



Atividade 02 - Assembly

- a.) $a = 10$
 $b = 15;$
 $c = 35;$
 $d = 25;$
 $a = (a + b) - (c - d);$
- b) $a = 10$
 $b = 10;$
 $c = 14;$
 $d = 12;$
 $a = (a - b) - (c - d);$
- c.)
for (int i = 0; i < 100; i++)
 System.out.println(i * 2);
- d.)
for (int i = 0; i < 100; i++)
 System.out.println(i + 4);
- e.) $a = 18; b = 18; d = 15; e = 40; g = 24; h = 12; k = 55; l = 12;$
if (a == b) {
 $c = d + e ;$
} else {
 $f = g - h ;$
}
 $j = k * l ;$
- f.) $a = 18; b = 20;$
if (a != b) {
 $c = 2 + b ;$
} else {
 $d = 3 + a ;$
}
}
- g.) int maior (int x, int y) {
 if (x >= y)
 return x;
 else return y;
}
- h.) int menor (int x, int y) {
 if (x < y)
 return x;
 else return y;
}



Atividade 02 - Assembly

i.)

```
int maximo (int a[], int n) {  
    int i, max;  
    max = a[0];  
    for (i = 1; i < n; i++)  
        if (a[i] > max)  
            max = a[i];  
    return max;  
}
```

j.)

```
int minimo (int a[], int n) {  
    int i, min;  
    min = a[0];  
    for (i = 1; i < n; i++)  
        if (a[i] < min)  
            min = a[i];  
    return min;  
}
```

k.)

```
int pertence(int x, int L[], int n) {  
    int i;  
    for(i=0; i<n; i++)  
        if( x == L[i] )  
            return 1;  
    return 0;  
}
```

l.)

```
int fibonaccirec(int n) {  
    if (n == 0)  
        return 0;  
    if (n == 1)  
        return 1;  
    return fibonaccirec(n - 1) + fibonaccirec(n - 2);  
}
```

m.)

```
int mdc_recurativa(int a, int b) {  
    if (a % b == 0)  
        return b;  
    return mdc_recurativa (b, a % b);  
}
```