

```
using System;
using System.Collections;
class Program{
    public static void Main(string[] args){
        // 1. Crie o ArrayList AL1 sem definir a capacidade inicial
        ArrayList AL1 = new ArrayList();
        // 2. Imprima a capacidade e a quantidade de elementos.
        Console.WriteLine("Capacidade: {0} Quantidade: {1}",
AL1.Capacity, AL1.Count);
        // 3. Adicione os números 19, 7 e 11.
        AL1.Add(19); AL1.Add(7); AL1.Add(11);
        // 4. Imprima a capacidade e a quantidade de elementos.
        Console.WriteLine("Capacidade: {0} Quantidade: {1}",
AL1.Capacity, AL1.Count);
        // 5. Adicione os números 5, 7 e 17.
        AL1.Add(5); AL1.Add(7); AL1.Add(17);
        // 6. Imprima a capacidade e a quantidade de elementos.
        Console.WriteLine("Capacidade: {0} Quantidade: {1}",
AL1.Capacity, AL1.Count);
        // 7. Imprima os elementos do ArrayList usando o comando foreach.
        foreach(int num in AL1){
            Console.Write(num + " ");
        } Console.WriteLine();
        // 8. Adicione o número 5 nas posições 0, 2 e 10. Ocorreu algum
        problema? Se sim, anote qual foi o motivo e resolva esse problema.
        AL1[0] = 5; AL1[2] = 5; AL1[5] = 5; /* O ultimo elemento
        estava na posição 5 , e não tem como ultrapassar*/
        // 9. Imprima os elementos do ArrayList usando o comando for.
        for(int i = 0; i < AL1.Count; i++){
            Console.Write(AL1[i] + " ");
        } Console.WriteLine();
        // 10. Imprima a primeira e a última posição que contém o número
        7.
        Console.WriteLine("Primeira posição: " + AL1.IndexOf(7));
        Console.WriteLine("Última posição: " + AL1.LastIndexOf(7));
        // 11. Imprima todas as posições que contém o número 5.
        Console.Write("posições: ");
        for(int i = 0; i < AL1.Count; i++){
            if(AL1[i].Equals(5)){
                Console.Write(i + " ");
            }
        }
        // 12. Adicione os números 5, 23, 47, 5 e 5.
        AL1.Add(5); AL1.Add(23); AL1.Add(47); AL1.Add(5); AL1.Add(5);
        Console.WriteLine();
        // 13. Imprima os elementos do ArrayList usando o comando while
        int j = 0;
```

```

        while(AL1.Count > j){
            Console.Write(AL1[j] + " ");
            j++;
        }Console.WriteLine();
        // 14. Imprima as posições do número 5 retornada pelos métodos:
        BinarySearch(), IndexOf() e LastIndexOf()
        Console.Write("Binary: " + AL1.BinarySearch(5) + " / ");
        Console.Write("IndexOf: " + AL1.IndexOf(5) + " / ");
        Console.Write("LastIndexOf: " + AL1.LastIndexOf(5));
        // 15. Ordene os elementos do ArrayList.
        AL1.Sort(); Console.WriteLine();
        // 16. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){
            Console.Write(num + " ");
        }Console.WriteLine();
        // 17. Remova o número 23.
        AL1.Remove(23);
        // 18. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){
            Console.Write(num + " ");
        }Console.WriteLine();
        // 19. Remova o elemento da posição 7.
        AL1.RemoveAt(7);
        // 20. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){
            Console.Write(num + " ");
        }Console.WriteLine();
        // 21. Remova os elementos das posições 2, 3 e 4.
        AL1.RemoveRange(2,3);
        // 22. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){
            Console.Write(num + " ");
        }Console.WriteLine();
        // 23. Inverta os elementos do ArrayList.
        AL1.Reverse();
        // 24. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){
            Console.Write(num + " ");
        }Console.WriteLine();
        // 25. Remova todos os elementos do ArrayList.
        AL1.Clear();
        // 26. Imprima os elementos do ArrayList (use o comando que você
        desejar).
        foreach(int num in AL1){

```

```

        Console.Write(num + " ");
    }
    // 27. Imprima a capacidade e a quantidade de elemento
    Console.WriteLine("Capacidade: {0} Quantidade: {1}",
AL1.Capacity, AL1.Count);
    }
}

```

1 a 17 – ArrayList // Lista

```

using System;
using System.Collections;
class Program{
    public static void Main(string[] args){

        // 1 - Crie um ArrayList e adicione 10 valores inteiros digitados
        pelo usuário. Ao final, imprima todos os elementos.
        int respInt;
        ArrayList AL = new ArrayList();
        for(int i = 0; i < 10; i++){
            respInt = int.Parse(Console.ReadLine());
            AL.Add(respInt);
        }
        for(int i = 0; i < AL.Count; i++){
            Console.Write(AL[i] + " ");
        }Console.WriteLine();

        // 2 - Crie um ArrayList e adicione 10 strings digitados pelo
        usuário. Ao final, imprima todos os elementos
        LimpaArray(AL);
        string respString;
        for(int i = 0; i < 10; i++){
            respString = Console.ReadLine();
            AL.Add(respString);
        }
        for(int i = 0; i < AL.Count; i++){
            Console.Write(AL[i] + " ");
        }Console.WriteLine();

        // 3 - Crie um ArrayList e adicione os números ímpares no
        intervalo entre 1 a 100. Calcule a soma dos números usando o comando for.
        LimpaArray(AL);
        int respFor = 0, somaFor = 0 ;
        for(int i = 0; i < 100; i++){
            respFor += 1 ;
            if(respFor % 2 != 0){
                AL.Add(respFor);
                somaFor += respFor;
            }
        }
    }
}

```

```

    }
}
Console.WriteLine("Soma ímpares com for: " + somaFor);

// 4 - Crie um ArrayList e adicione os números pares no intervalo
entre 1 a 100. Calcule a soma dos números usando o comando while.
LimpaArray(AL);
int respWhile = 0, somaWhile = 0 ;
while(respWhile <= 100){
    respWhile += 1 ;
    if(respWhile % 2 == 0){
        AL.Add(respWhile);
        somaWhile += respWhile;
    }
}
Console.WriteLine("Soma pares com While: " + somaWhile);

// 5 - Crie um ArrayList e adicione os números no intervalo entre
1 a 100. Calcule a soma dos números usando o comando do while.
LimpaArray(AL);
int respDo = 0, somaDo = 0 ;
do{
    respDo += 1 ;
    AL.Add(respDo);
    somaDo += respDo;
}while(respDo < 100);
Console.WriteLine("Soma numeros com DoWhile: " + somaDo);

// 6 - Crie um ArrayList e adicione os números pares no intervalo
entre 1 a 100. Calcule a soma dos números usando o comando foreach.
LimpaArray(AL);
int somaForeach = 0;
for (int i = 1; i <= 100; i++){
    if(i % 2 == 0){
        AL.Add(i);
    }
}
foreach(int numero in AL){
    somaForeach += numero;
}
Console.WriteLine("Soma pares com foreach: " + somaForeach);

// 7 - Crie um ArrayList contendo os seguintes elementos (5, 13,
19, 31, 3, 7, 11, 5, 57, 13, 5). Faça uma função que apague TODAS as
ocorrências de um determinado elemento. Use essa função para apagar todas
as ocorrências do número 5 e 13.
LimpaArray(AL);

```

```

AL = new ArrayList(){5, 13, 19, 31, 3, 7, 11, 5, 57, 13, 5};
int num = 5;
ApagaNumInArrayList(num,AL);
int num2 = 13;
ApagaNumInArrayList(num2,AL);
Console.Write("ArrayList: ");
for (int i = 0; i < AL.Count; i++)
{
    Console.Write(AL[i] + " ");
}
Console.WriteLine();

```

// 8 - Faça um programa que leia n números inteiros e os armazene em um ArrayList. Calcule a soma e a média aritmética (use o comando FOR e depois o FOREACH).

```

LimpaArray(AL);
double qnt = 0, soma = 0, mediaAritmética = 0;
string n = Console.ReadLine();
while(n != "FIM"){
    AL.Add(int.Parse(n));
    qnt++;
    n = Console.ReadLine();
}
foreach(int numero in AL){
    soma += numero;
}
mediaAritmética = soma/qnt;
Console.WriteLine("A soma dos elementos é: {0} e a média
Aritmética é: {1}" , soma , mediaAritmética);

```

// 11 - Faça um programa que preencha um ArrayList com elementos de diferentes tipos (int, double, float, boolean, String). Tente calcular a soma dos elementos. Evidentemente, isso irá provocar uma mensagem de erro. Que mensagem o Visual Studio retorna?

```

// AL = new ArrayList(){ 1, 2, "AED", new Queue(), "teste", 3.14
};

// foreach (var item in AL){
//     soma += item;
// }

// O operador "+" não pode ser aplicado a operandos dos tipos
"double" e "object" [ArrayList]

```

// 12 - Faça um programa que preencha um ArrayList com os números entre 1 e 25. Pedese:

```

// • Imprima todos os elementos
// • Imprima todos os elementos em ordem invertida
// • Imprima todos os elementos em posições ímpares (os elementos
da posição 1, 3, 5, ...)
// • Imprima todos os elementos ímpares

```

```

    // • Imprima apenas os elementos da primeira metade do vetor
    (posição 0 a 12).
    // OBS: você deve fazer esse programa 2 vezes. Primeiro usando o
    comando FOR e depois usando o comando FOREACH.
    LimpaArray(AL);
    AL = new ArrayList(){1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25};
    int metade = AL.Count/2;
    // For
    Console.WriteLine("-For-");
    Console.Write("Elementos: ");
    for(int i = 0; i < AL.Count; i++){
        Console.Write(AL[i] + " ");
    }
    Console.WriteLine();
    Console.Write("Elementos invertidos: ");
    AL.Reverse();
    for(int i = 0; i < AL.Count; i++){
        Console.Write(AL[i] + " ");
    }
    AL.Reverse();
    Console.WriteLine();
    Console.Write("Elementos posições ímpares: ");
    for(int i = 0; i < AL.Count; i++){
        if(i % 2 != 0){
            Console.Write(AL[i] + " ");
        }
    }
    Console.WriteLine();
    Console.Write("Elementos ímpares: ");
    for(int i = 0; i < AL.Count; i++){
        if((int)AL[i] % 2 != 0){
            Console.Write(AL[i] + " ");
        }
    }
    Console.WriteLine();
    Console.Write("Elementos primeira metade: ");
    for(int i = 0; i <= metade; i++){
        Console.Write(AL[i] + " ");
    }
    Console.WriteLine();
    // foreach
    Console.WriteLine("-Foreach-");
    Console.Write("Elementos: ");
    foreach (var item in AL)
    {
        Console.Write(item + " ");
    }
    Console.WriteLine();

```

```

Console.Write("Elementos invertidos: ");
AL.Reverse();
foreach (var item in AL)
{
    Console.Write(item + " ");
}
AL.Reverse();
Console.WriteLine();
Console.Write("Elementos posições ímpares: ");
int per = 0;
foreach (var item in AL)
{
    if(per % 2 != 0){
        Console.Write(item + " ");
    }
    per++;
}
Console.WriteLine();
Console.Write("Elementos ímpares: ");
foreach (var item in AL)
{
    if((int)item % 2 != 0){
        Console.Write(item + " ");
    }
}
Console.WriteLine();
Console.Write("Elementos primeira metade: ");
int cont = 0;
foreach (var item in AL)
{
    if(cont <= metade){
        Console.Write(item + " ");
    }
    cont++;
}

```

// 13 - Faça um programa que gere uma coleção com n números inteiros aleatórios (o valor de n deve ser informado pelo usuário no início da execução do programa. Imprima os elementos da coleção.

```

int quantidade = int.Parse(Console.ReadLine());
Random R = new Random();
for(int i = 0; i < quantidade; i++){
    AL.Add(R.Next(0,500));
}
Console.Write("ArrayList: ");
for (int i = 0; i < AL.Count; i++)
{
    Console.Write(AL[i] + " ");
}

```

```

        Console.WriteLine();

    }

    public static void LimpaArray(ArrayList AL){
        AL.Clear();
    }

    public static ArrayList ApagaNumInArrayList(int num, ArrayList AL){
        for(int i = 0; i < AL.Count; i++){
            if(AL.Contains(num)){
                AL.Remove(num);
            }
        }
        return AL;
    }

    // 14 - Crie uma função para inverter os dados da coleção recebida
    como parâmetro. Obs1: use qualquer outra estrutura que julgar necessária.
    Obs2: não utilize o método reverse da classe ArrayList.
    public static void InverteArrayList(ArrayList collection){
        int i = 0, j = collection.Count-1;
        while(i < j){
            object temp = collection[i];
            collection[i] = collection[j];
            collection[j] = temp;
            i++;
            j--;
        }
    }

    // 15 - Crie uma função que receba a coleção como parâmetro e retorne
    a soma de seus elementos. Obs: considere que todos seus dados são do tipo
    int.
    public static int SomaElementos(ArrayList collection){
        int soma = 0;
        foreach(int numero in collection){
            soma += numero;
        }
        return soma;
    }

    // 16 - Crie uma função que calcule o número de elementos positivos
    de uma coleção passada como parâmetro.
    public static int CalculaPositivos(ArrayList collection){
        int positivos = 0;
        foreach(int numero in collection){
            if(numero > 0){
                positivos++;
            }
        }
        return positivos;
    }
}

```



```

// 17 - Crie uma função que calcule o número de ocorrências em uma
coleção de um elemento passado como parâmetro.
public static int ContaOcorrencias(ArrayList collection, int busca){
    int ocorrencias = 0, i = 0;
    foreach(int numero in collection){
        if((int)collection[i] == busca){
            ocorrencias++;
        }
        i++;
    }
    return ocorrencias;
}
}

```

1 a 17 – Queue // Fila

```

using System;
using System.Collections;
class Program{
    public static void Main(string[] args){
        // 1 - Crie um ArrayList e adicione 10 valores inteiros digitados
        pelo usuário. Ao final, imprima todos os elementos.
        Queue Q = new Queue();
        int respInt;
        for(int i = 0; i < 10; i++){
            respInt = int.Parse(Console.ReadLine());
            Q.Enqueue(respInt);
        }
        for(int i = 0; i < 10; i++){
            Console.Write(Q.Dequeue() + " ");
        }Console.WriteLine();

        // 2 - Crie um ArrayList e adicione 10 strings digitados pelo
        usuário. Ao final, imprima todos os elementos
        LimpaFila(Q);
        string respString;
        for(int i = 0; i < 10; i++){
            respString = Console.ReadLine();
            Q.Enqueue(respString);
        }
        for(int i = 0; i < 10; i++){
            Console.Write(Q.Dequeue() + " ");
        }Console.WriteLine();

        // 3 - Crie um ArrayList e adicione os números ímpares no
        intervalo entre 1 a 100. Calcule a soma dos números usando o comando for.
        LimpaFila(Q);
        int somaFor = 0, respFor = 0;
    }
}

```

```

for (int i = 0; i < 100; i++)
{
    respFor += 1;
    if(respFor % 2 != 0){
        Q.Enqueue(respFor);
        somaFor += respFor;
    }
}
Console.WriteLine("Soma ímpares com for: " + somaFor);

// 4 - Crie um ArrayList e adicione os números pares no intervalo
entre 1 a 100. Calcule a soma dos números usando o comando while.
LimpaFila(Q);
int respWhile = 0, somaWhile = 0;
while(respWhile <= 100){
    respWhile += 1 ;
    if(respWhile % 2 == 0){
        Q.Enqueue(respWhile);
        somaWhile += respWhile;
    }
}
Console.WriteLine("Soma pares com While: " + somaWhile);

// 5 - Crie um ArrayList e adicione os números no intervalo entre
1 a 100. Calcule a soma dos números usando o comando do while.
LimpaFila(Q);
int respDo = 0, somaDo = 0;
do{
    respDo += 1 ;
    Q.Enqueue(respDo);
    somaDo += respDo;
}while(respDo < 100);
Console.WriteLine("Soma numeros com DoWhile: " + somaDo);

// 6 - Crie um ArrayList e adicione os números pares no intervalo
entre 1 a 100. Calcule a soma dos números usando o comando foreach.
LimpaFila(Q);
int somaForeach = 0;
for (int i = 1; i <= 100; i++){
    if(i % 2 == 0){
        Q.Enqueue(i);
    }
}
foreach(int numero in Q){
    somaForeach += numero;
}
Console.WriteLine("Soma pares com foreach: " + somaForeach);

```

// 7 - Crie um ArrayList contendo os seguintes elementos (5, 13, 19, 31, 3, 7, 11, 5, 57, 13, 5). Faça uma função que apague TODAS as ocorrências de um determinado elemento. Use essa função para apagar todas as ocorrências do número 5 e 13.

```
Q = new Queue();
int num = 5, num2 = 13;
Q.Enqueue(5); Q.Enqueue(13); Q.Enqueue(19); Q.Enqueue(31);
Q.Enqueue(3); Q.Enqueue(7); Q.Enqueue(11); Q.Enqueue(5); Q.Enqueue(57);
Q.Enqueue(13); Q.Enqueue(5);
ApagaNumInQueue(num, ref Q);
ApagaNumInQueue(num2, ref Q);
Console.Write("Queue: ");
foreach (var item in Q)
{
    Console.Write(item + " ");
}
Console.WriteLine();
```

// 9 - Faça um programa que leia n números inteiros e os armazene em um Queue. Calcule a soma e a média aritmética (use o comando FOREACH para iterar sobre os elementos).

```
LimpaFila(Q);
double qnt = 0, soma = 0, mediaAritmética = 0;
string n = Console.ReadLine();
while(n != "FIM"){
    Q.Enqueue(int.Parse(n));
    qnt++;
    n = Console.ReadLine();
}
foreach(int numero in Q){
    soma += numero;
}
mediaAritmética = soma/qnt;
Console.WriteLine("A soma dos elementos é: {0} e a média
Aritmética é: {1}" , soma , mediaAritmética);
```

// 11 - Faça um programa que preencha um ArrayList com elementos de diferentes tipos (int, double, float, boolean, String). Tente calcular a soma dos elementos. Evidentemente, isso irá provocar uma mensagem de erro. Que mensagem o Visual Studio retorna?

```
// AL = new ArrayList(){ 1, 2, "AED", new Queue(), "teste", 3.14
};

// foreach (var item in AL){
//     soma += item;
// }

// O operador "+" não pode ser aplicado a operandos dos tipos
"double" e "object" [ArrayList]
```

```

// 12 - Faça um programa que preencha um ArrayList com os números
entre 1 e 25. Pede-se:
// • Imprima todos os elementos
// • Imprima todos os elementos em ordem invertida
// • Imprima todos os elementos em posições ímpares (os elementos
da posição 1, 3, 5, ...)
// • Imprima todos os elementos ímpares
// • Imprima apenas os elementos da primeira metade do vetor
(posição 0 a 12).
// OBS: você deve fazer esse programa 2 vezes. Primeiro usando o
comando FOR e depois usando o comando FOREACH.
LimpaFila(Q);
int preencher = 1;
for(int i = 0; i < 25; i++){
    Q.Enqueue(preencher);
    preencher++;
}
int metade = 25/2;
Console.WriteLine("-Foreach-");
Console.Write("Elementos: ");
foreach (var item in Q)
{
    Console.Write(item + " ");
}
Console.WriteLine();
Console.Write("Elementos invertidos: ");
int[] elementos = Q.Cast<int>().ToArray();
Array.Reverse(elementos);
foreach (int numero in elementos) {
    Console.Write(numero + " ");
}
Console.WriteLine();
Console.Write("Elementos posições ímpares: ");
int per = 0;
foreach (var item in Q)
{
    if(per % 2 != 0){
        Console.Write(item + " ");
    }
    per++;
}
Console.WriteLine();
Console.Write("Elementos ímpares: ");
foreach (var item in Q)
{
    if((int)item % 2 != 0){
        Console.Write(item + " ");
    }
}
}

```

```

Console.WriteLine();
Console.Write("Elementos primeira metade: ");
int cont = 0;
foreach (var item in Q)
{
    if(cont <= metade){
        Console.Write(item + " ");
    }
    cont++;
} Console.WriteLine();

```

// 13 - Faça um programa que gere uma coleção com n números inteiros aleatórios (o valor de n deve ser informado pelo usuário no início da execução do programa. Imprima os elementos da coleção

```

LimpaFila(Q);
int quantidade = int.Parse(Console.ReadLine());
Random R = new Random();
for(int i = 0; i < quantidade; i++){
    Q.Enqueue(R.Next());
}
Console.Write("ArrayList: ");
foreach (var item in Q)
{
    Console.Write(item + " ");
}
Console.WriteLine();
}
public static void LimpaFila(Queue Q){
    Q.Clear();
}
public static void ApagaNumInQueue(int num, ref Queue Q){
    Queue temp = new Queue();
    foreach (var item in Q)
    {
        if(num != (int)item){
            temp.Enqueue(item);
        }
    }
    Q = temp;
}

```

// 14 - Crie uma função para inverter os dados da coleção recebida como parâmetro. Obs1: use qualquer outra estrutura que julgar necessária. Obs2: não utilize o método reverse da classe ArrayList.

```

public static void InverteFila(ref Queue collection){
    int[] elementos = collection.Cast<int>().ToArray();
    Array.Reverse(elementos);
    foreach (int numero in elementos) {
        Console.Write(numero + " ");
    }
}

```

```

    }
}

// 15 - Crie uma função que receba a coleção como parâmetro e retorne
a soma de seus elementos. Obs: considere que todos seus dados são do tipo
int.
public static int SomaElementos(ref Queue collection){
    int soma = 0;
    foreach(int num in collection){
        soma += num;
    }
    return soma;
}

// 16 - Crie uma função que calcule o número de elementos positivos
de uma coleção passada como parâmetro.
public static int CalculaPositivos(ref Queue collection){
    int positivos = 0;
    foreach(int numero in collection){
        Console.WriteLine(numero);
        if(numero > 0){
            positivos++;
        }
    }
    return positivos;
}

// 17 - Crie uma função que calcule o número de ocorrências em uma
coleção de um elemento passado como parâmetro.
public static int ContaOcorrencias(ref Queue collection, int busca){
    int ocorrencias = 0;
    foreach(int num in collection){
        if((int)collection.Dequeue() == busca){
            ocorrencias++;
        }
    }
    return ocorrencias;
}
}

```

1 a 17 – Stack // Pilha

```

using System;
using System.Collections;
class Program{
    public static void Main(string[] args){
        // 1 - Crie um ArrayList e adicione 10 valores inteiros digitados
        pelo usuário. Ao final, imprima todos os elementos.
        int respInt;
        Stack S = new Stack();
        for (int i = 0; i < 10; i++)

```

```

    {
        respInt = int.Parse(Console.ReadLine());
        S.Push(respInt);
    }
    for (int i = 0; i < 10; i++)
    {
        Console.Write(S.Pop() + " ");
    } Console.WriteLine();

    // 2 - Crie um ArrayList e adicione 10 strings digitados pelo
    usuário. Ao final, imprima todos os elementos
    LimpaPilha(S);
    string respString;
    for (int i = 0; i < 10; i++)
    {
        respString = Console.ReadLine();
        S.Push(respString);
    }
    for (int i = 0; i < 10; i++)
    {
        Console.Write(S.Pop() + " ");
    } Console.WriteLine();

    // 3 - Crie um ArrayList e adicione os números ímpares no
    intervalo entre 1 a 100. Calcule a soma dos números usando o comando for.
    LimpaPilha(S);
    int respFor = 0, somaFor = 0 ;
    for (int i = 0; i < 100; i++)
    {
        respFor += 1;
        if(respFor % 2 != 0){
            S.Push(respFor);
            somaFor += respFor;
        }
    }
    Console.WriteLine("Soma ímpares com for: " + somaFor);

    // 4 - Crie um ArrayList e adicione os números pares no intervalo
    entre 1 a 100. Calcule a soma dos números usando o comando while.
    LimpaPilha(S);
    int respWhile = 0, somaWhile = 0 ;
    while(respWhile <= 100){
        respWhile += 1 ;
        if(respWhile % 2 == 0){
            S.Push(respWhile);
            somaWhile += respWhile;
        }
    }
    Console.WriteLine("Soma pares com While: " + somaWhile);

```

// 5 - Crie um ArrayList e adicione os números no intervalo entre 1 a 100. Calcule a soma dos números usando o comando do while.

```
LimpaPilha(S);
int respDo = 0, somaDo = 0 ;
do{
    respDo += 1 ;
    S.Push(respDo);
    somaDo += respDo;
}while(respDo < 100);
Console.WriteLine("Soma numeros com DoWhile: " + somaDo);
```

// 6 - Crie um ArrayList e adicione os números pares no intervalo entre 1 a 100. Calcule a soma dos números usando o comando foreach.

```
LimpaPilha(S);
int somaForeach = 0;
for (int i = 1; i <= 100; i++){
    if(i % 2 == 0){
        S.Push(i);
    }
}
foreach(int numero in S){
    somaForeach += numero;
}
Console.WriteLine("Soma pares com foreach: " + somaForeach);
```

// 7 - Crie um ArrayList contendo os seguintes elementos (5, 13, 19, 31, 3, 7, 11, 5, 57, 13, 5). Faça uma função que apague TODAS as ocorrências de um determinado elemento. Use essa função para apagar todas as ocorrências do número 5 e 13.

```
S = new Stack();
int num = 5, num2 = 13;
S.Push(5); S.Push(13); S.Push(19); S.Push(31); S.Push(3);
S.Push(7); S.Push(11); S.Push(5); S.Push(57); S.Push(13); S.Push(5);
ApagaNumInStack(num, ref S);
ApagaNumInStack(num2, ref S);
Console.Write("Stack: ");
foreach (var item in S)
{
    Console.Write(item + " ");
}
Console.WriteLine();
```

// 10 - Faça um programa que leia n números inteiros e os armazene em um Stack. Calcule a soma e a média aritmética (use o comando FOREACH para iterar sobre os elementos).

```
LimpaPilha(S);
double qnt = 0, soma = 0, mediaAritmética = 0;
string n = Console.ReadLine();
```



```

        while(n != "FIM"){
            S.Push(int.Parse(n));
            qnt++;
            n = Console.ReadLine();
        }
        foreach(int numero in S){
            soma += numero;
        }
        mediaAritmética = soma/qnt;
        Console.WriteLine("A soma dos elementos é: {0} e a média
Aritmética é: {1}" , soma , mediaAritmética);

// 11 - Faça um programa que preencha um ArrayList com elementos
de diferentes tipos (int, double, float, boolean, String). Tente calcular
a soma dos elementos. Evidentemente, isso irá provocar uma mensagem de
erro. Que mensagem o Visual Studio retorna?
// AL = new ArrayList(){ 1, 2, "AED", new Queue(), "teste", 3.14
};

// foreach (var item in AL){
//     soma += item;
// }
// O operador "+=" não pode ser aplicado a operandos dos tipos
"double" e "object" [ArrayList]

// 12 - Faça um programa que preencha um ArrayList com os números
entre 1 e 25. Pede-se:
// • Imprima todos os elementos
// • Imprima todos os elementos em ordem invertida
// • Imprima todos os elementos em posições ímpares (os elementos
da posição 1, 3, 5, ...)
// • Imprima todos os elementos ímpares
// • Imprima apenas os elementos da primeira metade do vetor
(posição 0 a 12).
// OBS: você deve fazer esse programa 2 vezes. Primeiro usando o
comando FOR e depois usando o comando FOREACH.
LimpaPilha(S);
int preencher = 1;
for(int i = 0; i < 25; i++){
    S.Push(preencher);
    preencher++;
}
int metade = 25/2;
Console.WriteLine("-Foreach-");
Console.Write("Elementos: ");
foreach (var item in S)
{
    Console.Write(item + " ");
}
Console.WriteLine();

```

```

Console.Write("Elementos invertidos: ");
int[] elementos = S.Cast<int>().ToArray();
Array.Reverse(elementos);
foreach (int numero in elementos) {
    Console.Write(numero + " ");
}
Console.WriteLine();
Console.Write("Elementos posições ímpares: ");
int per = 0;
foreach (var item in S)
{
    if(per % 2 != 0){
        Console.Write(item + " ");
    }
    per++;
}
Console.WriteLine();
Console.Write("Elementos ímpares: ");
foreach (var item in S)
{
    if((int)item % 2 != 0){
        Console.Write(item + " ");
    }
}
Console.WriteLine();
Console.Write("Elementos primeira metade: ");
int cont = 0;
foreach (var item in S)
{
    if(cont <= metade){
        Console.Write(item + " ");
    }
    cont++;
} Console.WriteLine();

```

// 13 - Faça um programa que gere uma coleção com n números inteiros aleatórios (o valor de n deve ser informado pelo usuário no início da execução do programa. Imprima os elementos da coleção

```

LimpaPilha(S);
int quantidade = int.Parse(Console.ReadLine());
Random R = new Random();
for(int i = 0; i < quantidade; i++){
    S.Push(R.Next());
}
Console.Write("ArrayList: ");
foreach (var item in S)
{
    Console.Write(item + " ");
}

```

```

        Console.WriteLine();
    }
    public static void LimpaPilha(Stack S){
        S.Clear();
    }
    public static void ApagaNumInStack(int num, ref Stack S){
        Stack temp = new Stack();
        foreach (var item in S)
        {
            if(num != (int)item){
                temp.Push(item);
            }
        }
        S = temp;
    }
    // 14 - Crie uma função para inverter os dados da coleção recebida
    como parâmetro. Obs1: use qualquer outra estrutura que julgar necessária.
    Obs2: não utilize o método reverse da classe ArrayList.
    public static void InvertePilha(ref Stack collection){
        int[] elementos = collection.Cast<int>().ToArray();
        Array.Reverse(elementos);
        foreach (int numero in elementos) {
            Console.Write(numero + " ");
        }
    }
    // 15 - Crie uma função que receba a coleção como parâmetro e retorne
    a soma de seus elementos. Obs: considere que todos seus dados são do tipo
    int.
    public static int SomaElementos(ref Stack collection){
        int soma = 0;
        foreach(int num in collection){
            soma += num;
        }
        return soma;
    }
    // 16 - Crie uma função que calcule o número de elementos positivos
    de uma coleção passada como parâmetro.
    public static int CalculaPositivos(ref Stack collection){
        int positivos = 0;
        foreach(int numero in collection){
            Console.WriteLine(numero);
            if(numero > 0){
                positivos++;
            }
        }
        return positivos;
    }
}

```

```
// 17 - Crie uma função que calcule o número de ocorrências em uma
coleção de um elemento passado como parâmetro.
public static int ContaOcorrencias(ref Stack collection, int busca){
    int ocorrencias = 0;
    foreach(int num in collection){
        if((int)collection.Pop() == busca){
            ocorrencias++;
        }
    }
    return ocorrencias;
}
}
```