

14. Desenvolver um programa iterativo e um programa recursivo, equivalente ao programa monolítico apresentado a seguir.

Programa Monolítico – Instruções Rotuladas

```
R1: Se T1 então vá_para R2 senão vá_para R3;  
R2: Faça F1 vá_para R3;  
R3: Se T2 então vá_para R6 senão vá_para R4;  
R4: Faça F2 vá_para R5;  
R5: Faça F3 vá_para R1;
```

Programa Iterativo

```
programa  
  se T1 então  
    F1;  
  fim se  
  até T2 faça  
    F2;  
    F3;  
    se T1 então  
      F1;  
    fim se  
  fim até  
fim programa
```

Programa Recursivo

```
função principal()  
  loop();  
fim função  
  
função loop()  
  se T1 então  
    F1;  
  fim se  
  se T2 então  
    retornar;  
  senão  
    F2;  
    F3;  
    loop1();  
  fim se  
fim função
```

```
#include <stdio.h>

int a = 0;

int b = 0;

void init() { a = 3; b = 5; }

int T1() { return (a > 0) ? a-- : 0; }

int T2() { return (b-- > 0) ? 0 : 1; }

void F1() { printf("F1 "); }

void F2() { printf("F2 "); }

void F3() { printf("F3 "); }

void monolitico() {
    R1: if (T1()) goto R2; else goto R3;
    R2: F1();
    R3: if (T2()) goto R6; else goto R4;
    R4: F2();
    R5: F3(); goto R1;
    R6:
}

void iterativo() {
    if (T1()) {
        F1();
    }
    // O C não possui a instrução até faça
    while (!T2()) {
        F2();
        F3();
        if (T1()) {
            F1();
        }
    }
}

void recursivo() {
    if (T1()) {
        F1();
    }

    if (T2()) {
        return;
    }
    else {
        F2();
        F3();
        recursivo();
    }
}
```

```
int main() {  
    init();  
    printf("Monolítico ");  
    monolitico();  
  
    init();  
    printf("\nIterativo ");  
    iterativo();  
  
    init();  
    printf("\nRecursivo ");  
    recursivo();  
  
    return 0;  
}
```

Monolítico	F1	F2	F3	F1	F2	F3	F1	F2	F3	F2	F3	F2	F3
Iterativo	F1	F2	F3	F1	F2	F3	F1	F2	F3	F2	F3	F2	F3
Recursivo	F1	F2	F3	F1	F2	F3	F1	F2	F3	F2	F3	F2	F3