

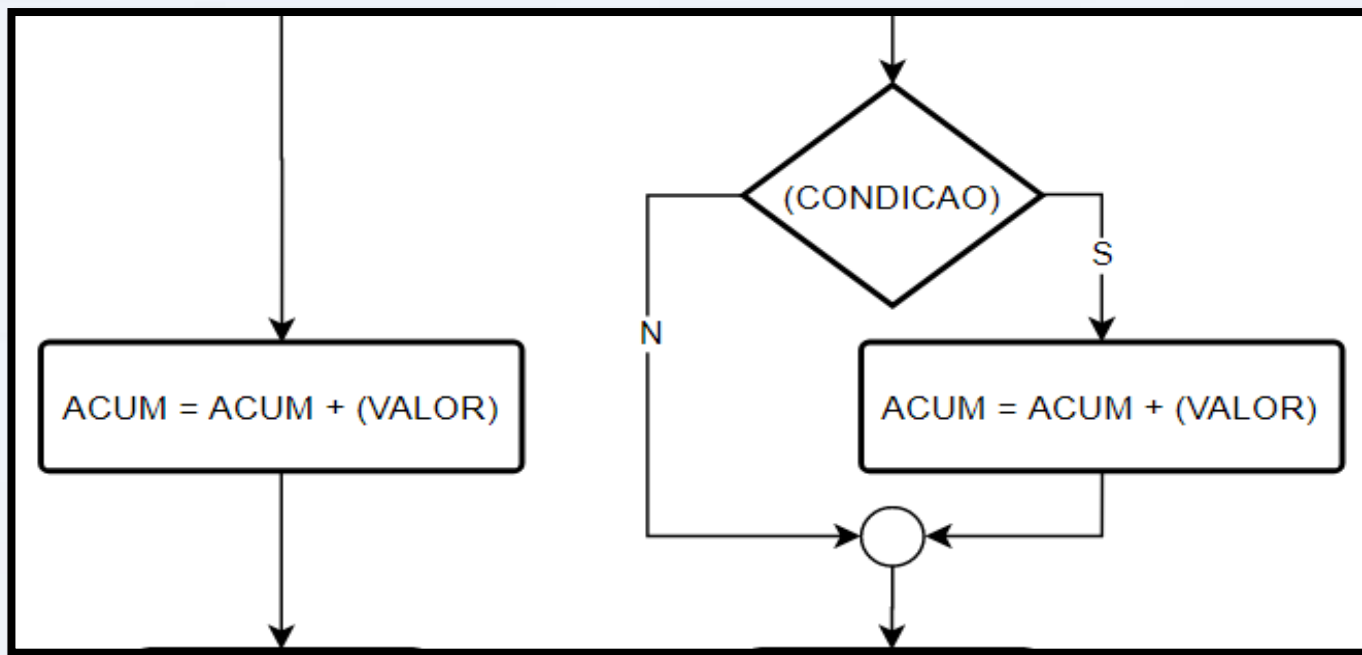
LABORATÓRIO DE PROGRAMAÇÃO

FUNDAMENTOS C

Contadores e Acumuladores

FUNDAMENTOS

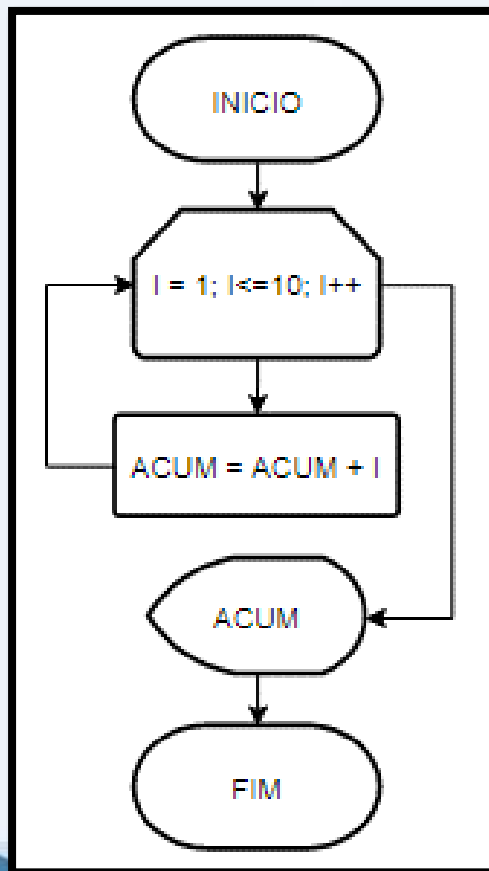
- ACUMULADOR



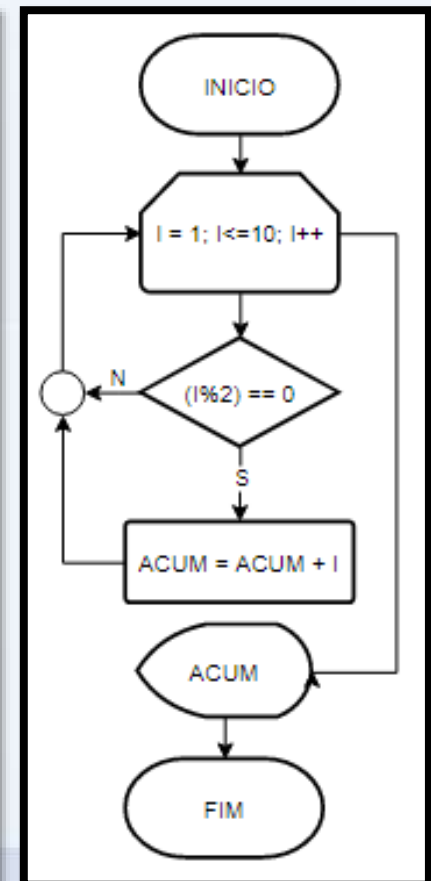
Marçal, 2020

FUNDAMENTOS

- **ACUMULADOR** – Consiste em uma variável responsável por armazenar (acumular) valores durante um determinado processamento.

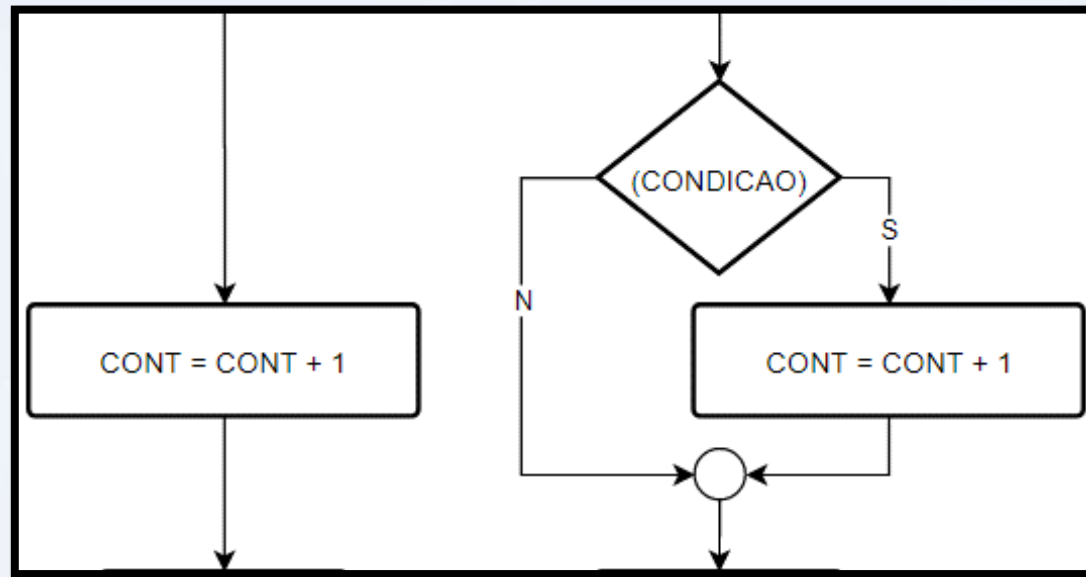


```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      int j = 10, i = 0;
6      int acum = 0;
7
8      /* Acumulando todos os valores
9       de um intervalo de valores */
10     for (i = 1; i <= 10; i++) {
11         acum = acum + i;
12     }
13     printf("Somar valores (1-10): %i \n", acum);
14
15     /* Acumulando todos os valores
16     pares de um intervalo */
17     acum = 0;
18     for (i = 1; i <= 10; i++) {
19         if ((i%2) == 0) {
20             acum = acum + i;
21         }
22     }
23     printf("Somar pares (1-10): %i \n", acum);
24
25     return 0;
26 }
```



FUNDAMENTOS

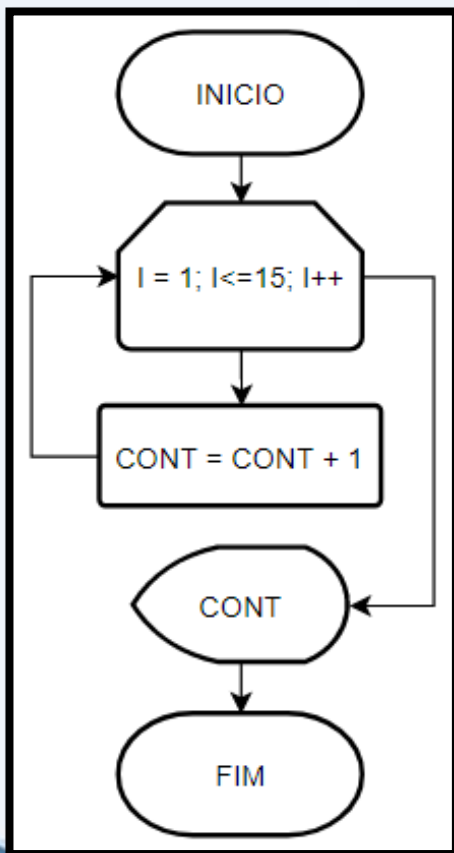
- CONTADOR



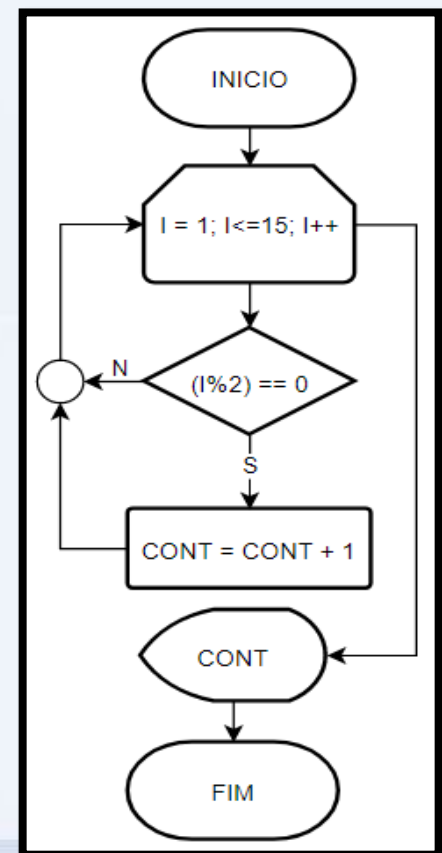
Marçal, 2020

FUNDAMENTOS

- **CONTADOR** – Consiste em uma variável responsável por contar o número de uma determinada ocorrência em um processamento.



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      int j = 10, i = 0;
6      int cont = 0;
7
8      /* Contar numero valores processados */
9      for (i = 1; i <= 15; i++) {
10         cont = cont + 1;
11     }
12     printf("Num. process. (1-15): %i \n", cont);
13
14     /* Contar numero pares processados */
15     cont = 0;
16     for (i = 1; i <= 15; i++) {
17         if ((i%2) == 0) {
18             cont = cont + 1;
19         }
20     }
21     printf("Num. pares (1-15): %i \n", cont);
22
23     return 0;
24 }
```



DÚVIDAS/PERGUNTAS

