

Exercicios 15

```
programa {
    // 15.2 Escreva um algoritmo que leia um número inteiro positivo, n, e uma seq
    // inteiros, determine e escreva a quantidade de segmentos de números iguais c
    // compõem essa sequência. Por exemplo, para n=9 e a sequência 5, 2, 2, 4, 4, 4
    // então a saída deve ser o número 4, pois
    // 5, 2, 2, 4, 4, 4, 1, 1
    // possui quatro segmentos de números iguais consecutivos:
    funcao inicio() {
        inteiro a, i = 0, vetor[10], segmentos = 1

        escreva("Escreva um número inteiro: ")
        leia(a)

        enquanto (a < 0) {
            escreva("Erro: Digite um valor positivo: ")
            leia(a)
        }

        escreva("Escreva uma sequencia separado por ,")
        leia(vetor)

        para(inteiro i = 0; i < a; i++) {
            se (vetor[i] != vetor[i-1]) {
                segmentos = segmentos+1
            }
        }

        escreva("numero de Segmentos: ", segmentos)
    }
}
```

```

programa {
    // 15.5 Escreva um algoritmo que leia um número inteiro positivo,  $n > 1$ , e dete
    // sua decomposição em fatores primos, exibindo a multiplicidade de cada fator.
    // se a entrada for  $n = 18$ , então a saída deve ser da forma
    // 2 com multiplicidade 1
    // 3 com multiplicidade 2
    // pois  $2 \cdot 3 \cdot 3 = 18$  e 2 e 3 são números primos.
    funcao inicio() {
        inteiro n, multi = 0

        escreva("Escreva um número inteiro: ")
        leia(n)

        enquanto (n < 0) {
            escreva("Erro: Digite um valor positivo: ")
            leia(n)
        }

        para(inteiro fator = 2; fator <= n; fator++) {
            enquanto (n % fator == 0) {
                multi = multi + 1
                n = n / fator
            }

            se (multi > 0) {
                escreva(fator, " com multiplicidade ", multi)
            }
        }
    }
}

```

```

programa {
// 15.4 Escreva um algoritmo que leia um número inteiro positivo,  $n > 1$ , e dete
// sua decomposição em fatores primos, exibindo a multiplicidade de cada fator.
// se a entrada for  $n = 18$ , então a saída deve ser da forma
// 2 com multiplicidade 1
// 3 com multiplicidade 2
// pois  $2 \cdot 3 \cdot 3 = 18$  e 2 e 3 são números primos.
funcao inicio() {
    inteiro n, multi = 0

    escreva("Escreva um número inteiro: ")
    leia(n)

    enquanto (n < 0) {
        escreva("Erro: Digite um valor positivo: ")
        leia(n)
    }

    para(inteiro fator = 2; fator <= n; fator++) {
        enquanto (n % fator == 0) {
            multi = multi +1
            n = n / fator
        }

        se (multi > 0) {
            escreva(fator, " com multiplicidade ", multi)
        }
    }
}
}

```

```

programa {
    // 15.3 Escreva um algoritmo que leia um número inteiro positivo,  $n > 1$ , e dete
    // sua decomposição em fatores primos, exibindo a multiplicidade de cada fator.
    // se a entrada for  $n = 18$ , então a saída deve ser da forma
    // 2 com multiplicidade 1
    // 3 com multiplicidade 2
    // pois  $2 \cdot 3 \cdot 3 = 18$  e 2 e 3 são números primos.
    funcao inicio() {
        inteiro n, multi = 0

        escreva("Escreva um número inteiro: ")
        leia(n)

        enquanto (n < 0) {
            escreva("Erro: Digite um valor positivo: ")
            leia(n)
        }

        para(inteiro fator = 2; fator <= n; fator++) {
            enquanto (n % fator == 0) {
                multi = multi + 1
                n = n / fator
            }

            se (multi > 0) {
                escreva(fator, " com multiplicidade ", multi)
            }
        }
    }
}

```

```

programa {
// 15.6 Escreva um algoritmo que leia dois números inteiros positivos, n e m, e
// ordenadas, em ordem não-decrescente, com n e m números inteiros, respectivamente
// seguida, o algoritmo deve criar e escrever como saída uma única sequência ordenada
// que n, m ≤ 100.
funcao inicio() {
    inteiro n, m, vet_1[100], vet_2[100], indice1, indice2, vet_mesc[200]

    escreva("Escreva um número inteiro: ")
    leia(n)

    enquanto (n < 0) {
        escreva("Erro: Digite um valor positivo: ")
        leia(n)
    }

    escreva("Escreva outro número inteiro: ")
    leia(m)

    enquanto (m < 0) {
        escreva("Erro: Digite um valor positivo: ")
        leia(m)
    }

    para(inteiro i = 0; i < n + m; i++) {
        se (indice1 == n) {
            vet_mesc = vet_2 + indice2
            indice2 = indice2 + 1
        } senao se (indice2 == m) {
            vet_mesc = vet_1 + indice1
            indice1 = indice1 + 1
        } senao {
            se (vet_1[indice1] <= vet_2[indice2]) {
                vet_mesc = vet_1 + indice1
                indice1 = indice1 + 1
            } senao {
                vet_mesc = vet_2 + indice2
                indice2 = indice2 + 1
            }
        }
    }

    escreva(vet_mesc)
}
}

```