13. Desenvolver um programa monolítico e um programa recursivo, equivalente ao programa recursivo apresentado a seguir, apresentado na linguagem de programação C.

```
#include <stdio.h>

int sub1(int a);
int sub2(int a);

int sub1(int a)
{
  printf("sub1 - %d\n", a);
  if (a > 0)
    return sub2(a - 1);
  else
    return 0;
}

int sub2(int a)
{
  printf("sub2 - %d\n", a);
  if (a > 0 && a % 2 == 0)
    return sub2(a - 1);
  else
    return sub1(a);
}

int main()
{
  printf("main - %d", sub1(10));
  return 0;
}
```

```
sub1 - 10
sub2 - 9
sub1 - 9
sub2 - 8
sub2 - 7
sub1 - 7
sub2 - 6
sub2 - 5
sub1 - 5
sub2 - 4
sub2 - 3
sub1 - 3
sub2 - 2
sub2 - 1
sub1 - 1
sub2 - 0
sub1 - 0
main - 0
```

```
#include <stdio.h>

int main() {

  int a = 10;

  printf("sub1 - %d\n", a);

  while (a > 0) {

    a = a - 1;

    printf("sub2 - %d\n", a);

    while (a > 0 && a % 2 == 0) {

      a = a - 1;

      printf("sub2 - %d\n", a);
    }

    printf("sub1 - %d\n", a);
  }

  printf("main - %d", a);

  return 0;
}
```

```
sub1 - 10
sub2 - 9
sub1 - 9
sub2 - 8
sub2 - 7
sub1 - 7
sub2 - 6
sub2 - 5
sub1 - 5
sub2 - 4
sub2 - 3
sub1 - 3
sub2 - 2
sub2 - 1
sub1 - 1
sub2 - 0
sub1 - 0
main - 0
```

```
#include <stdio.h>

int main()
{
  int a = 10;

  R1: printf("sub1 - %d\n", a);

  R2: if (a > 0) goto R3; else goto R9;

  R3: a = a - 1;

  R4: printf("sub2 - %d\n", a);

  R5: if (a > 0 && a % 2 == 0) goto R6; else goto R8;

  R6: a = a - 1;

  R7: printf("sub2 - %d\n", a); goto R5;

  R8: printf("sub1 - %d\n", a); goto R2;

  R9: printf("main - %d", a);

  return 0;
}
```

```
sub1 - 10
sub2 - 9
sub1 - 9
sub2 - 8
sub2 - 7
sub1 - 7
sub2 - 6
sub2 - 5
sub1 - 5
sub2 - 4
sub2 - 3
sub1 - 3
sub2 - 2
sub2 - 1
sub1 - 1
sub2 - 0
sub1 - 0
main - 0
```