ý: Tham khảo thêm về các hàm xử lý chuỗi: <https://docs.microsoft.com/en-us/dotnet/api/system.string?view=netframework-4.8>

**Câu 5:** Viết hàm nhập vào số điện sử dụng của tháng và tính tiền điện theo phương pháp lũy tiến.

- Nếu số điện  $\leq 50$  kw thì giá mỗi số điện là 2000 đ.
- Nếu số điện  $> 50$  kw và  $\leq 100$  kw thì giá mỗi số điện vượt hạn mức 50 kw là 2500 đ.
- Nếu số điện  $> 100$  kw thì giá mỗi số điện vượt hạn mức 50 kw là 3500 đ.

**Câu 6:** Viết hàm để tổ chức 1 menu để gọi các hàm đã viết ở bài 1 đến bài 5 như mẫu:

```
#####
```

1. Bài 1
2. Bài 2
3. Bài 3
4. Bài 4
5. Bài 5
6. Thoát

```
#####
```

Chọn chức năng: \_

Chọn chức năng từ 1 đến 6 để gọi các hàm đã viết ở bài 1 đến 5

Hướng dẫn:

- Dùng hàm `WriteLine()` để xuất ra màn hình menu như mẫu.
- Dùng hàm `ReadLine()` để nhận ký tự số từ 1 đến 6 nhập từ bàn phím tương đương với 6 chức năng trong menu (chuyển đổi về kiểu `int`).
- Sử dụng lệnh `Switch` để gọi đến các hàm tương ứng đã viết ở bài 1 đến 5.

**V. Bài tập thêm**

**Câu 1:** Viết chương trình tìm ước số chung lớn nhất của hai số nguyên nhập từ bàn phím.

**Câu 2:** Viết chương trình nhập tử và mẫu một phân số, đơn giản phân số và xuất phân số sau khi đơn giản.

**Câu 3:** Viết chương trình nhập vào số nguyên dương  $n$ , liệt kê tất cả các ước số của  $n$ .

**Câu 4:** Viết chương trình chơi game kéo búa bao.

**Câu 5:** Viết chương trình nhập vào số  $n$  và in ra màn hình các số lẻ từ 0 –  $n$ .

**Câu 6:** Viết chương trình nhập vào ngày tháng năm, xuất ra màn hình là ngày thứ mấy (thứ tự) và là tuần thứ mấy của năm đó.

**Câu 7:** Viết chương trình nhập vào năm, tính và xuất ra năm đó có bao nhiêu ngày thứ sáu ngày 13.

**Câu 8:** Viết chương trình nhập vào thông tin của một tam giác (vị trí 03 đỉnh của tam giác); xuất ra Chu vi và diện tích của tam giác.

Hướng dẫn: dùng `Struct` để xác định vị trí  $x, y$  của một đỉnh.

**Câu 9:** Nhập vào một số nguyên dương, xuất ra số chữ số của  $n$ . VD:  $n = 5403$ , KQ  $n$  có 4 chữ số.

**Câu 10:** Nhập vào một số nguyên dương  $n$ , in ra cách đọc số  $n$  này. VD: 213456, hai trăm mười ba ngàn bốn trăm năm mươi sáu.

Lưu ý: Có thể làm thêm các bài tập ở đây: <https://projecteuler.net/archives>