UnB Departamento de Ciência da Computação

ALGORITMOS RESOLVIDOS

Pseudocódigo, Fluxograma, Linguagem C

Resumo

Lista de Algoritmos Resolvidos da Disciplina Algoritmos e Programação de Computadores.
Tratam-se da solução de algoritmos propostos pela plataforma URI Online Judge.
Departamento de Ciência da Computação - Instituto de Ciências Exatas.

Alexandre Zaghetto, Stéphannie Chiang, Rafaela Sinhoroto

zaghetto@unb.br, stephanniechiang@gmail.com, rsinhoroto@yahoo.com

Sequência Simples

Algoritmo 1

Leia 2 valores de ponto flutuante de dupla precisão A e B, que correspondem a 2 notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 3.5 e a nota B tem peso 7.5 (A soma dos pesos portanto é 11). Assuma que cada nota pode ir de 0 até 10.0, sempre com uma casa decimal.

Entrada

O arquivo de entrada contém 2 valores com uma casa decimal cada um.

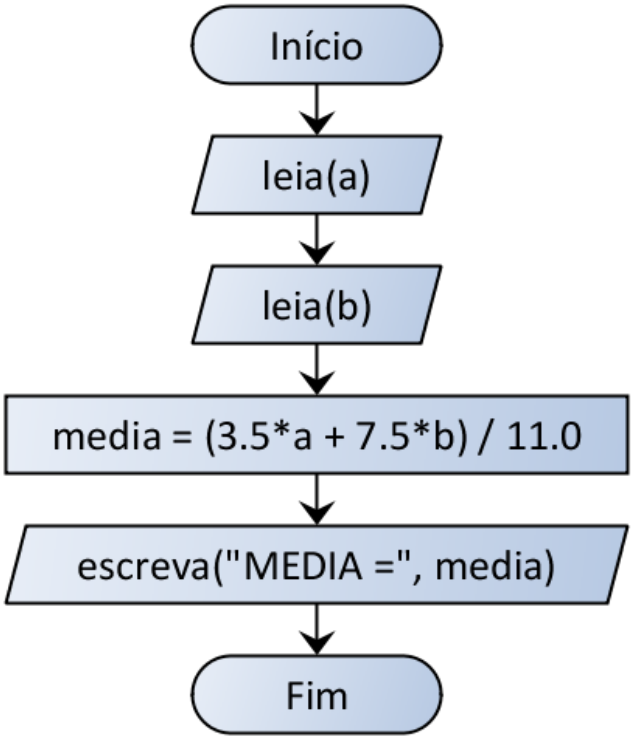
Saída

Calcule e imprima a variável **MEDIA** conforme exemplo abaixo, com 5 dígitos após o ponto decimal e com um espaço em branco antes e depois da igualdade. Utilize variáveis de dupla precisão (double) e como todos os problemas, não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "Presentation Error".

Pseudocódigo

```
algoritmo "sequencia_simples_1"

// seção de declarações
var
      real: a, b, media;

// seção de comandos
inicio
      leia(a);
      leia(b);

      media = (3.5*a + 7.5*b)/11.0;

      escreva("MEDIA = ", media);

fimalgoritmo
```

Fluxograma


Início
leia(a)
leia(b)
media = (3.5*a + 7.5*b) / 11.0
escreva("MEDIA =", media)
Fim

Linguagem de Programação C

```
#include <stdio.h>
#include <stdlib.h>

int main(){

double a, b, media;

scanf("%lf", &a);
scanf("%lf", &b);

media = (3.5*a + 7.5*b)/11.0;

printf("MEDIA = %.5lf\n", media);

return 0;

}
```

Algoritmo 2

Leia 3 valores, no caso variáveis A, B e C, que são as três notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 2, a nota B tem peso 3 e a nota C tem peso 5. Considere que cada nota pode ir de 0 até 10.0, sempre com uma casa decimal.

Entrada

O arquivo de entrada contém 3 valores com uma casa decimal, de dupla precisão (double).

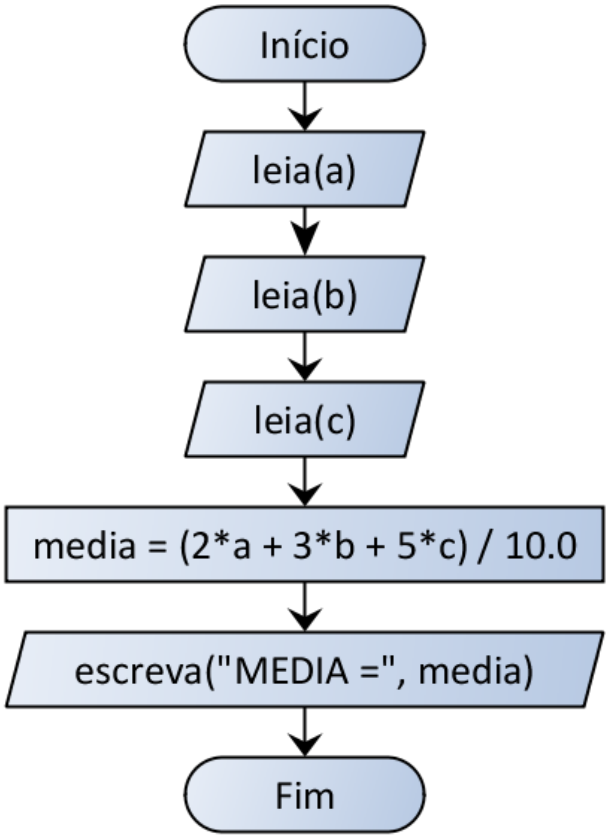
Saída

Imprima a variável **MEDIA** conforme exemplo abaixo, com 1 dígito após o ponto decimal e com um espaço em branco antes e depois da igualdade. Assim como todos os problemas, não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "Presentation Error".

Pseudocódigo

```
algoritmo "sequencia_simples_2"

// seção de declarações
var
      inteiro: a, b, c, media;

// seção de comandos
inicio
      leia(a);
      leia(b);
      leia(c);

      media = (2*a + 3*b + 5*c)/10.0;

      escreva("MEDIA = ", media);

fimalgoritmo
```

Fluxograma


Início
leia(a)
leia(b)
leia(c)
media = (2*a + 3*b + 5*c) / 10.0
escreva("MEDIA =", media)
Fim

Linguagem de Programação C

```
#include <stdio.h>
#include <stdlib.h>

int main(){

double a, b, c, media;

scanf("%lf", &a);
scanf("%lf", &b);
scanf("%lf", &c);

media = (2*a + 3*b + 5*c)/10.0;

printf("MEDIA = %.1lf\n", media);

return 0;

}
```

Algoritmo 3

Faça um programa que leia o nome de um vendedor, o seu salário fixo e o total de vendas efetuadas por ele no mês (em dinheiro). Sabendo que este vendedor ganha 15% de comissão sobre suas vendas efetuadas, informar o total a receber no final do mês, com duas casas decimais.

Entrada

O arquivo de entrada contém um texto (primeiro nome do vendedor) e 2 valores de dupla precisão (double) com duas casas decimais, representando o salário fixo do vendedor e montante total das vendas efetuadas por este vendedor, respectivamente.

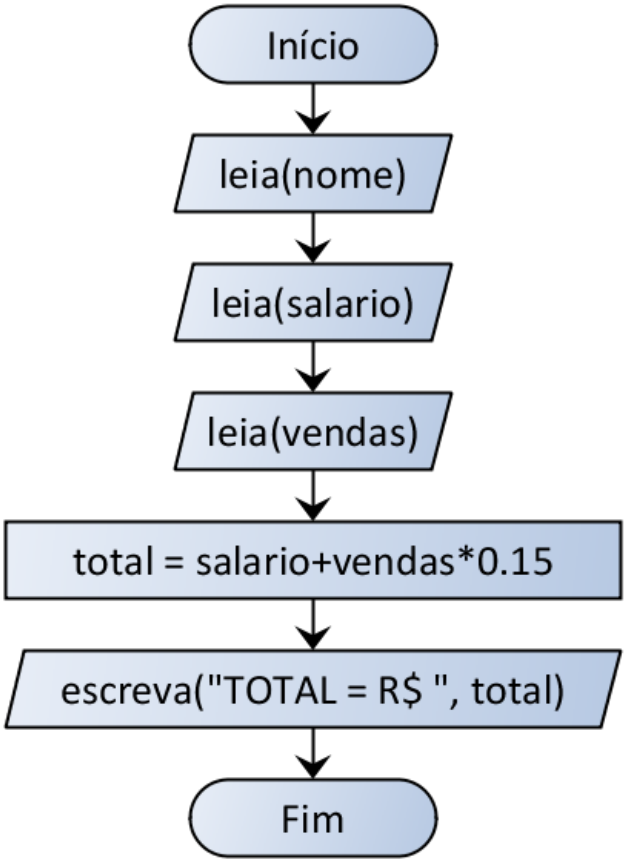
Saída

Imprima o total que o funcionário deverá receber, conforme exemplo fornecido.

Pseudocódigo

```
algoritmo "sequencia_simples_3"

// seção de declarações
var
      nome: vetor [100] de caracter;
      real: vendas;
      real: salario;
      real: total

// seção de comandos
inicio
      leia(nome);
      leia(salario);
      leia(vendas);

      total = salario+vendas*0.15;

      escreva("TOTAL = R$", total);

fimalgoritmo
```

Fluxograma


Início
leia(nome)
leia(salario)
leia(vendas)
total = salario+vendas*0.15
escreva("TOTAL = R$ ", total)
Fim

Linguagem de Programação C

```
#include <stdio.h>
#include <stdlib.h>

int main() {

char   nome[100];
double vendas;
double salario;

scanf("%[^\n]", nome);
scanf("%lf",     &salario);
scanf("%lf",     &vendas);

printf("TOTAL = R$ %.2f\n", salario+vendas*0.15);

return 0;
}
```

Algoritmo 4

Leia quatro valores inteiros A, B, C e D. A seguir, calcule e mostre a diferença do produto de A e B pelo produto de C e D segundo a fórmula: DIFERENCA = (A * B - C * D).

Entrada

O arquivo de entrada contém 4 valores inteiros.

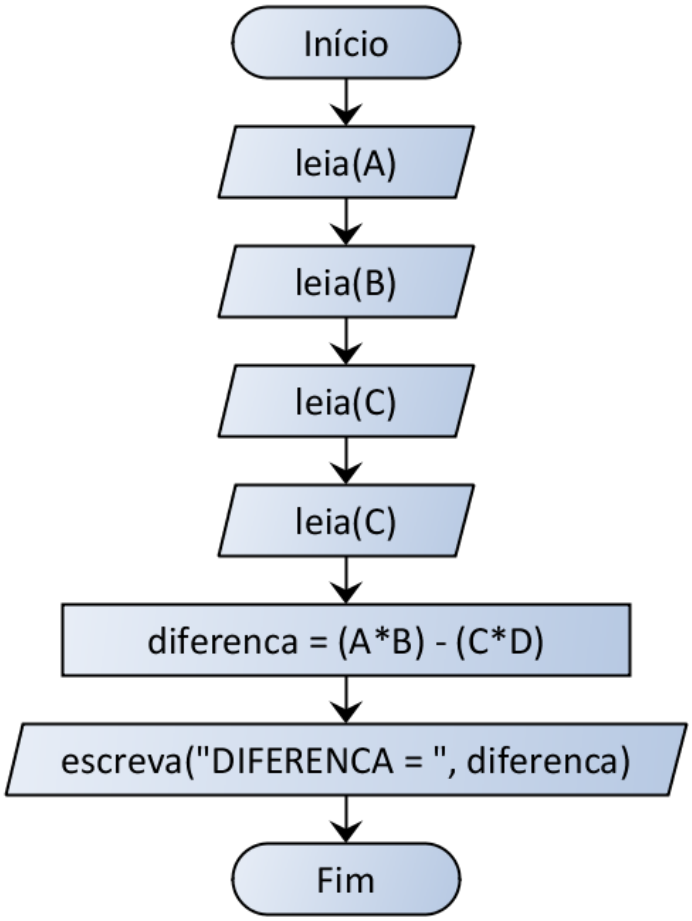
Saída

Imprima a mensagem **DIFERENCA** com todas as letras maiúsculas, conforme exemplo abaixo, com um espaço em branco antes e depois da igualdade.

Pseudocódigo

```
algoritmo "sequencia_simples_4"

// seção de declarações
var
      inteiro: A, B, C, D, DIFERENCA;

// seção de comandos
inicio
      leia(A);
      leia(B);
      leia(C);
      leia(D);

      DIFERENCA = (A * B) - (C * D);

      escreva("DIFERENCA = ", DIFERENCA);

fimalgoritmo
```

Fluxograma


Início
leia(A)
leia(B)
leia(C)
leia(C)
diferenca = (A*B) - (C*D)
escreva("DIFERENCA = ", diferenca)
Fim

Linguagem de Programação C

```
#include <stdio.h>
#include <stdlib.h>

int main() {

int A, B, C, D, DIFERENCA;

scanf("%d", &A);
scanf("%d", &B);
scanf("%d", &C);
scanf("%d", &D);

printf("DIFERENCA = %d\n",  (A * B - C * D) );

return 0;
}
```

Algoritmo 5

Escreva um programa que leia o número de um funcionário, seu número de horas trabalhadas, o valor que recebe por hora e calcula o salário desse funcionário. A seguir, mostre o número e o salário do funcionário, com duas casas decimais.

Entrada

O arquivo de entrada contém 2 números inteiros e 1 número com duas casas decimais, representando o número, quantidade de horas trabalhadas e o valor que o funcionário recebe por hora trabalhada, respectivamente.

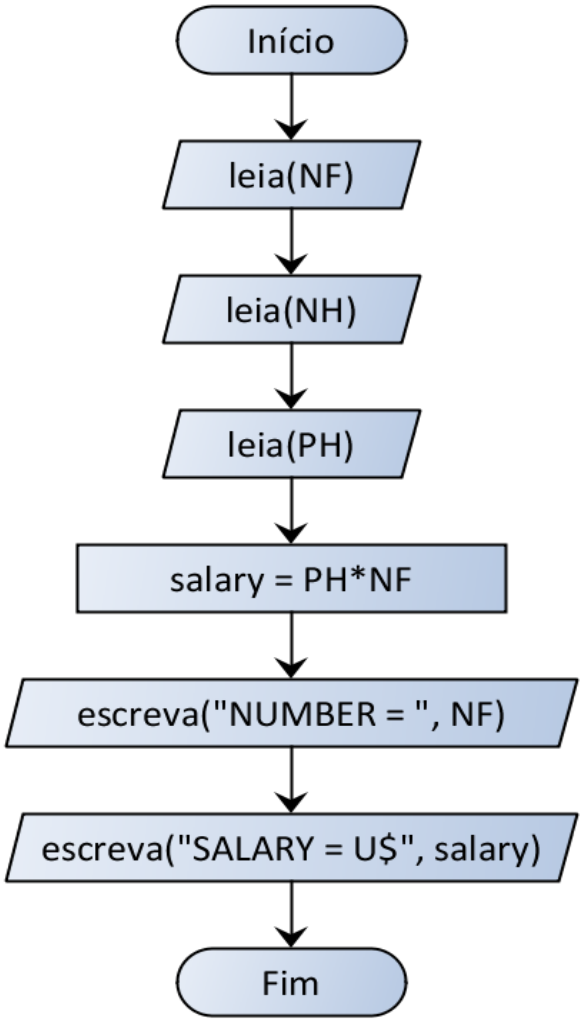
Saída

Imprima o número e o salário do funcionário, conforme exemplo fornecido, com um espaço em branco antes e depois da igualdade. No caso do salário, também deve haver um espaço em branco após o $.

Pseudocódigo

```
algoritmo "sequencia_simples_5"

// seção de declarações
var
      inteiro: NF, NH;
      real: PH, salary;

// seção de comandos
inicio
      leia(NF);
      leia(NH);
      leia(PH);

      salary = PH*NH;

      escreva("NUMBER = ", NF);
      escreva("SALARY = U$", salary);

fimalgoritmo
```

Fluxograma


Início
leia(NF)
leia(NH)
leia(PH)
salary = PH*NF
escreva("NUMBER = ", NF)
escreva("SALARY = U$", salary)
Fim

Linguagem de Programação C

```
#include <stdio.h>
#include <stdlib.h>

int main() {

int   NF, NH;
float PH;

scanf("%d", &NF);
scanf("%d", &NH);
scanf("%f", &PH);

printf("NUMBER = %d\n", NF);
printf("SALARY = U$ %.2f\n", PH*NH);

return 0;
}
```

Alternativas

Algoritmo 1

Leia 4 valores inteiros A, B, C e D. A seguir, se B for maior do que C e se D for maior do que A e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for par escrever a mensagem "**Valores aceitos**", senão escrever "**Valores nao aceitos**".

Entrada

Quatro números inteiros A, B, C e D.

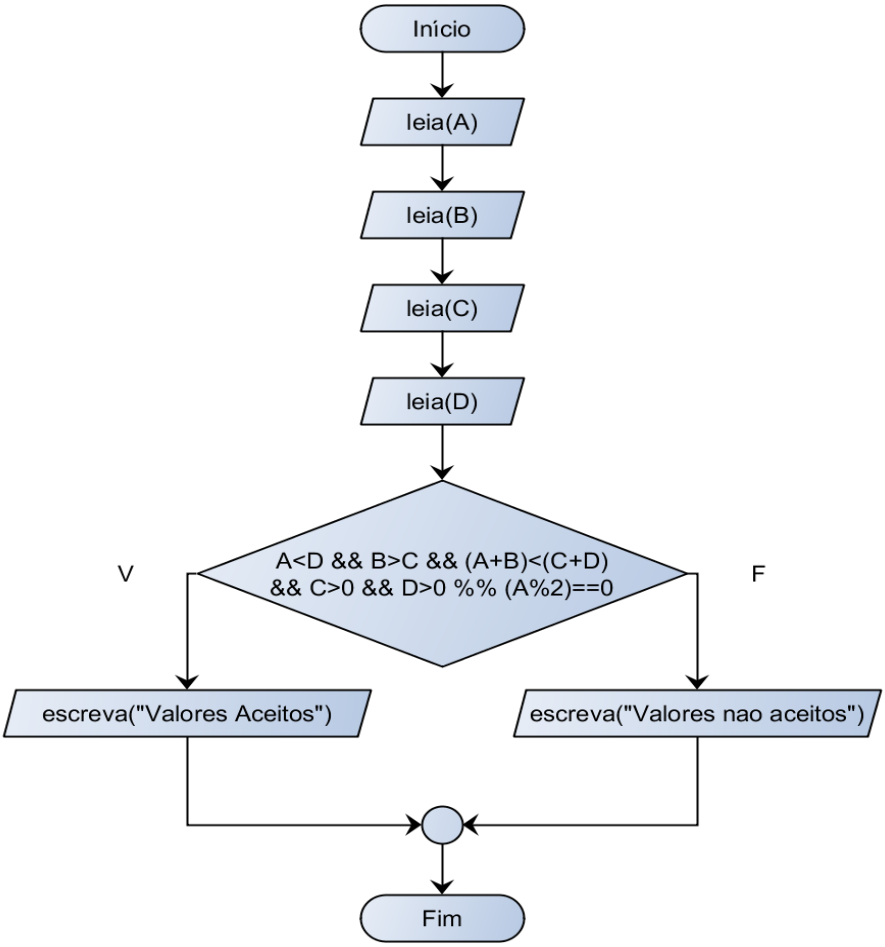
Saída

Mostre a respectiva mensagem após a validação dos valores.

Pseudocódigo

```
algoritmo "alternativas_1"

// seção de declarações
var
      inteiro: A, B, C, D;

// seção de comandos
inicio
      leia(A);
      leia(B);
      leia(C);
      leia(D);

      se (A<D && B>C && (A+B)<(C+D) && C>0 && D>0 && (A%2)==0) entao
            escreva("Valores aceitos");
      senao
            escreva("Valores nao aceitos");
      fimse

fimalgoritmo
```

Fluxograma


Início
leia(A)
leia(B)
leia(C)
leia(D)
A<D && B>C && (A+B)<(C+D)
&& C>0 && D>0 %% (A%2)==0
V
F
escreva("Valores Aceitos")
escreva("Valores nao aceitos")
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int A, B, C, D;

scanf("%d", &A);
scanf("%d", &B);
scanf("%d", &C);
scanf("%f", &D);

if(A<D && B>C && (A+B)<(C+D) && C>0 && D>0 && (A%2)==0)
      printf("Valores aceitos\n");
else
      printf("Valores nao aceitos\n");

return 0;
}
```

Algoritmo 2

Com base na tabela abaixo, escreva um programa que leia o código de um item e a quantidade deste item. A seguir, calcule e mostre o valor da conta a pagar.

CODIGO	ESPECIFICAÇÃO	PREÇO
1	Cachorro Quente	R$ 4.00
2	X-Salada	R$ 4.50
3	X-Bacon	R$ 5.00
4	Torrada simples	R$ 2.00
5	Refrigerante	R$ 1.50

Entrada

O arquivo de entrada contém dois valores inteiros correspondentes ao código e à quantidade de um item conforme tabela acima.

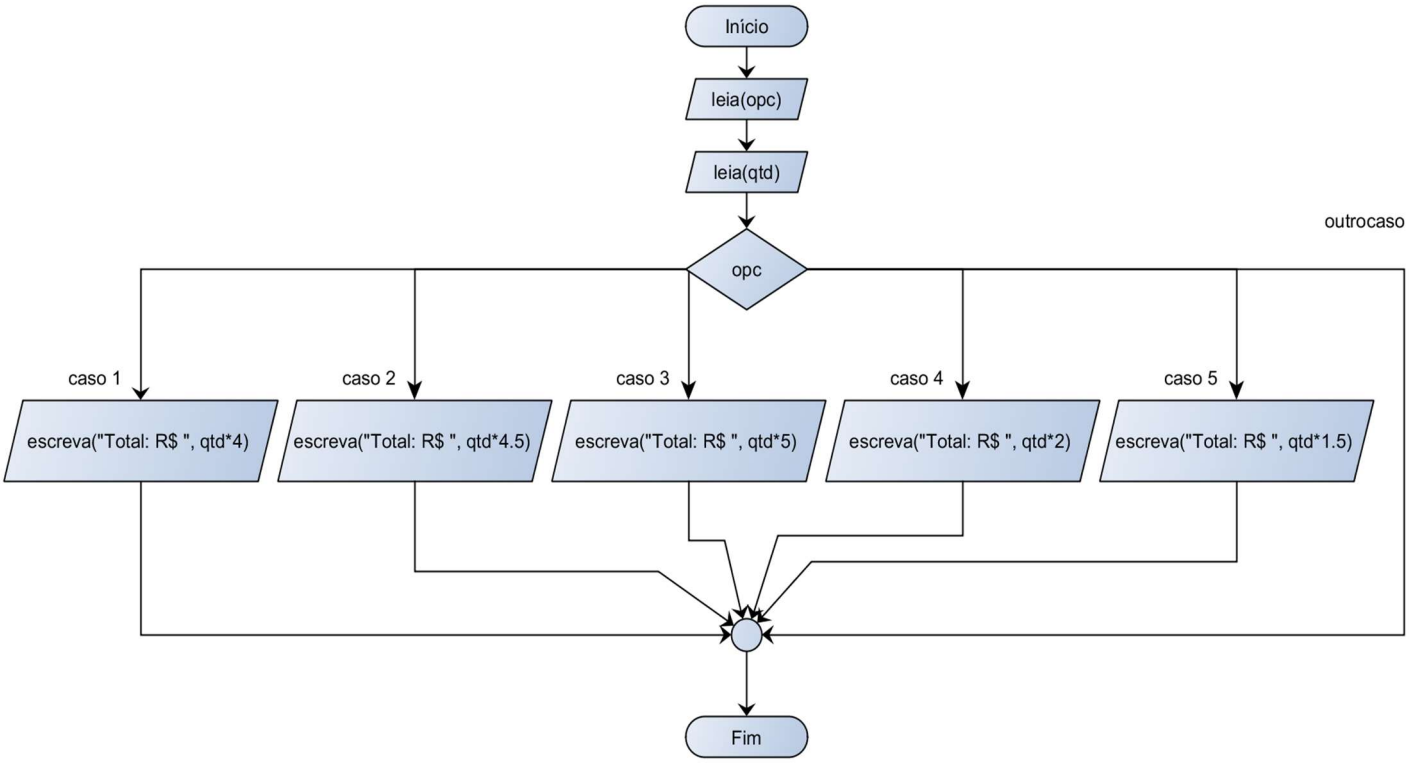
Saída

O arquivo de saída deve conter a mensagem "Total: R$ " seguido pelo valor a ser pago, com 2 casas após o ponto decimal.

Pseudocódigo

```
algoritmo "alternativas_2"

// seção de declarações
var
      inteiro: opc, qtd;

// seção de comandos
inicio
      leia(opc);
      leia(qtd);

      escolha opc
            caso 1
                  escreva("Total: R$", qtd*4.00);

            caso 2
                  escreva("Total: R$", qtd*4.50);

            caso 3
                  escreva("Total: R$", qtd*5.00);

            caso 4
                  escreva("Total: R$", qtd*2.00);

            caso 5
                  escreva("Total: R$", qtd*1.50);

            outrocaso

      fimescolha

fimalgoritmo
```

Fluxograma

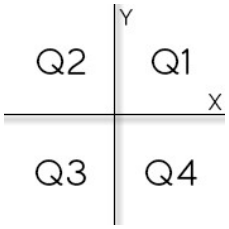

Início
leia(opc)
leia(qtd)
opc
outrocaso
caso 1
caso 2
caso 3
caso 4
caso 5
escreva("Total: R$ ", qtd*4)
escreva("Total: R$ ", qtd*4.5)
escreva("Total: R$ ", qtd*5)
escreva("Total: R$ ", qtd*2)
escreva("Total: R$ ", qtd*1.5)
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){
int opc, qtd;
scanf("%d %d", &opc, &qtd);
switch(opc){
      case 1:
            printf("Total: R$ %.2f\n", qtd*4.00);
            break;
      case 2:
            printf("Total: R$ %.2f\n", qtd*4.50);
            break;
      case 3:
            printf("Total: R$ %.2f\n", qtd*5.00);
            break;
      case 4:
            printf("Total: R$ %.2f\n", qtd*2.00);
            break;
      case 5:
            printf("Total: R$ %.2f\n", qtd*1.50);
            break;
      default:
            break;
}
return 0;
}
```

Algoritmo 3

Leia 2 valores com uma casa decimal (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem (x = y = 0).



Se o ponto estiver na origem, escreva a mensagem “Origem”.

Se o ponto estiver sobre um dos eixos escreva “Eixo X” ou “Eixo Y”, conforme for a situação.

Entrada

A entrada contem as coordenadas de um ponto.

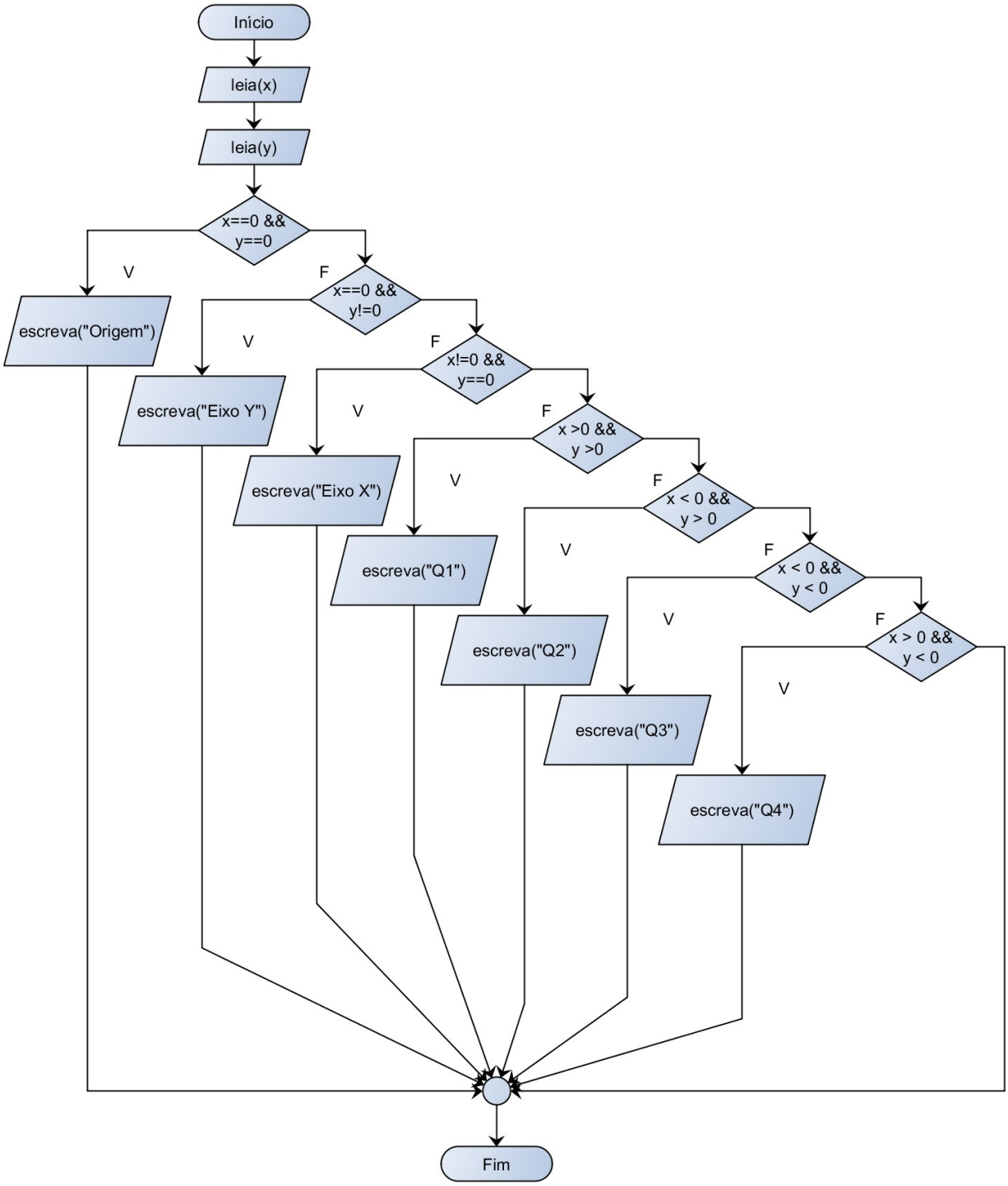
Saída

A saída deve apresentar o quadrante em que o ponto se encontra.

Pseudocódigo

```
algoritmo "alternativas_3"

// seção de declarações
var
      real: x, y;

// seção de comandos
inicio
      leia(x);
      leia(y);

      se (x==0 && y==0) entao
            escreva("Origem");

      senao se (x==0 && y!=0) entao
            escreva("Eixo Y");

      senao se (x!=0 && y==0) entao
            escreva("Eixo X");

      senao se (x>0 && y>0) entao
            escreva("Q1");

      senao se (x<0 && y>0) entao
            escreva("Q2");

      senao se (x<0 && y<0) entao
            escreva("Q3");

      senao se (x>0 && y<0) entao
            escreva("Q4");

      fimse

fimalgoritmo
```

Fluxograma


Início
leia(x)
leia(y)
x==0 && y==0
V
escreva("Origem")
F
x==0 && y!=0
V
escreva("Eixo Y")
F
x!=0 && y==0
V
escreva("Eixo X")
F
x >0 && y >0
V
escreva("Q1")
F
x < 0 && y > 0
V
escreva("Q2")
F
x < 0 && y < 0
V
escreva("Q3")
F
x > 0 && y < 0
V
escreva("Q4")
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

float x, y;

scanf("%f %f", &x, &y);

if(x==0 && y==0){
     printf("Origem\n");
}else if(x==0 && y!=0){
     printf("Eixo Y\n");
}else if(x!=0 && y==0){
     printf("Eixo X\n");
}else if(x>0 && y>0){
     printf("Q1\n");
}else if(x<0 && y>0){
     printf("Q2\n");
}else if(x<0 && y<0){
     printf("Q3\n");
}else if(x>0 && y<0){
     printf("Q4\n");
}

return 0;
}
```

Algoritmo 4

Leia 2 valores inteiros (A e B). Após, o programa deve mostrar uma mensagem "**Sao Multiplos**" ou "**Nao sao Multiplos**", indicando se os valores lidos são múltiplos entre si.

Entrada

A entrada contém valores inteiros.

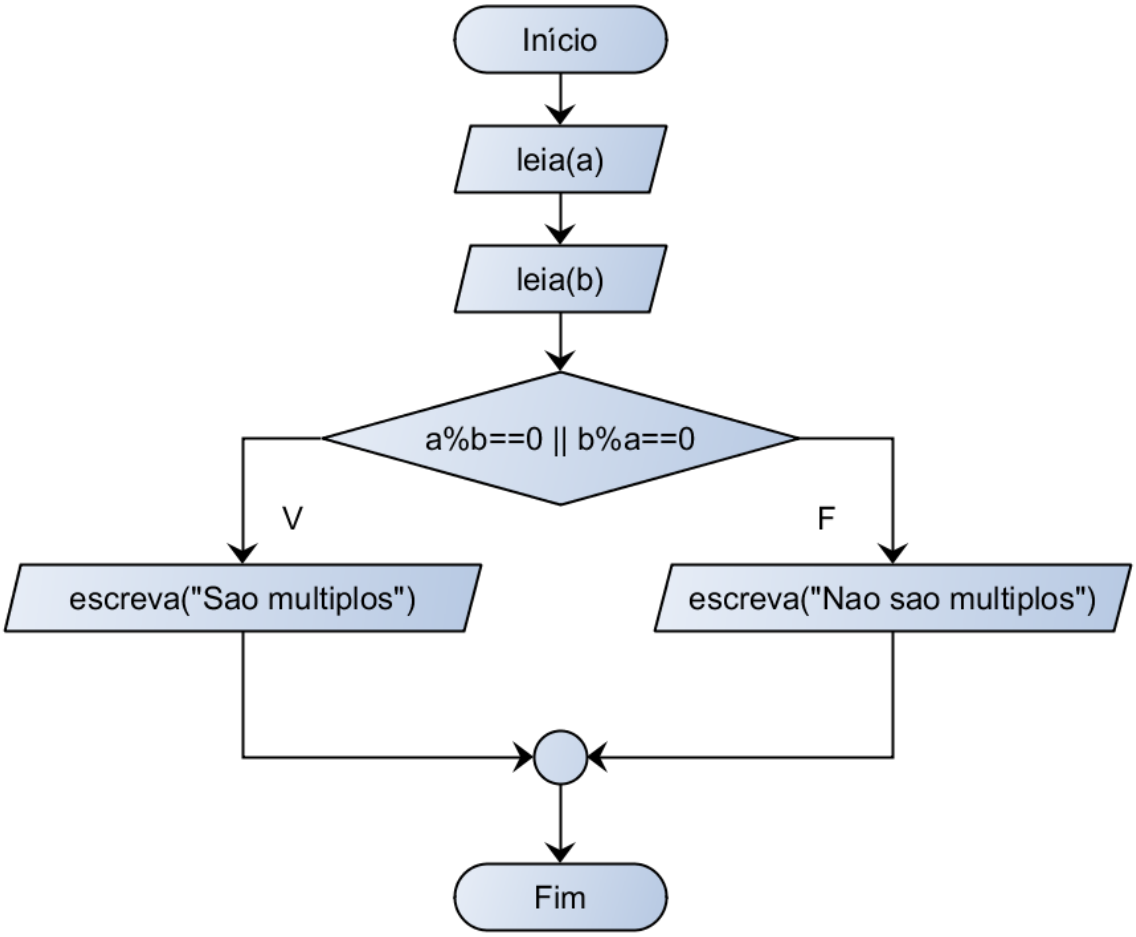
Saída

A saída deve conter uma das mensagens conforme descrito acima.

Pseudocódigo

```
algoritmo "alternativas_4"

// seção de declarações
var
      inteiro: a, b;

// seção de comandos
inicio
      leia(a);
      leia(b);

      se (a%b==0 || b%a==0) entao
            escreva("Sao Multiplos");
      senao
            escreva("Nao sao Multiplos");
      fimse

fimalgoritmo
```

Fluxograma


Início
leia(a)
leia(b)
a%b==0 || b%a==0
V
F
escreva("Sao multiplos")
escreva("Nao sao multiplos")
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int a, b;

scanf("%d", &a);
scanf("%d", &b);

if(a%b==0 || b%a==0){
      printf("Sao Multiplos\n");
}else{
      printf("Nao sao Multiplos\n");
}

return 0;
}
```

Algoritmo 5

Leia a hora inicial e a hora final de um jogo. A seguir calcule a duração do jogo, sabendo que o mesmo pode começar em um dia e terminar em outro, tendo uma duração máxima de 24 horas.

Entrada

Dois números inteiros representando o incío e o fim do jogo.

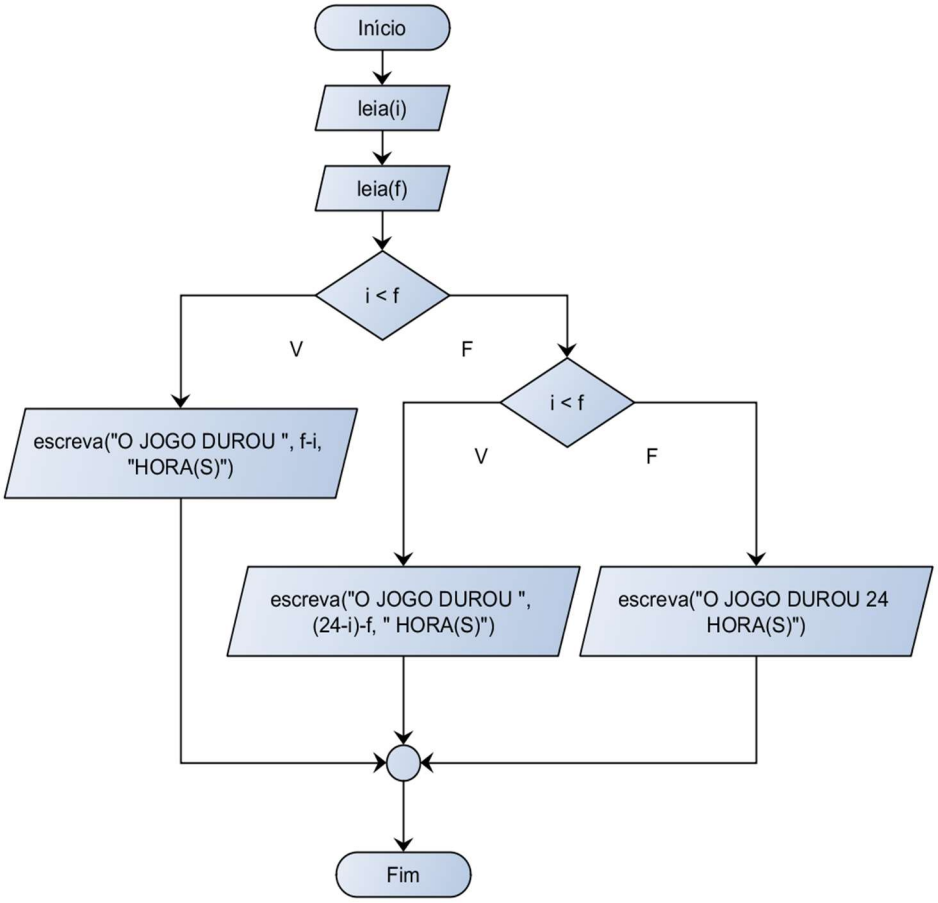
Saída

Mostre a duração do jogo conforme a seguinte mensagem: “O JOGO DUROU X HORA(S)”.

Pseudocódigo

```
algoritmo "alternativas_5"

// seção de declarações
var
      inteiro: i, f;

// seção de comandos
inicio
      leia(i);
      leia(f);

      se (f>i) entao
            escreva("O JOGO DUROU ", f-i, " HORA(S)");
      senao se (i>f) entao
            escreva("O JOGO DUROU ", (24-i)+f, " HORA(S)");
      senao se (i==f) entao
            escreva("O JOGO DUROU 24 HORA(S)");
      fimse

fimalgoritmo
```

Fluxograma


Início
leia(i)
leia(f)
i < f
V
F
escreva("O JOGO DUROU ", f-i, "HORA(S)")
i < f
V
F
escreva("O JOGO DUROU ", (24-i)-f, " HORA(S)")
escreva("O JOGO DUROU 24 HORA(S)")
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int i, f;

scanf("%d", &i);
scanf("%d", &f);

if(f>i){
     printf("O JOGO DUROU %d HORA(S)\n", f-i);
}else if(i>f){
     printf("O JOGO DUROU %d HORA(S)\n", (24-i)+f);
}else if (i==f){
     printf("O JOGO DUROU 24 HORA(S)\n");
}
return 0;
}
```

Repetições

Algoritmo 1

Faça um programa que mostre os números pares entre 1 e 100, inclusive.

Entrada

Neste problema extremamente simples de repetição não há entrada.

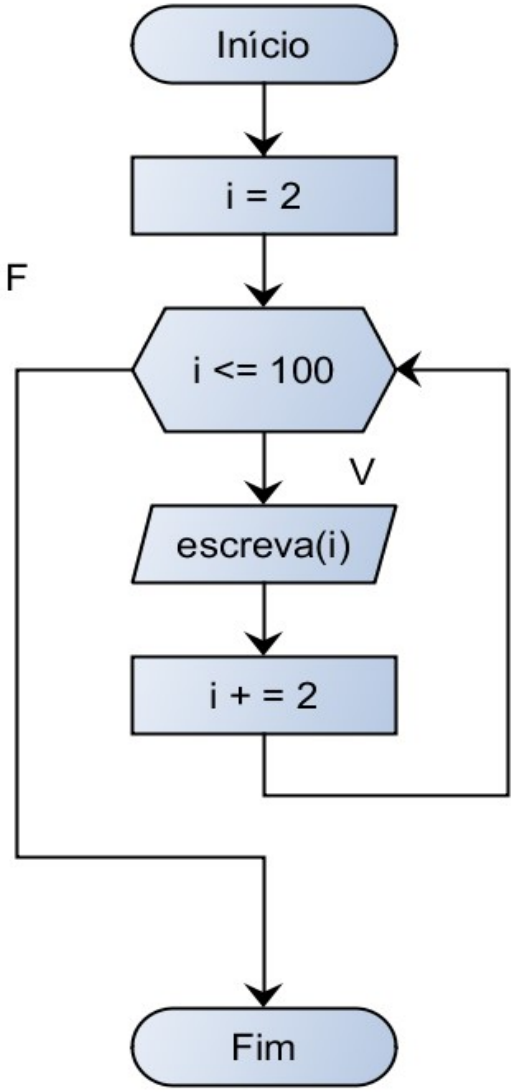
Saída

Imprima todos os números pares entre 1 e 100, inclusive se for o caso, um em cada linha.

Pseudocódigo

```
algoritmo "repeticoes_1"

// seção de declarações
var
      inteiro: i;

// seção de comandos
inicio
      para i de 2 ate 100 passo 2 faca
            escreva(i);
      fimpara

fimalgoritmo
```

Fluxograma



Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int i;

for(i=2; i<=100; i+=2)
     printf("%d\n", i);

return 0;
}
```

Algoritmo 2

Leia 1 valor inteiro N (2 < N < 1000). A seguir, mostre a tabuada de N:
1 x N = N 2 x N = 2N ... 10 x N = 10N

Entrada

A entrada contém um valor inteiro **N** (2 < **N** < 1000).

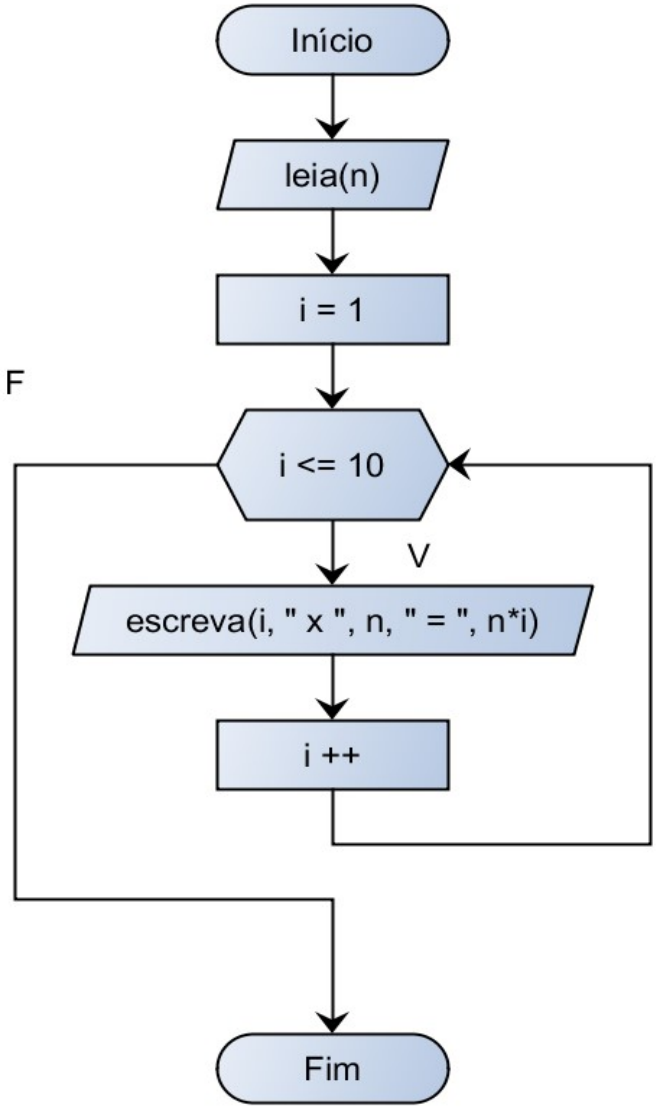
Saída

Imprima a tabuada de N, conforme o exemplo fornecido.

Pseudocódigo

```
algoritmo "repeticoes_2"

// seção de declarações
var
      inteiro: N, i;

// seção de comandos
inicio
      leia(N);

      para i de 1 ate 10 passo 1 faca
            escreva(i, " x ", N, " = ", N*i);
      fimpara

fimalgoritmo
```

Fluxograma



Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int N, i;

scanf("%d", &N);

for(i=1; i<=10; i++)
      printf("%d x %d = %d\n", i, N, N*i);

return 0;
}
```

Algoritmo 3

Leia um valor inteiro **N** que é a quantidade de casos de teste que vem a seguir. Cada caso de teste consiste de dois inteiros **X** e **Y**. Você deve apresentar a soma de todos os ímpares existentes ***entre*** **X** e **Y**.

Entrada

A primeira linha de entrada é um inteiro **N** que é a quantidade de casos de teste que vem a seguir. Cada caso de teste consiste em uma linha contendo dois inteiros **X** e **Y**.

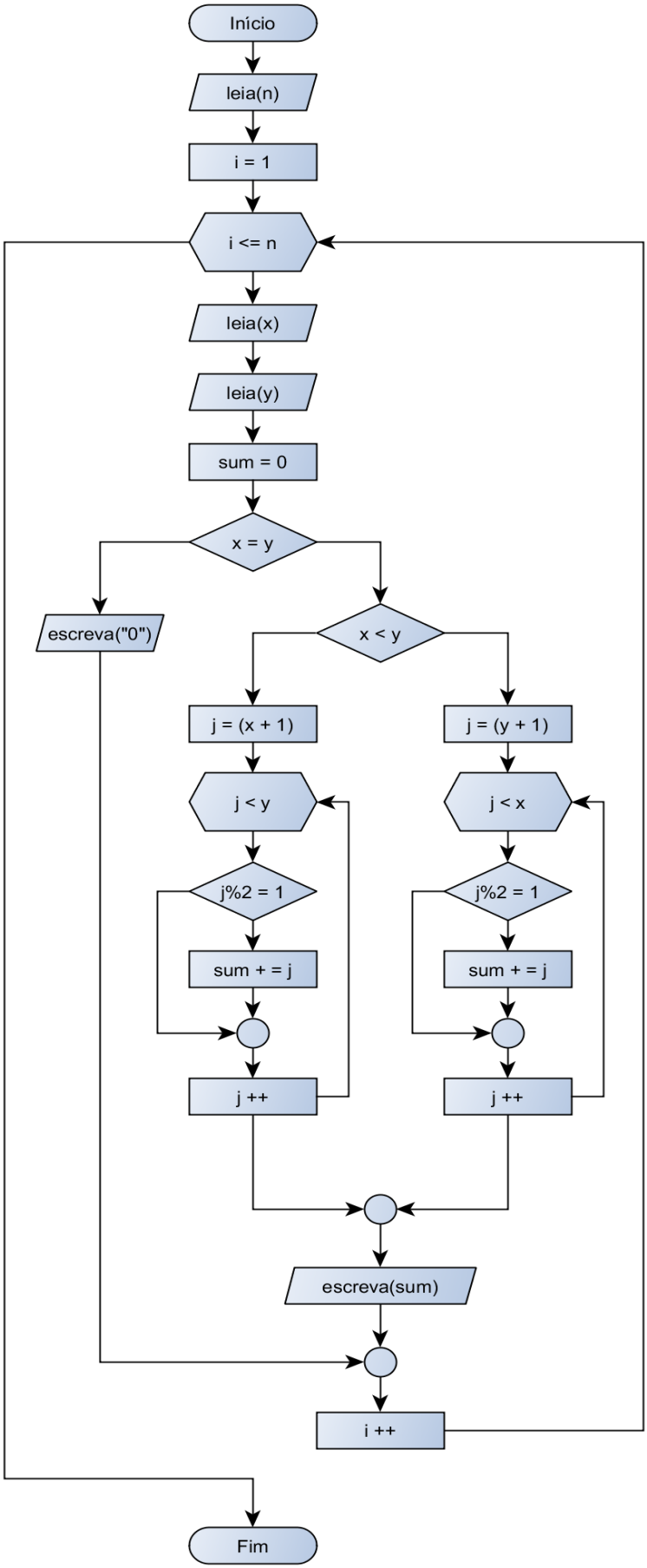
Saída

Imprima a soma de todos os valores ímpares ***entre*** **X** e **Y**.

Pseudocódigo

```
algoritmo "repeticoes_3"
// seção de declarações
var
      inteiro: N, x, y, i, j, sum;

// seção de comandos
inicio
      leia(N);

      para i de 1 ate N passo 1 faca
            leia(x);
            leia(y);

            se (x==y) entao
                  escreva("0");
            fimse
            se (x<y) entao
                  sum = 0;
                  para j de (x+1) ate y-1 passo 1 faca
                        se (j%2==1) entao
                              sum+=j;
                        fimse
                  fimpara
                  escreva(sum);
            fimse
            se (x>y) entao
                  sum = 0;
                  para j de (y+1) ate x-1 passo 1 faca
                        se (j%2==1) entao
                              sum+=j;
                        fimse
                  fimpara
                  escreva(sum);
            fimse
      fimpara

fimalgoritmo
```

Fluxograma


Início
leia(n)
i = 1
i <= n
leia(x)
leia(y)
sum = 0
x = y
escreva("0")
x < y
j = (x + 1)
j = (y + 1)
j < y
j < x
j%2 = 1
j%2 = 1
sum + = j
sum + = j
j ++
j ++
escreva(sum)
i ++
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int N, x, y, i, j, sum;

scanf("%d", &N);

for(i=1; i<=N; i++){
      scanf("%d %d", &x, &y);
      if(x==y)
            printf("0\n");
      if(x<y){
            for(j=(x+1), sum=0; j<y; j++)
                  if(j%2==1)
                        sum+=j;
            printf("%d\n", sum);
      }
      if(x>y){
            for(j=(y+1), sum=0; j<x; j++)
                  if(j%2==1)
                        sum+=j;
            printf("%d\n", sum);
      }
}
return 0;
}
```

Algoritmo 4

Faça um programa que leia as notas referentes às duas avaliações de um aluno. Calcule e imprima a média semestral. Faça com que o algoritmo só aceite notas válidas (uma nota válida deve pertencer ao intervalo [0,10]). Cada nota deve ser validada separadamente.

Entrada

A entrada contém vários valores reais, positivos ou negativos. O programa deve ser encerrado quando forem lidas duas notas válidas.

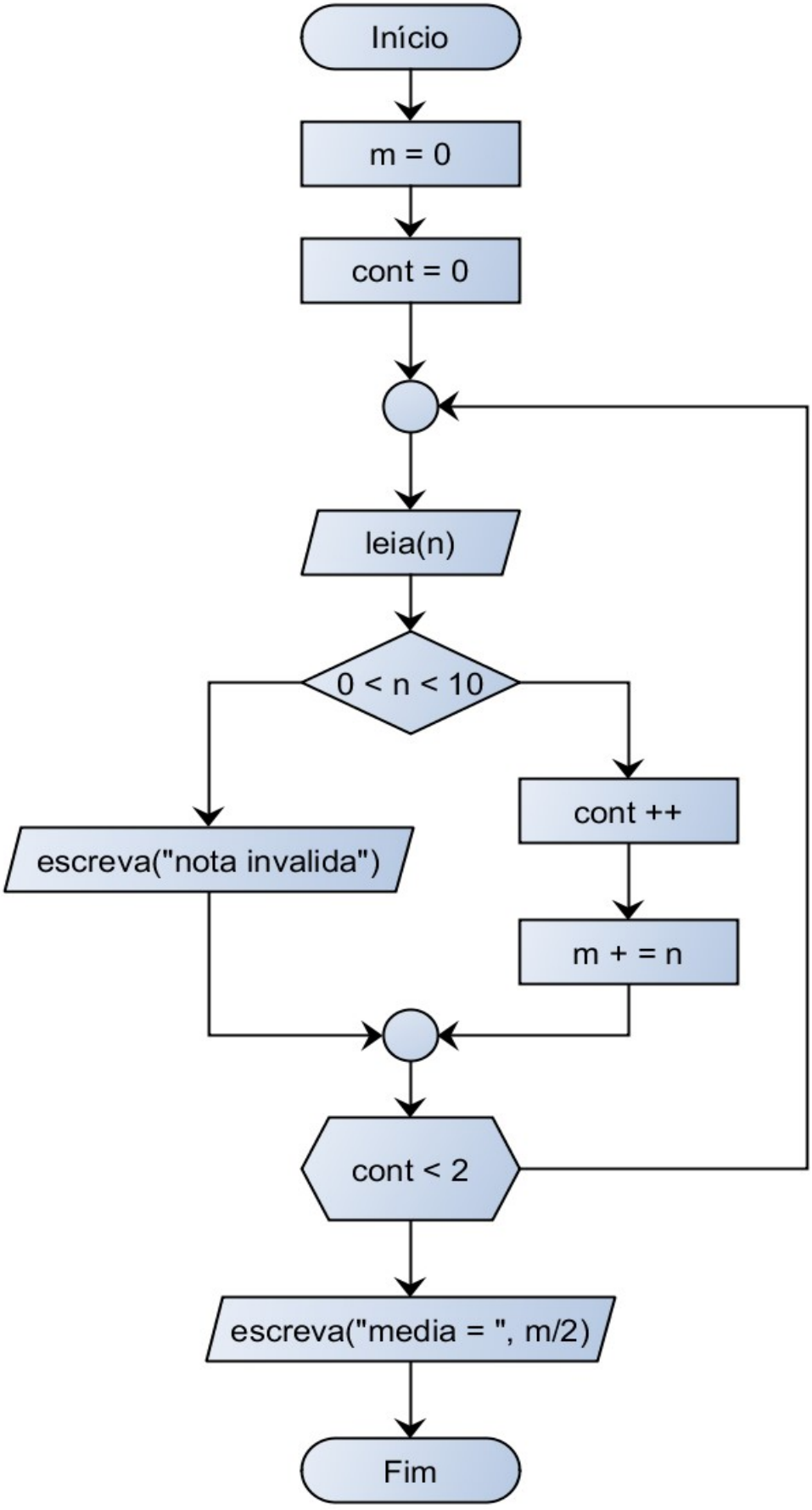
Saída

Se uma nota inválida for lida, deve ser impressa a mensagem "nota invalida". Quando duas notas válidas forem lidas, deve ser impressa a mensagem "media =" seguido do valor do cálculo. O valor deve ser apresentado com duas casas após o ponto decimal.

Pseudocódigo

```
algoritmo "repeticoes_4"

// seção de declarações
var
      real: n, m=0;
      inteiro: cont=0;

// seção de comandos
inicio
      faca
            leia(n);

            se (n<0 || n>10) entao
                  escreva("nota invalida");
            senao
                  cont++;
                  m+=n;
            fimse
      enquanto (cont<2);
      escreva("media = ", m/2);

fimalgoritmo
```

Fluxograma


Início
m = 0
cont = 0
leia(n)
0 < n < 10
cont ++
escreva("nota invalida")
m + = n
cont < 2
escreva("media = ", m/2)
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

float n, m=0;
int cont=0;

do{
      scanf("%f", &n);
      if(n<0 || n>10)
            printf("nota invalida\n");
      else{
            cont++;
            m+=n;
      }
}while(cont<2);
printf("media = %.2f\n", m/2);
return 0;
}
```

Algoritmo 5

Ler um valor N. Calcular e escrever seu respectivo fatorial. Fatorial de N = N * (N-1) * (N-2) * (N-3) * ... * 1.

Entrada

A entrada contém um valor inteiro N (0 < N < 13).

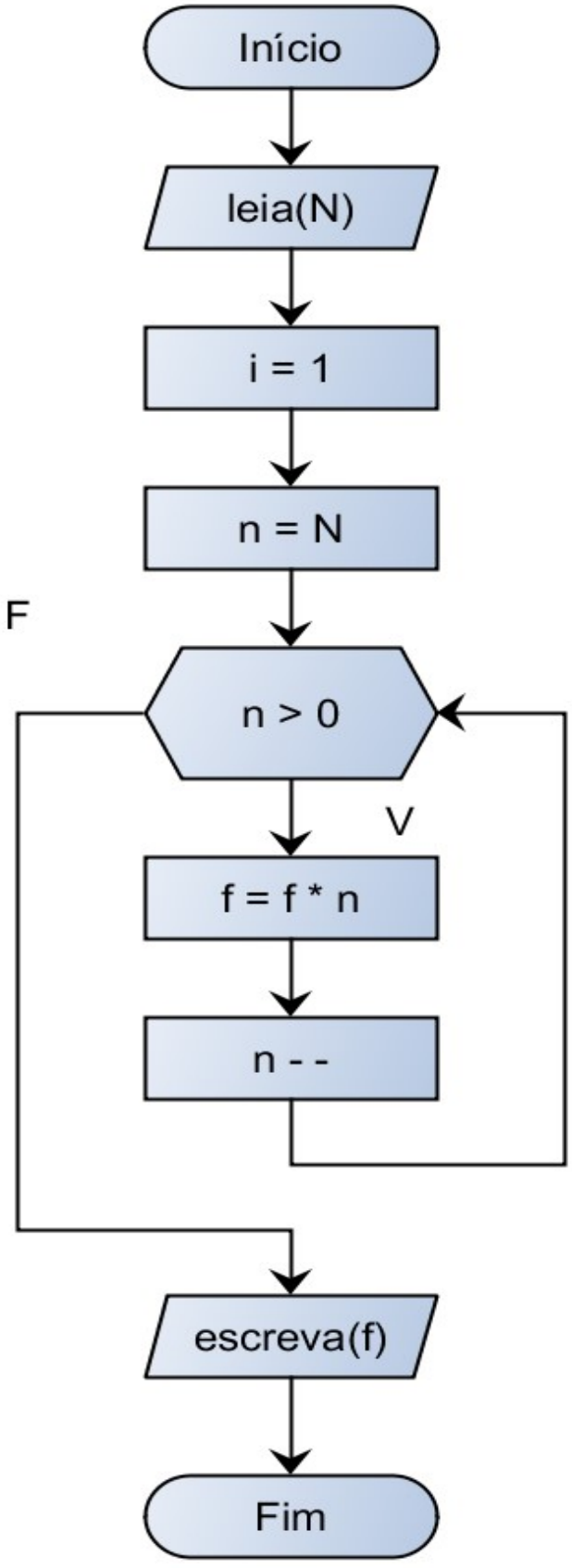
Saída

A saída contém um valor inteiro, correspondente ao fatorial de N.

Pseudocódigo

```
algoritmo "repeticoes_5"

// seção de declarações
var
      inteiro: N, f, n;

// seção de comandos
inicio
      leia(N);

      f = 1;
      para n de N ate 1 passo 1 faca
            f=f*n;
      fimpara

      escreva(f);

fimalgoritmo
```

Fluxograma



Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int N, f, n;

scanf("%d", &N);

for(n=N, f=1; n>0; n--)
      f=f*n;
printf("%d\n", f);

return 0;
}
```

Vetores

(obs.: de vetores haverá apenas dois exemplos devido à pequena quantidade de problemas dos mesmos)

Algoritmo 1

Faça um programa que leia um vetor X[10]. Substitua a seguir, todos os valores nulos e negativos do vetor X por 1. Em seguida mostre o vetor X.

Entrada

A entrada contém 10 valores inteiros, podendo ser positivos ou negativos.

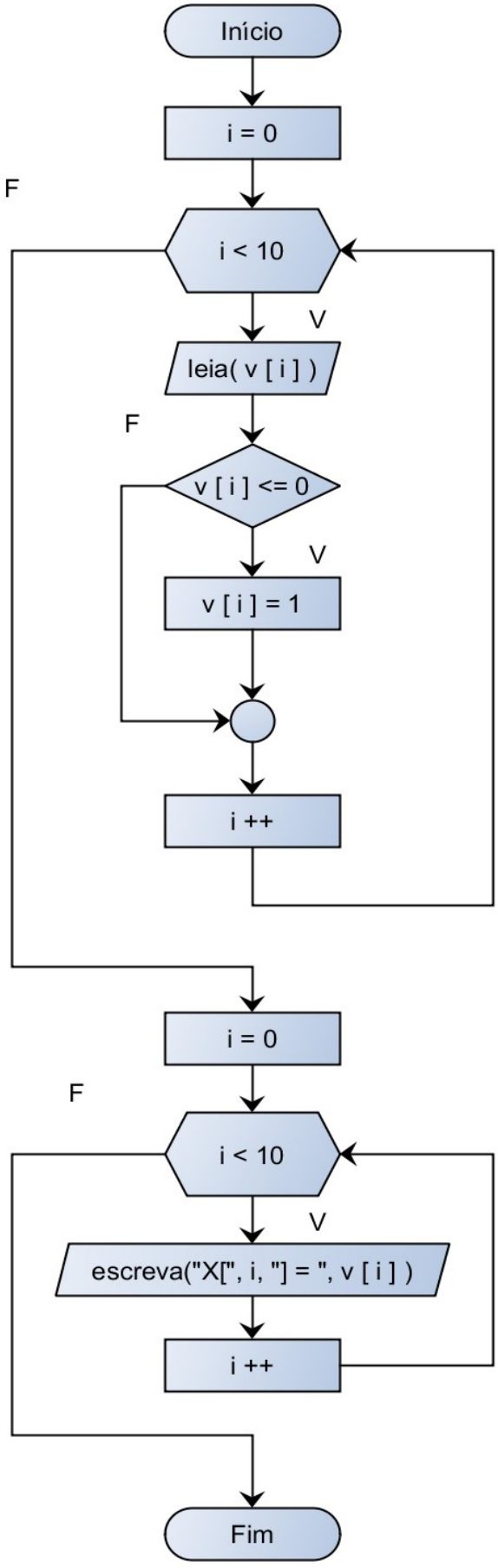
Saída

Para cada posição do vetor, escreva "X[**i**] = **x**", onde **i** é a posição do vetor e **x** é o valor armazenado naquela posição.

Pseudocódigo

```
algoritmo "vetores_1"

// seção de declarações
var
      inteiro: i;
      v: vetor [10] de inteiro;

// seção de comandos
inicio
      para i de 0 ate 9 passo 1 faca
            leia(v[i]);
            se (v[i]<=0) entao
                  v[i]=1;
            fimse
      fimpara

      para i de 0 ate 9 passo 1 faca
            escreva("X[", i, "] = ", v[i]);
      fimpara

fimalgoritmo
```

Fluxograma


Início
i = 0
F
i < 10
V
leia(v [i])
F
v [i] <= 0
V
v [i] = 1
i ++
i = 0
F
i < 10
V
escreva("X[", i, "] = ", v [i])
i ++
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int v[10], i;

for (i=0; i<10; i++){
      scanf("%d", &v[i]);
      if (v[i]<=0)
            v[i]=1;
}
for (i=0; i<10; i++)
      printf("X[%d] = %d\n", i, v[i]);

return 0;
}
```

Algoritmo 2

Leia um valor e faça um programa que coloque o valor lido na primeira posição de um vetor N[10]. Em cada posição subsequente, coloque o dobro do valor da posição anterior. Por exemplo, se o valor lido for 1, os valores do vetor devem ser 1,2,4,8 e assim sucessivamente. Mostre o vetor em seguida.

Entrada

A entrada contém um valor inteiro **(V<=50)**.

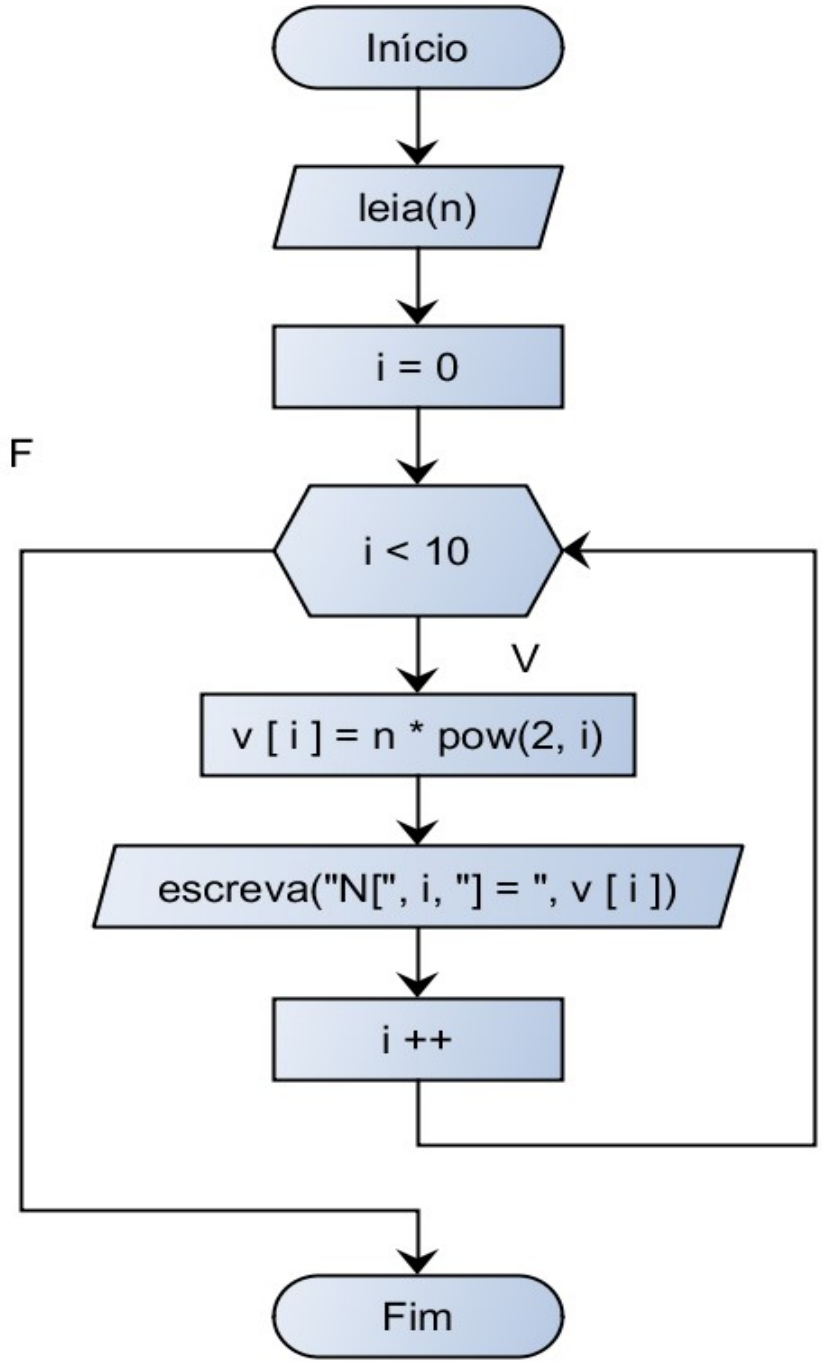
Saída

Para cada posição do vetor, escreva "N[**i**] = **X**", onde **i** é a posição do vetor e **X** é o valor armazenado na posição **i**. O primeiro número do vetor N (N[0]) irá receber o valor de V.

Pseudocódigo

```
algoritmo "vetores_2"

// seção de declarações
var
      inteiro: i, n;
      v: vetor [10] de inteiro;

// seção de comandos
inicio
      leia(n);
      para i de 0 ate 9 passo 1 faca
            v[i]=n*pow(2,i);
            escreva("N[", i, "] = ", v[i]);
      fimpara

fimalgoritmo
```

Fluxograma



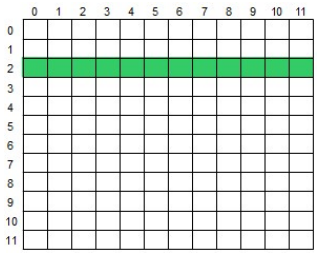
Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main(){

int v[10], i, n;

scanf("%d", &n);
for (i=0; i<10; i++){
      v[i]=n*pow(2,i);
      printf("N[%d] = %d\n", i, v[i]);
}

return 0;
}
```

Matrizes

Algoritmo 1

Neste problema você deve ler um número, indicando uma linha da matriz na qual uma operação deve ser realizada, um caractere maiúsculo, indicando a operação que será realizada, e todos os elementos de uma matriz **M**[12][12]. Em seguida, calcule e mostre a soma ou a média dos elementos que estão na área verde da matriz, conforme for o caso. A imagem abaixo ilustra o caso da entrada do valor 2 para a linha da matriz, demonstrando os elementos que deverão ser considerados na operação.



Entrada

A primeira linha de entrada contem um número **L** (0 ≤ **L** ≤ 11) indicando a linha que será considerada para operação. A segunda linha de entrada contém um único caractere Maiúsculo **T** ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem os 144 valores de ponto flutuante que compõem a matriz, sendo que a mesma é preenchida linha por linha.

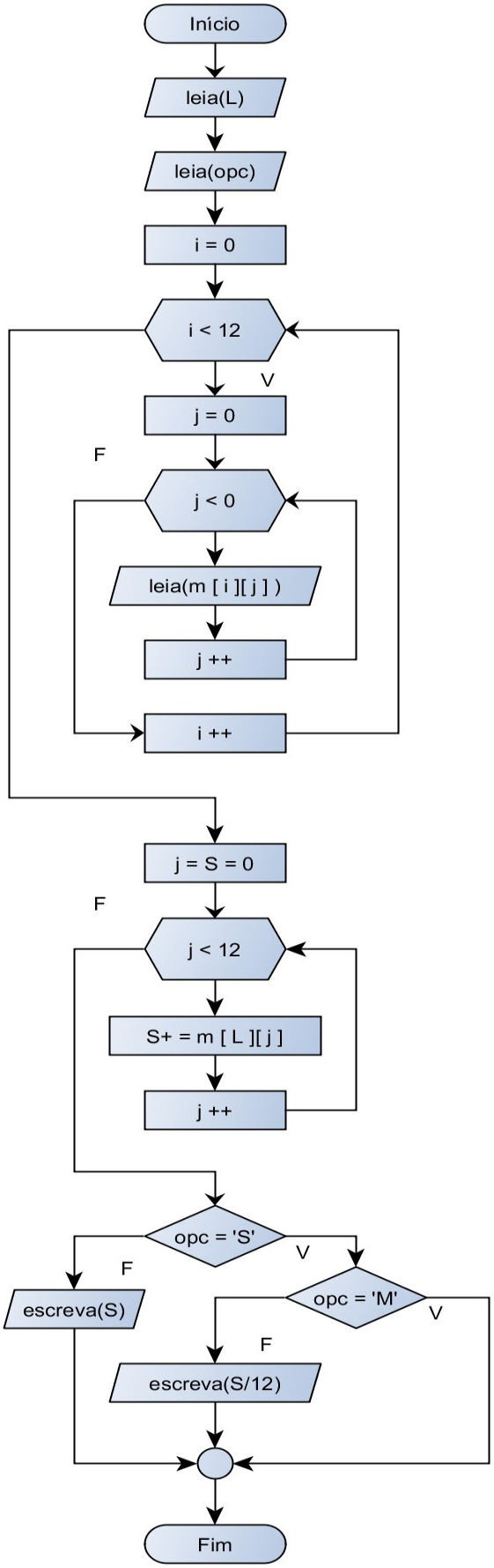
Saída

Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Pseudocódigo

```
algoritmo "matrizes_1"

// seção de declarações
var
      inteiro: l, i, j;
      caracter: opc;
      real: S;
      m: matriz [12,12] de real;

// seção de comandos
inicio
      leia(l);
      leia(opc);

      para i de 0 ate 11 passo 1 faca
            para j de 0 ate 11 passo 1 faca
                  leia(m[i][j]);
            fimpara
      fimpara

      para j de 0 ate 11 passo 1 faca
                  S+=m[l][j];
      fimpara

      se (opc=='S') entao
            escreva(S);
      senao se (opc=='M') entao
            escreva(S/12);
      fimse

fimalgoritmo
```

Fluxograma

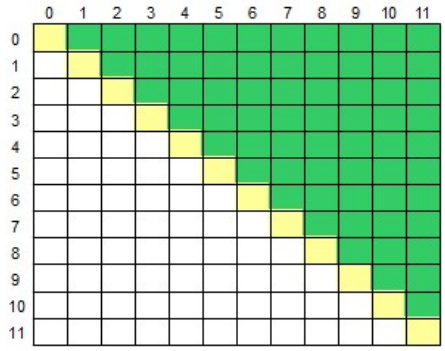

Início
leia(L)
leia(opc)
i = 0
i < 12
V
j = 0
F
j < 0
leia(m [i][j])
j ++
i ++
j = S = 0
F
j < 12
S+ = m [L][j]
j ++
opc = 'S'
V
F
escreva(S)
opc = 'M'
V
F
escreva(S/12)
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int l, i, j;
char opc;
float m[12][12], S;
scanf("%d %c", &l, &opc);
for(i=0; i<12; i++)
     for(j=0; j<12; j++)
          scanf("%f", &m[i][j]);
for(j=0, s=0; j<12; j++)
          S+=m[l][j];
if(opc=='S')
     printf("%.1f\n", S);
else if(opc=='M')
     printf("%.1f\n", S/12);
return 0;
}
```

Algoritmo 2

Leia um caractere maiúsculo, que indica uma operação que deve ser realizada e uma matriz **M**[12][12]. Em seguida, calcule e mostre a soma ou a média considerando somente aqueles elementos que estão acima da diagonal principal da matriz, conforme ilustrado abaixo (área verde).



Entrada

A primeira linha de entrada contem um único caractere Maiúsculo **O** ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem os 144 valores de ponto flutuante que compõem a matriz.

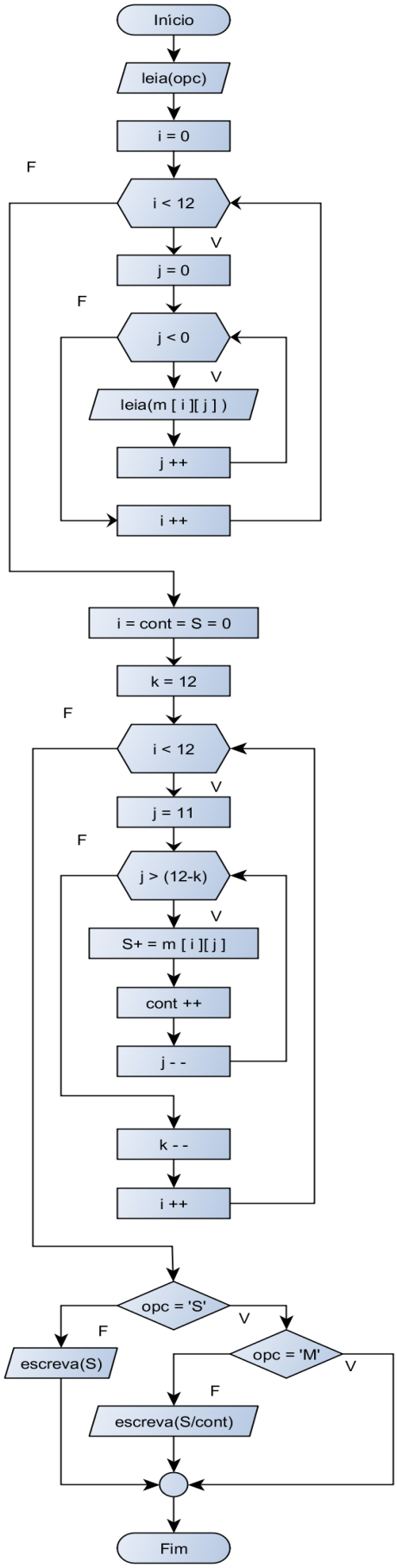
Saída

Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Pseudocódigo

```
algoritmo "matrizes_2"

// seção de declarações
var
      inteiro: l, i, j, cont, k;
      caracter: O;
      real: S;
      m: matriz [12,12] de real;

// seção de comandos
inicio
      leia(O);

      para i de 0 ate 11 passo 1 faca
            para j de 0 ate 11 passo 1 faca
                  leia(m[i][j]);
            fimpara
      fimpara

      k = 12;
      cont = 0;
      para i de 0 ate 11 passo 1 faca
            k--;
            para j de 11 ate (12-k) passo 1 faca
                  cont++;
                  S+=m[i][j];
            fimpara
      fimpara

      se (O=='S') entao
            escreva(S);
      senao se (O=='M') entao
            escreva(S/cont);
      fimse
fimalgoritmo
```

Fluxograma

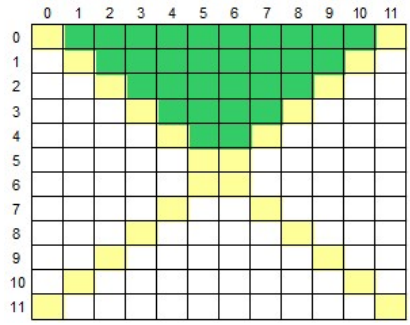

Inicio
leia(opc)
i = 0
F
i < 12
V
j = 0
F
j < 0
V
leia(m [i][j])
j ++
i ++
i = cont = S = 0
k = 12
F
i < 12
V
j = 11
F
j > (12-k)
V
S+ = m [i][j]
cont ++
j - -
k - -
i ++
opc = 'S'
V
F
escreva(S)
opc = 'M'
V
F
escreva(S/cont)
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

int l, i, j, cont, k;
char O;
float m[12][12], S;
scanf("%c", &O);
for(i=0; i<12; i++)
     for(j=0; j<12; j++)
          scanf("%f", &m[i][j]);
for(i=0, k=12, cont=0; i<12; i++, k--)
     for(j=11; j>(12-k); j--, cont++)
          S+=m[i][j];
if(O=='S')
     printf("%.1f\n", S);
else if(O=='M')
     printf("%.1f\n", S/cont);
return 0;
}
```

Algoritmo 3

Leia um caractere maiúsculo, que indica uma operação que deve ser realizada e uma matriz **M**[12][12]. Em seguida, calcule e mostre a soma ou a média considerando somente aqueles elementos que estão na área superior da matriz, conforme ilustrado abaixo (área verde).



Entrada

A primeira linha de entrada contem um único caractere Maiúsculo **O** ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem 144 valores com ponto flutuante de dupla precisão que compõem a matriz.

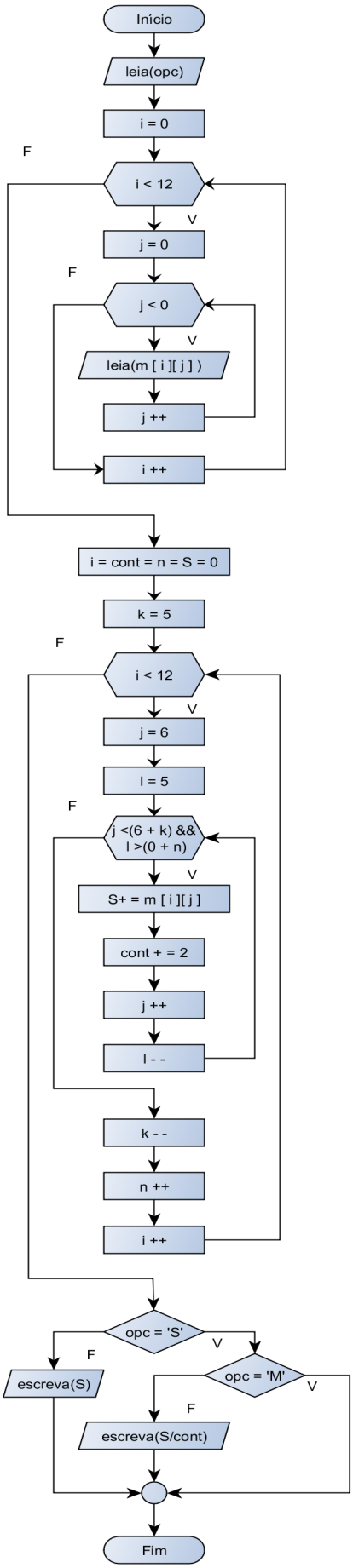
Saída

Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Pseudocódigo

```
algoritmo "matrizes_3"

// seção de declarações
var
      inteiro: i, j, cont, k, l, n;
      caracter: O;
      real: S;
      m: matriz [12,12] de real;

// seção de comandos
inicio
      leia(O);

      para i de 0 ate 11 passo 1 faca
            para j de 0 ate 11 passo 1 faca
                  leia(m[i][j]);
            fimpara
      fimpara

      k = 5;
      n = 0;
      cont = 0;
      para i de 0 ate 4 passo 1 faca
            k--;
            n++;
            para j de 6 ate (6+k) passo 1 e l de 5 ate (0+n) passo 1 faca
                  cont+=2;
                  S+=m[i][j];
            fimpara
      fimpara

      se (O=='S') entao
            escreva(S);
      senao se (O=='M') entao
            escreva(S/cont);
      fimse
fimalgoritmo
```

Fluxograma


Início
leia(opc)
i = 0
F
i < 12
V
j = 0
F
j < 0
V
leia(m [i][j])
j ++
i ++
i = cont = n = S = 0
k = 5
F
i < 12
V
j = 6
l = 5
F
j <(6 + k) &&
l >(0 + n)
V
S+ = m [i][j]
cont + = 2
j ++
l - -
k - -
n ++
i ++
opc = 'S'
V
F
escreva(S)
opc = 'M'
V
F
escreva(S/cont)
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main(){

      int i, j, cont, k, l, n;
      char O;
      float m[12][12], S;
      scanf("%c", &O);
      for(i=0; i<12; i++)
            for(j=0; j<12; j++)
                  scanf("%f", &m[i][j]);
      for(i=0, k=5, n=0, cont=0; i<5; i++, k--, n++)
            for(j=6, l=5; j<(6+k), l>(0+n); j++, cont+=2, l--)
                  S+=m[i][j];
      if(O=='S')
            printf("%.1f\n", S);
      else if(O=='M')
            printf("%.1f\n", S/cont);
      return 0;
}
```

Algoritmo 4

Escreva um algoritmo que leia um inteiro N (0 ≤ N ≤ 100), correspondente a ordem de uma matriz M de inteiros, e construa a matriz de acordo com o exemplo abaixo.

Entrada

A entrada consiste de vários inteiros, um valor por linha, correspondentes as ordens das matrizes a serem construídas. O final da entrada é marcado por um valor de ordem igual a zero (0).

Saída

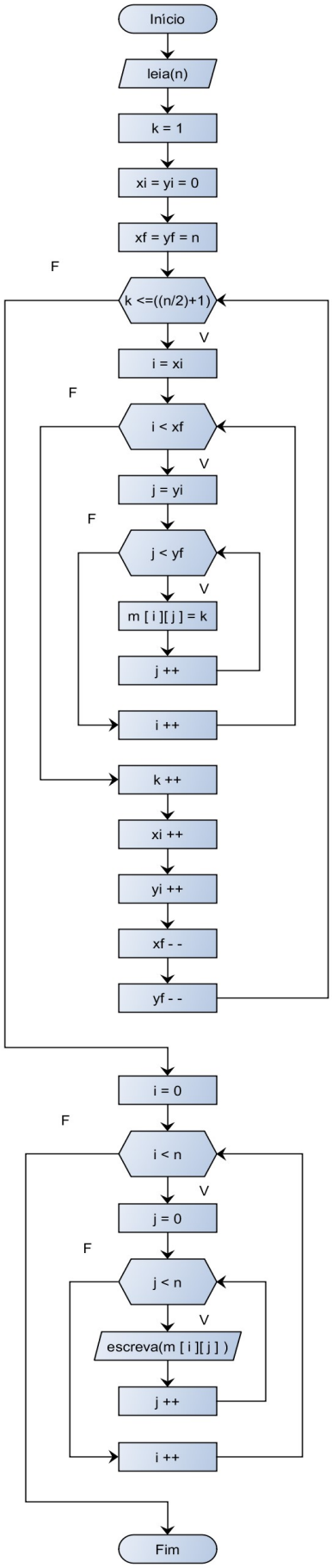
Para cada inteiro da entrada imprima a matriz correspondente, de acordo com o exemplo. Os valores das matrizes devem ser formatados em um campo de tamanho 3 justificados à direita e separados por espaço. Após o último caractere de cada linha da matriz não deve haver espaços em branco. Após a impressão de cada matriz deve ser deixada uma linha em branco.

Exemplo de Entrada	Exemplo de Saída
1 2 3 4 5 0	1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 3 2 1 1 2 2 2 1 1 1 1 1 1

Pseudocódigo

```
algoritmo "matrizes_4”

// seção de declarações
var
      inteiro: n, i, j, k, xi, yi, xf, yf;
      m: matriz de inteiro;

// seção de comandos
inicio
      faca
            leia(n);

            m: matriz [n,n] de inteiro;

            xi = 0;
            yi = 0;
            xf = n;
            yf = n;
            para k de 1 ate ((n/2)+1) passo 1 faca
                  xi++;
                  yi++;
                  xf—-;
                  yf—-;
                  para i de xi ate xf passo 1 faca
                        para j de yi ate yf passo 1 faca
                                    m[i][j] = k;
                        fimpara
                  fimpara
            fimpara

            para i de 0 ate n passo 1 faca
                  para j de 0 ate n passo 1 faca
                        se (j == (n-1)) entao
                                    escreva (m[i][j]);
                        senao
                                    escreva (m[i][j]);
                              escreva (" ");
                        fimse
                  fimpara
            fimpara

      enquanto (n > 0);
fimalgoritmo
```

Fluxograma


Início
leia(n)
k = 1
xi = yi = 0
xf = yf = n
F
k <=((n/2)+1)
V
i = xi
F
i < xf
V
j = yi
F
j < yf
V
m [i][j] = k
j ++
i ++
k ++
xi ++
yi ++
xf - -
yf - -
i = 0
F
i < n
V
j = 0
F
j < n
V
escreva(m [i][j])
j ++
i ++
Fim

Linguagem de Programação C

```
#include <stdlib.h>
#include <stdio.h>

int main () {

    int n, i, j, k, xi, yi, xf, yf;
    int **m;

    do {
        scanf("%d", &n);

        m = (int**) malloc (n*sizeof(int*));
        for (i = 0; i < n; i++)
            m[i] = (int*) malloc (n*sizeof(int));


        for (k = 1, xi = 0, yi = 0, xf = n, yf = n; k <= ((n/2)+1);
k++, xi++, yi++, xf--, yf--)
            for (i = xi; i < xf; i++)
                for (j = yi; j < yf; j++)
                    m[i][j] = k;


        for (i = 0; i < n ; i++) {
            for (j = 0; j < n; j++)
                if (j == (n-1))
                    printf ("  %d", m[i][j]);
                else
                    printf ("  %d ", m[i][j]);
            printf ("\n");
        }
        printf("\n");

    } while (n > 0);

    return 0;
}
```