



Lista de Exercícios 03a – Algoritmos – Repetição

- 1) Escreva um algoritmo em PORTUGOL que imprima todos os números inteiros de 0 a 50.

```
algoritmo L3P001;  
var  
  inteiro: C;  
início  
  para C de 0 até 50 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p001;  
var  
  C: integer;  
begin  
  for C := 0 to 50 do  
    writeln(C);  
end.
```

```
for C = 0 : 50  
  disp(C);  
end
```



- 2) Escreva um algoritmo em PORTUGOL que imprima todos os números inteiros do intervalo fechado de 1 a 100.

```
algoritmo L3P002;  
var  
  inteiro: C;  
início  
  para C de 1 até 100 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p002;  
var  
  C: integer;  
begin  
  for C := 1 to 100 do  
    writeln(C);  
  end.
```

```
for C = 1 : 100  
  disp(C);  
end
```



- 3) Escreva um algoritmo em PORTUGOL que imprima todos os números inteiros de 100 a 1 (em ordem decrescente).

```
algoritmo L3P003;  
var  
  inteiro: C;  
início  
  para C de 100 até 1 passo - 1 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p003;  
var  
  C: integer;  
begin  
  for C := 100 downto 1 do  
    writeln(C);  
  end.
```

```
for C = 100 : -1 : 1  
  disp(C);  
end
```



- 4) Escreva um algoritmo em PORTUGOL que imprima todos os números inteiros de 100 a 200.

```
algoritmo L3P004;  
var  
  inteiro: C;  
inicio  
  para C de 100 até 200 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p004;  
var  
  C: integer;  
begin  
  for C := 100 to 200 do  
    writeln(C);  
  end.
```

```
for C = 100 : 200  
  disp( C );  
end
```



- 5) Escreva um algoritmo em PORTUGOL que imprima todos os números inteiros de 200 a 100 (em ordem decrescente).

```
algoritmo L3P005;  
var  
  inteiro: C;  
inicio  
  para C de 200 até 100 passo - 1 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p005;  
var  
  C: integer;  
begin  
  for C := 200 downto 100 do  
    writeln(C);  
  end.
```

```
for C = 200 : -1 : 100  
  disp(C);  
end
```



- 6) Escreva um algoritmo em PORTUGOL que imprima todos os números múltiplos de 5, no intervalo fechado de 1 a 500.

```
algoritmo L3P006;  
var  
  inteiro: C;  
inicio  
  para C de 5 até 500 passo 5 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p006;  
var  
  C: integer;  
begin  
  for C := 5 to 500 do  
    if (C mod 5 = 0 ) then  
      writeln(C);  
  end.
```

```
for C = 5 : 500  
  if ( mod( C, 5 ) == 0 )  
    disp( C );  
  end  
end
```



- 7) Escreva um algoritmo em PORTUGOL que imprima todos os números pares do intervalo fechado de 1 a 100.

```
algoritmo L3P007;  
var  
  inteiro: C;  
início  
  para C de 2 até 100 passo 2 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p007;  
var  
  C: integer;  
begin  
  for C := 2 to 100 do  
    if ( C mod 2 = 0) then  
      writeln(C);  
  end.
```

```
for C = 2 : 100  
  if ( mod( C , 2 ) == 0)  
    disp( C );  
  end  
end
```



- 8) Escreva um algoritmo em PORTUGOL que imprima os 100 primeiros números ímpares.

```
algoritmo L3P008;  
var  
  inteiro: C;  
início  
  para C de 1 até 200 - 1 passo 2 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p008;  
var  
  C: integer;  
begin  
  for C := 1 to 200-1 do  
    if ( C mod 2 = 1 ) then  
      writeln(C);  
  end.
```

```
for C = 1 : 200-1  
  if ( mod( C , 2 ) == 1 )  
    disp( C );  
  end  
end
```




- 9) Escreva um algoritmo em PORTUGOL que imprima o quadrado dos números no intervalo fechado de 1 a 20.

```
algoritmo L3P009;  
var  
  inteiro: C;  
inicio  
  para C de 1 até 20 faça  
    imprima( C*C );  
  fim-para  
fim
```

```
program l3p009;  
var  
  C: integer;  
begin  
  for C := 1 to 20 do  
    writeln( C*C );  
  end.
```

```
for C = 1 : 20  
  disp( C*C );  
end
```



- 10) Escreva um algoritmo em PORTUGOL que imprima todos os números ímpares do intervalo fechado de 1 a 100.

```
algoritmo L3P010;  
var  
  inteiro: C;  
inicio  
  para C de 1 até 100 passo 2 faça  
    imprima(C);  
  fim-para  
fim
```

```
program l3p010;  
var  
  C: integer;  
begin  
  for C := 1 to 100 do  
    if ( C mod 2 = 1) then  
      writeln(C);  
  end.
```

```
for C = 1 : 100  
  if ( mod( C , 2 ) == 1 )  
    disp(C);  
  end  
end
```



- 11) Escreva um algoritmo em PORTUGOL que receba dez números do usuário e imprima a metade de cada número.

```
algoritmo L3P011;  
var  
  inteiro: C;  
  real: VALOR, METADE;  
inicio  
  para C de 1 até 10 faça  
    leia(VALOR);  
    METADE ← VALOR / 2;  
    imprima(METADE);  
  fim-para  
fim  
  
program l3p011;  
var  
  C: integer;  
  VALOR, METADE: real;  
begin  
  for C := 1 to 10 do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      METADE := VALOR / 2;  
      writeLn(METADE:3:2);  
    end;  
  end.  
  
for C = 1 : 10  
  VALOR = input('Digite um valor: ');  
  METADE = VALOR / 2;  
  disp( METADE );  
end
```



- 12) Escreva um algoritmo em PORTUGOL que receba dez números do usuário e imprima o quadrado de cada número.

algoritmo L3P012;

var

inteiro: C;

real: VALOR, QUAD;

início

para C **de** 1 **até** 10 **faça**

leia(VALOR);

 QUAD ← VALOR ** 2;

imprima(QUAD);

fim-para

fim

program l3p012;

var

 C: integer;

 VALOR, QUAD: real;

begin

 for C := 1 to 10 do

 begin

 write('Digite um valor: ');

 readLn(VALOR);

 QUAD := VALOR * VALOR;

 writeln(QUAD:3:2);

 end;

end.

for C = 1 : 10

 VALOR = input('Digite um valor: ');

 QUAD = VALOR * VALOR;

 disp(QUAD);

end



- 13) Escreva um algoritmo em PORTUGOL que receba dez números do usuário e imprima o cubo de cada número.

algoritmo L3P013;

var

inteiro: C;

real: VALOR, CUB;

início

para C **de** 1 **até** 10 **faça**

leia(VALOR);

 CUB ← VALOR ** 3;

imprima(CUB);

fim-para

fim

program l3p013;

var

 C: integer;

 VALOR, CUB: real;

begin

 for C := 1 to 10 do

 begin

 write('Digite um valor: ');

 readLn(VALOR);

 CUB := VALOR * VALOR * VALOR;

 writeln(CUB:3:2);

 end;

end.

for C = 1 : 10

 VALOR = input('Digite um valor: ');

 CUB = VALOR * VALOR * VALOR;

 disp(CUB);

end



- 14) Escreva um algoritmo em PORTUGOL que receba quinze números do usuário e imprima a raiz quadrada de cada número.

```
algoritmo L3P014;  
var  
  inteiro: C;  
  real: VALOR, RAIZ;  
inicio  
  para C de 1 até 15 faça  
    leia(VALOR);  
    se ( VALOR >= 0 ) então  
      RAIZ ← raiz(VALOR);  
      imprima(RAIZ);  
    senão  
      imprima("Não é possível calcular a raiz quadrada! Número negativo!");  
    fim-se  
  fim-para  
fim  
  
program l3p014;  
var  
  C: integer;  
  VALOR, RAIZ: real;  
begin  
  for C := 1 to 15 do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR >= 0 ) then begin  
        RAIZ := Sqrt(VALOR);  
        writeLn(RAIZ:3:2); end  
      else  
        writeLn('Nao eh possivel calcular a raiz quadrada! Numero negativo!');  
      end;  
    end;  
  end.  
  
for C = 1 : 15  
  VALOR = input('Digite um valor: ');  
  if ( VALOR >= 0 )  
    RAIZ = sqrt( VALOR );  
    disp( RAIZ );  
  else  
    disp('Nao eh possivel calcular a raiz quadrada! Numero negativo!');  
  end  
end
```



- 15) Escreva um algoritmo em PORTUGOL que receba oito números do usuário e imprima o logaritmo de cada um deles na base 10.

```
algoritmo L3P015;
var
  inteiro: C;
  real: VALOR, LOGX;
início
  para C de 1 até 8 faça
    leia(VALOR);
    se ( VALOR >= 0 ) então
      LOGX ← log(VALOR) / log(10);
      imprima(LOGX);
    senão
      imprima("Não é possível calcular o log na base dez! Número negativo!");
    fim-se
  fim-para
fim

program l3p015;
var
  C: integer;
  VALOR, LOGX: real;
begin
  for C := 1 to 8 do
    begin
      write('Digite um valor: ');
      readLn(VALOR);
      if ( VALOR > 0 ) then begin
        LOGX := Ln(VALOR) / Ln(10);
        writeLn(LOGX:3:2); end
      else
        writeLn('Nao eh possivel calcular o log na base dez! Numero negativo!');
      end;
    end;
  end.

for C = 1 : 8
  VALOR = input('Digite um valor: ');
  if ( VALOR > 0 )
    LOGX = log(VALOR) / log(10);
    disp(LOGX);
  else
    disp('Nao eh possivel calcular o log na base dez! Numero negativo!');
  end
end
```



- 16) Criar um algoritmo em PORTUGOL que imprima todos os números de 1 até 100, inclusive, e a soma de todos eles.

```
algoritmo L3P016;  
var  
  inteiro: C, SOMA;  
inicio  
  SOMA ← 0;  
  para C de 1 até 100 faça  
    imprima(C);  
    SOMA ← SOMA + C;  
  fim-para  
  imprima("O valor da soma é: ",SOMA);  
fim  
  
program l3p016;  
var  
  C,SOMA: integer;  
begin  
  SOMA := 0;  
  for C := 1 to 100 do  
    begin  
      writeln(C);  
      SOMA := SOMA + C;  
    end;  
  writeln('O valor da soma eh: ',SOMA);  
end.  
  
SOMA = 0;  
for C = 1 : 100  
  disp( C );  
  SOMA = SOMA + C;  
end  
fprintf(1,'O valor da soma eh: %d\n',SOMA);
```




- 17) Criar um algoritmo em PORTUGOL que imprima todos os números de 1 até 100, inclusive, e a soma do quadrado desses números.

```
algoritmo L3P017;  
var  
  inteiro: C, SOMA;  
inicio  
  SOMA ← 0;  
  para C de 1 até 100 faça  
    imprima(C);  
    SOMA ← SOMA + C ** 2;  
  fim-para  
  imprima("O valor da soma dos quadrados é: ", SOMA);  
fim  
  
program l3p017;  
var  
  C: integer;  
  SOMA: real;  
begin  
  SOMA := 0;  
  for C := 1 to 100 do  
    begin  
      writeln(C);  
      SOMA := SOMA + C * C;  
    end;  
  writeln('O valor da soma dos quadrados eh: ', SOMA:1:0);  
end.  
  
SOMA = 0;  
for C = 1 : 100  
  disp( C );  
  SOMA = SOMA + C * C;  
end  
fprintf(1, 'O valor da soma dos quadrados eh: %d\n', SOMA);
```



- 18) Criar um algoritmo em PORTUGOL que imprima todos os números de 1 até 100, inclusive, e a soma da metade desses números.

```
algoritmo L3P018;
var
  inteiro: C;
  real: SOMA;
inicio
  SOMA ← 0;
  para C de 1 até 100 faça
    imprima(C);
    SOMA ← SOMA + C / 2;
  fim-para
  imprima("O valor da soma das metades é: ",SOMA);
fim

program l3p018;
var
  C: integer;
  SOMA: real;
begin
  SOMA := 0;
  for C := 1 to 100 do
    begin
      writeln(C);
      SOMA := SOMA + C / 2;
    end;
  writeln('O valor da soma das metades eh: ',SOMA:3:2);
end.

SOMA = 0;
for C = 1 : 100
  disp( C );
  SOMA = SOMA + C / 2;
end
fprintf(1,'O valor da soma das metades eh: %f\n',SOMA);
```



- 19) Criar um algoritmo em PORTUGOL que imprima todos os números de 1 até 100, inclusive, e a soma do cubo desses números.

```
algoritmo L3P019;  
var  
  inteiro: C, SOMA;  
inicio  
  SOMA ← 0;  
  para C de 1 até 100 faça  
    imprima(C);  
    SOMA ← SOMA + C ** 3;  
  fim-para  
  imprima("O valor da soma dos cubos é: ",SOMA);  
fim  
  
program l3p019;  
var  
  C: integer;  
  SOMA: real; {estouro}  
begin  
  SOMA := 0;  
  for C := 1 to 100 do  
    begin  
      writeln(C);  
      SOMA := SOMA + Exp (3 * Ln(C) );  
    end;  
  writeln('O valor da soma dos cubos eh: ',SOMA:3:2);  
end.  
  
SOMA = 0;  
for C = 1 : 100  
  disp( C );  
  SOMA = SOMA + ( C * C * C );  
end  
fprintf(1,'O valor da soma dos cubos eh: %d\n',SOMA);
```



- 20) Criar um algoritmo em PORTUGOL que imprima todos os números de 1 até 100, inclusive, e a média de todos eles.

```
algoritmo L3P020;  
var  
  inteiro: C, SOMA;  
  real: MEDIA;  
inicio  
  SOMA ← 0;  
  para C de 1 até 100 faça  
    imprima(C);  
    SOMA ← SOMA + C;  
  fim-para  
  MEDIA ← SOMA / 100;  
  imprima("O valor da média é: ",MEDIA);  
fim  
  
program l3p020;  
var  
  C : integer;  
  SOMA,MEDIA: real;  
begin  
  SOMA := 0;  
  for C := 1 to 100 do  
    begin  
      writeln(C);  
      SOMA := SOMA + C;  
    end;  
  MEDIA := SOMA / 100;  
  writeln('O valor da media eh: ',MEDIA:3:2);  
end.  
  
SOMA = 0;  
for C = 1:100  
  disp( C );  
  SOMA = SOMA + C;  
end  
MEDIA = SOMA / 100;  
fprintf(1,'O valor da media eh: %f\n',MEDIA);
```



- 21) Criar um algoritmo em PORTUGOL que leia um número (NUM), e depois leia NUM números inteiros e imprima o maior deles. Suponha que todos os números lidos serão positivos.

```
algoritmo L3P021;  
var  
  inteiro: C, NUM, VALOR, MAIOR;  
inicio  
  leia(NUM);  
  MAIOR ← 0;  
  para C de 1 até NUM faça  
    leia(VALOR);  
    se ( VALOR > MAIOR ) então  
      MAIOR ← VALOR;  
    fim-se  
  fim-para  
  imprima("O maior valor é: ",MAIOR);  
fim
```

```
program l3p021;  
var  
  C, NUM, VALOR, MAIOR: integer;  
begin  
  write('Digite um valor: ');  
  readLn(NUM);  
  MAIOR := 0;  
  for C := 1 to NUM do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR > MAIOR ) then  
        MAIOR := VALOR;  
      end;  
    writeLn('O maior valor eh: ',MAIOR);  
  end.  
end.
```

```
NUM = input('Digite um valor: ');  
MAIOR = 0;  
for C = 1 : NUM  
  VALOR = input('Digite um valor: ');  
  if ( VALOR > MAIOR )  
    MAIOR = VALOR;  
  end  
end  
fprintf(1, 'O maior valor eh: %d\n', MAIOR);
```



22) Criar um algoritmo em PORTUGOL que leia um número (NUM), e depois leia NUM números inteiros e imprima o maior deles.

```
algoritmo L3P022;  
var  
  inteiro: C, NUM, VALOR, MAIOR;  
inicio  
  leia(NUM);  
  leia(VALOR);  
  MAIOR ← VALOR;  
  para C de 2 até NUM faça  
    leia(VALOR);  
    se ( VALOR > MAIOR ) então  
      MAIOR ← VALOR;  
  fim-se  
  fim-para  
  imprima("O maior valor é: ",MAIOR);  
fim  
  
program l3p022;  
var  
  C, NUM, VALOR, MAIOR: integer;  
begin  
  write('Digite um valor: ');  
  readLn(NUM);  
  write('Digite um valor: ');  
  readLn(VALOR);  
  MAIOR := VALOR;  
  for C := 2 to NUM do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR > MAIOR ) then  
        MAIOR := VALOR;  
    end;  
    writeLn('O maior valor eh: ',MAIOR);  
  end.  
  
NUM = input('Digite um valor: ');  
VALOR = input('Digite um valor: ');  
MAIOR = VALOR;  
for C = 2 : NUM  
  VALOR = input('Digite um valor: ');  
  if ( VALOR > MAIOR )  
    MAIOR = VALOR;  
end  
end  
fprintf(1,'O maior valor eh: %d\n',MAIOR);
```



23) Criar um algoritmo em PORTUGOL que leia um número (NUM), e depois leia NUM números inteiros e imprima o menor deles.

```
algoritmo L3P023;  
var  
  inteiro: C, NUM, VALOR, MENOR;  
inicio  
  leia(NUM);  
  leia(VALOR);  
  MENOR ← VALOR;  
  para C de 2 até NUM faça  
    leia(VALOR);  
    se ( VALOR < MENOR ) então  
      MENOR ← VALOR;  
  fim-se  
  fim-para  
  imprima("O menor valor é: ",MENOR);  
fim  
  
program l3p023;  
var  
  C, NUM, VALOR, MENOR: integer;  
begin  
  write('Digite um valor: ');  
  read(NUM);  
  write('Digite um valor: ');  
  read(VALOR);  
  MENOR := VALOR;  
  for C := 2 to NUM do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR < MENOR ) then  
        MENOR := VALOR;  
      end;  
      writeLn('O menor valor eh: ',MENOR);  
    end.  
  
  NUM = input('Digite um valor: ');  
  VALOR = input('Digite um valor: ');  
  MENOR = VALOR;  
  for C = 2 : NUM  
    VALOR = input('Digite um valor: ');  
    if ( VALOR < MENOR )  
      MENOR = VALOR;  
    end  
  end  
  fprintf(1,'O menor valor eh: %d\n',MENOR);
```



24) Criar um algoritmo em PORTUGOL que leia dez números inteiros e imprima o maior e o menor número da lista.

```
algoritmo L3P024;  
var  
  inteiro: C, VALOR, MAIOR, MENOR;  
início  
  leia(VALOR);  
  MENOR ← VALOR;  
  MAIOR ← VALOR;  
  para C de 2 até 10 faça  
    leia(VALOR);  
    se ( VALOR < MENOR ) então  
      MENOR ← VALOR;  
    senão  
      se ( VALOR > MAIOR ) então  
        MAIOR ← VALOR;  
    fim-se  
  fim-se  
  fim-para  
  imprima("O menor valor é: ", MENOR);  
  imprima("O maior valor é: ", MAIOR);  
fim
```

```
program l3p024;  
var  
  C, VALOR, MENOR, MAIOR: integer;  
begin  
  write('Digite um valor: ');  
  readLn(VALOR);  
  MENOR := VALOR;  
  MAIOR := VALOR;  
  for C := 2 to 10 do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR < MENOR ) then  
        MENOR := VALOR  
      else  
        if ( VALOR > MAIOR ) then  
          MAIOR := VALOR;  
        end;  
      writeLn('O menor valor eh: ', MENOR);  
      writeLn('O maior valor eh: ', MAIOR);  
    end;  
  end.
```

```
VALOR = input('Digite um valor: ');  
MENOR = VALOR;  
MAIOR = VALOR;  
for C = 2 : 10  
  VALOR = input('Digite um valor: ');  
  if ( VALOR < MENOR )  
    MENOR = VALOR;  
  else  
    if ( VALOR > MAIOR )  
      MAIOR = VALOR;  
    end  
  end  
end  
end  
fprintf(1, 'O menor valor eh: %d\n', MENOR);  
fprintf(1, 'O maior valor eh: %d\n', MAIOR);
```




25) Criar um algoritmo em PORTUGOL que leia dez números inteiros e imprima o maior e o segundo maior número da lista.

```
algoritmo L3P025;  
var  
  inteiro: C, VALOR, M1, M2;  
início  
  { M1 é o primeiro maior, e M2 é o segundo maior }  
  leia(VALOR);  
  M1 ← VALOR;  
  para C de 2 até 10 faça  
    leia(VALOR);  
    se ( VALOR > M1 ) então  
      M2 ← M1;  
      M1 ← VALOR;  
    senão  
      se ( VALOR > M2 ) ou ( C = 2 ) então  
        M2 ← VALOR;  
      fim-se  
    fim-se  
  fim-para  
  imprima("O primeiro maior é: ",M1);  
  imprima("O segundo maior é: ",M2);  
fim
```

```
program l3p025;  
var  
  C, VALOR, M1, M2: integer;  
begin  
  { M1 eh o primeiro maior, e M2 eh o segundo maior }  
  write('Digite um valor: ');  
  readLn(VALOR);  
  M1 := VALOR;  
  for C := 2 to 10 do  
    begin  
      write('Digite um valor: ');  
      readLn(VALOR);  
      if ( VALOR > M1 ) then begin  
        M2 := M1;  
        M1 := VALOR; end  
      else  
        if ( VALOR > M2 ) or ( C = 2 ) then  
          M2 := VALOR;  
        end;  
      writeLn('O primeiro maior eh: ',M1);  
      writeLn('O segundo maior eh: ',M2);  
    end.  
end.
```

```
% M1 eh o primeiro maior, e M2 eh o segundo maior }  
VALOR = input('Digite um valor: ');  
M1 = VALOR;  
for C = 2 : 10  
  VALOR = input('Digite um valor: ');  
  if ( VALOR > M1 )  
    M2 = M1;  
    M1 = VALOR;  
  else  
    if ( VALOR > M2 ) | ( C == 2 )  
      M2 = VALOR;  
    end  
  end  
end  
fprintf(1,'O primeiro maior eh: %d\n',M1);  
fprintf(1,'O segundo maior eh: %d\n',M2);
```



- 26) Criar um algoritmo em PORTUGOL que leia os limites inferior e superior de um intervalo e imprima todos os números pares no intervalo aberto e seu somatório. Suponha que os dados digitados são para um intervalo crescente, ou seja, o primeiro valor é menor que o segundo.

```
algoritmo L3P026;  
var  
  inteiro: INF, SUP, C, SOMA;  
início  
  leia(INF, SUP);  
  SOMA ← 0;  
  para C de INF+1 até SUP-1 faça  
    se ( C mod 2 = 0 ) então  
      imprima( C );  
      SOMA ← SOMA + C;  
    fim-se  
  fim-para  
  imprima("O valor da soma é: ",SOMA);  
fim
```

```
program l3p026;  
var  
  INF, SUP, C, SOMA: integer;  
begin  
  write('Entre com o limite Inferior: ');  
  readLn(INF);  
  write('Entre com o limite Superior: ');  
  readLn(SUP);  
  SOMA := 0;  
  for C := INF+1 to SUP-1 do  
    begin  
      if ( C mod 2 = 0 ) then begin  
        writeLn( C );  
        SOMA := SOMA + C;  
      end;  
    end;  
  writeLn('O valor da soma eh: ',SOMA);  
end.
```

```
INF = input('Entre com o limite Inferior: ');  
SUP = input('Entre com o limite Superior: ');  
SOMA = 0;  
for C = INF+1 : SUP-1  
  if ( mod( C , 2 ) == 0 )  
    disp( C );  
    SOMA = SOMA + C;  
  end  
end  
fprintf(1,'O valor da soma eh: %d\n',SOMA);
```



27) Criar um algoritmo em PORTUGOL que leia um número (NUM) e então imprima os múltiplos de 3 e 5, ao mesmo tempo, no intervalo fechado de 1 a NUM.

```
algoritmo L3P027;  
var  
  inteiro: NUM,C;  
inicio  
  leia(NUM);  
  para C de 1 até NUM faça  
    se ( C mod 3 = 0 ) e ( C mod 5 = 0 ) então  
      imprima( C );  
    fim-se  
  fim-para  
fim  
  
program l3p027;  
var  
  NUM,C: integer;  
begin  
  write('Digite um numero: ');  
  readLn(NUM);  
  for C := 1 to NUM do  
    begin  
      if ( C mod 3 = 0 ) and ( C mod 5 = 0 ) then  
        writeLn( C );  
    end;  
  end.  
  
NUM = input('Digite um numero: ');  
for C = 1 : NUM  
  if ( mod ( C , 3 ) == 0 ) & ( mod ( C , 5 ) == 0 )  
    disp( C );  
  end  
end
```



28) Escreva um algoritmo em PORTUGOL que leia 200 números inteiros e imprima quantos são pares e quantos são ímpares.

```
algoritmo L3P028;  
var  
  inteiro: C, NPAR, NIMPARG;  
  real: VALOR;  
início  
  NPAR ← 0;  
  NIMPARG ← 0;  
  para C de 1 até 200 faça  
    leia(VALOR);  
    se ( C mod 2 = 0 ) então  
      NPAR ← NPAR + 1;  
    senão  
      NIMPARG ← NIMPARG + 1;  
    fim-se  
  fim-para  
  imprima("A quantidade de números pares digitados: ", NPAR );  
  imprima("A quantidade de números ímpares digitados: ", NIMPARG);  
fim
```

```
program l3p028;  
var  
  C, NPAR, NIMPARG, VALOR: integer;  
begin  
  NPAR := 0;  
  NIMPARG := 0;  
  for C := 1 to 200 do  
    begin  
      write('Digite um numero: ');  
      readln(VALOR);  
      if ( VALOR mod 2 = 0 ) then  
        NPAR := NPAR + 1  
      else  
        NIMPARG := NIMPARG + 1;  
      end;  
      writeln('A quantidade de numeros pares digitados: ', NPAR );  
      writeln('A quantidade de numeros impares digitados: ', NIMPARG);  
    end.  
end.
```

```
NPAR = 0;  
NIMPARG = 0;  
for C = 1 : 200  
  VALOR = input('Digite um numero: ');  
  if ( mod( VALOR , 2 ) == 0 )  
    NPAR = NPAR + 1;  
  else  
    NIMPARG = NIMPARG + 1;  
  end  
end  
fprintf(1,'A quantidade de numeros pares digitados:  %d\n',NPAR );  
fprintf(1,'A quantidade de numeros impares digitados: %d\n',NIMPARG);
```



29) Escreva um algoritmo em PORTUGOL que receba 15 números e imprima quantos números maiores que 30 foram digitados.

```
algoritmo L3P029;
var
  inteiro: C, N30;
  real: VALOR;
início
  N30 ← 0;
  para C de 1 até 15 faça
    leia(VALOR);
    se ( VALOR > 30 ) então
      N30 ← N30 + 1;
    fim-se
  fim-para
  imprima("A quantidade de números maiores que 30 digitados: ",N30);
fim

program l3p029;
var
  C, N30: integer;
  VALOR: real;
begin
  N30 := 0;
  for C := 1 to 15 do
    begin
      write('Digite um numero: ');
      readLn(VALOR);
      if ( VALOR > 30 )
        then N30 := N30 + 1;
      end;
      writeLn('A quantidade de numeros maiores que 30 digitados: ',N30);
    end.

N30 = 0;
for C = 1 : 15
  VALOR = input('Digite um numero: ');
  if ( VALOR > 30 )
    N30 = N30 + 1;
  end
end
fprintf(1,'A quantidade de numeros maiores que 30 digitados: %d\n',N30);
```



30) Escreva um algoritmo em PORTUGOL que leia 20 números e imprima a soma dos positivos e o total de números negativos.

```
algoritmo L3P030;
var
  inteiro: C, NNEG;
  real: VALOR, SOMAP;
início
  SOMAP ← 0;
  NNEG ← 0;
  para C de 1 até 20 faça
    leia(VALOR);
    se ( VALOR >= 0 ) então
      SOMAP ← SOMAP + VALOR; { número positivo }
    senão
      NNEG ← NNEG + 1; { número negativo }
    fim-se
  fim-para
  imprima("A soma dos número positivos digitados é: ",SOMAP);
  imprima("A quantidade de números negativos digitados é: ",NNEG);
fim

program l3p030;
var
  C, NNEG: integer;
  VALOR, SOMAP: real;
begin
  SOMAP := 0;
  NNEG := 0;
  for C := 1 to 20 do
    begin
      write('Digite um numero: ');
      readLn(VALOR);
      if ( VALOR >= 0 ) then
        SOMAP := SOMAP + VALOR { numero positivo }
      else
        NNEG := NNEG + 1; { numero negativo }
      end;
      writeLn('A soma dos numero positivos digitados eh: ',SOMAP:3:2);
      writeLn('A quantidade de numeros negativos digitados eh: ',NNEG);
    end.

  SOMAP = 0;
  NNEG = 0;
  for C = 1 : 20
    VALOR = input('Digite um numero: ');
    if ( VALOR >= 0 )
      SOMAP = SOMAP + VALOR; % numero positivo
    else
      NNEG = NNEG + 1; % numero negativo
    end
  end
  fprintf(1,'A soma dos numero positivos digitados eh: %f\n',SOMAP);
  fprintf(1,'A quantidade de numeros negativos digitados eh: %d\n',NNEG);
```



- 31) Escreva um algoritmo em PORTUGOL que realize o produto de A (número real) por B (número inteiro), ou seja, $A * B$, através de adições (somas). Esses dois valores são passados pelo usuário através do teclado.

```
algoritmo L3P031;  
var  
  inteiro: B, C;  
  real: A, MULT;  
início  
  MULT  $\leftarrow$  0;  
  leia (A,B);  
  para C de 1 até B faça  
    MULT  $\leftarrow$  MULT + A;  
  fim-para  
  imprima ("O produto de A por B é: ",MULT);  
fim  
  
program l3p031;  
var  
  B, C: integer;  
  A, MULT: real;  
begin  
  write('Entre com valor de A: ');  
  readLn(A);  
  write('Entre com valor de B: ');  
  readLn(B);  
  MULT := 0;  
  for C := 1 to B do  
    MULT := MULT + A;  
  writeLn('O produto de A por B eh: ',MULT:3:2);  
end.  
  
% B, C: integer;  
% A, MULT: real;  
A = input('Entre com valor de A: ');  
B = input('Entre com valor de B: ');  
MULT = 0;  
for C = 1 : B  
  MULT = MULT + A;  
end  
fprintf(1,'O produto de A por B eh: %d\n',MULT);
```



- 32) Escreva um algoritmo em PORTUGOL que realize a potência de A (número real) por B (número inteiro e positivo), ou seja, A^B , através de multiplicações sucessivas. Esses dois valores são passados pelo usuário através do teclado.

```
algoritmo L3P032;  
var  
  inteiro: B, C;  
  real: A, POT;  
inicio  
  POT ← 1;  
  leia (A,B);  
  para C de 1 até B faça  
    POT ← POT * A;  
  fim-para  
  imprima ("A potência de A por B é: ",POT);  
fim  
  
program l3p032;  
var  
  B, C: integer;  
  A, POT: real;  
begin  
  write('Entre com valor de A: ');  
  readLn(A);  
  write('Entre com valor de B: ');  
  readLn(B);  
  POT := 1;  
  for C := 1 to B do  
    POT := POT * A;  
  writeLn('A potencia de A por B eh: ',POT:3:2);  
end.  
  
% B, C: integer;  
% A, POT: real;  
A = input('Entre com valor de A: ');  
B = input('Entre com valor de B: ');  
POT = 1;  
for C = 1 : B  
  POT = POT * A;  
end  
fprintf(1,'A potencia de A por B eh: %d\n',POT);
```




- 33) Escreva um algoritmo em PORTUGOL que calcule o resto da divisão de A por B (número inteiros e positivos), ou seja, A **mod** B, através de subtrações sucessivas. Esses dois valores são passados pelo usuário através do teclado.

```
algoritmo L3P033;  
var  
  inteiro: A, B, MOD;  
inicio  
  leia (A,B) ;  
  MOD ← A;  
  enquanto ( MOD >= B ) faça  
    MOD ← MOD - B;  
  fim-enquanto  
  imprima("o resto da divisão de A por B (A mod B) é: ",MOD);  
fim  
  
program l3p033;  
var  
  A, B, MD: integer;  
begin  
  write('Entre com valor de A: ');  
  readLn(A);  
  write('Entre com valor de B: ');  
  readLn(B);  
  MD := A;  
  while ( MD >= B ) do  
    MD := MD - B;  
  writeLn('o resto da divisao de A por B (A mod B) eh: ',MD);  
end.  
  
% A, B, MD: integer;  
A = input('Entre com valor de A: ');  
B = input('Entre com valor de B: ');  
MD = A;  
while ( MD >= B )  
  MD = MD - B;  
end  
fprintf(1,'o resto da divisao de A por B (A mod B) eh: %d\n',MD);
```



- 34) Escreva um algoritmo em PORTUGOL que calcule o quociente da divisão de A por B (número inteiros e positivos), ou seja, A **div** B, através de subtrações sucessivas. Esses dois valores são passados pelo usuário através do teclado.

```
algoritmo L3P034;  
var  
  inteiro: A, B, DV, MD;  
inicio  
  leia (A,B);  
  MD ← A;  
  DV ← 0;  
  enquanto (MD >= B) faça  
    DV ← DV + 1;  
    MD ← MD - B;  
  fim-enquanto  
  imprima("O quociente da divisão de A por B (A div B) é: ",DV);  
fim  
  
program l3p034;  
var  
  A, B, DV, MD: integer;  
begin  
  write('Entre com valor de A: ');  
  readLn(A);  
  write('Entre com valor de B: ');  
  readLn(B);  
  MD := A;  
  DV := 0;  
  while ( MD >= B ) do  
    begin  
      DV := DV + 1;  
      MD := MD - B;  
    end;  
  writeLn('O quociente da divisao de A por B (A div B) eh: ',DV);  
end.  
  
A = input('Entre com valor de A: ');  
B = input('Entre com valor de B: ');  
MD = A;  
DV = 0;  
while ( MD >= B )  
  DV = DV + 1;  
  MD = MD - B;  
end  
fprintf(1,'O quociente da divisao de A por B (A div B) eh: %d\n',DV);
```



- 35) Escreva um algoritmo em PORTUGOL que determine se dois valores inteiros e positivos A e B são **primos** entre si. (dois números inteiros são ditos primos entre si, caso não exista divisor comum aos dois números).

```
algoritmo L3P035;  
var  
  inteiro: A, B, C;  
início  
  lógico: SIT;  
  SIT ← verdadeiro;  
  leia (A,B);  
  MOD ← A;  
  C ← 1;  
  enquanto ( SIT e C <= A e C <= B ) faça  
    se (A mod C = 0) e (B mod C = 0)  
      então SIT ← falso;  
    fim-se  
    C ← C + 1;  
  fim-enquanto  
  se ( SIT ) então  
    imprima ("A e B são primos entre si ");  
  senão  
    imprima ("A e B não são primos entre si");  
  fim-se  
fim
```

```
program l3p035;  
var  
  A, B, C: integer;  
  SIT: boolean;  
begin  
  SIT := true;  
  write('Entre com valor de A: ');  
  readLn(A);  
  write('Entre com valor de B: ');  
  readLn(B);  
  C := 2;  
  while ( SIT ) and ( C <= A ) and ( C <= B ) do  
    begin  
      if (A mod C = 0) and (B mod C = 0) then  
        SIT := false;  
      C := C + 1;  
    end;  
  if ( SIT ) then  
    writeLn('A e B sao primos entre si ')  
  else  
    writeLn('A e B nao sao primos entre si');  
end.
```

```
SIT = 1; % true  
A = input('Entre com valor de A: ');  
B = input('Entre com valor de B: ');  
C = 2;  
while ( SIT ) & ( C <= A ) & ( C <= B )  
  if ( mod(A,C) == 0 ) & (mod(B,C) == 0)  
    SIT = 0; % false  
  end  
  C = C + 1;  
end  
if ( SIT )  
  disp('A e B sao primos entre si ');  
else  
  disp('A e B nao sao primos entre si');  
end
```



36) Escreva um algoritmo em PORTUGOL para calcular o fatorial do número N, cujo valor é obtido através do usuário pelo teclado.

```
algoritmo L3P036;  
var  
  inteiro: N, FAT, C;  
inicio  
  FAT ← 1;  
  leia(N);  
  para C de 2 até N faça  
    FAT ← FAT * C;  
  fim-para  
  imprima("O valor do fatorial de N é: ",FAT);  
fim  
  
program l3p036;  
var  
  N, C: integer;  
  FAT: real;  
begin  
  write('Digite um numero: ');  
  readLn(N);  
  FAT := 1;  
  for C := 2 to N do  
    FAT := FAT * C;  
  writeLn('O valor do fatorial de N eh: ',FAT:2:0 );  
end.  
  
% N, C: integer;  
% FAT: real;  
N = input('Digite um numero: ');  
FAT = 1;  
for C = 2 : N  
  FAT = FAT * C;  
end  
fprintf(1,'O valor do fatorial de N eh: %d\n',FAT);
```



37) Escreva um algoritmo em PORTUGOL que determine todos os divisores de um dado número N.

```
algoritmo L3P037;  
var  
  inteiro: N, C;  
inicio  
  leia(N);  
  para C de 1 até N faça  
    se (N mod C = 0 ) então  
      imprima(C, " é divisor de ",N);  
    fim-se  
  fim-para  
fim  
  
program l3p037;  
var  
  N, C: integer;  
begin  
  write('Digite um numero: ');  
  readLn(N);  
  for C := 1 to N do  
    if (N mod C = 0 ) then  
      writeLn(C, ' eh divisor de ',N);  
  end.  
  
% N, C: integer;  
N = input('Digite um numero: ');  
for C = 1 : N  
  if ( mod( N , C ) == 0 )  
    fprintf(1, '%d eh divisor de %d\n',C,N);  
  end  
end
```



38) Escreva um algoritmo em PORTUGOL que determine se um dado número N (digitado pelo usuário) é primo ou não.

```
algoritmo L3P038;  
var  
  inteiro: N, C;  
  lógico: SIT;  
início  
  SIT ← verdadeiro;  
  leia(N);  
  C ← 2;  
  enquanto ( SIT e C < N ) faça  
    se (N mod C = 0 ) então  
      SIT ← falso;  
    fim-se  
    C ← C + 1;  
  fim-enquanto  
  se ( SIT ) então  
    imprima(N, " é primo!");  
  senão  
    imprima(N, " não é primo!");  
  fim-se  
fim
```

```
program l3p038;  
var  
  N, C: integer;  
  SIT: boolean;  
begin  
  write('Digite um numero: ');  
  read(N);  
  SIT := true;  
  C := 2;  
  while ( SIT ) and ( C < N ) do  
    begin  
      if (N mod C = 0 ) then  
        SIT := false;  
        C := C + 1;  
      end;  
      if ( SIT ) then  
        writeln(N, ' eh primo!')  
      else  
        writeln(N, ' nao eh primo!');  
    end.  
end.
```

```
% N, C: integer;  
% SIT: boolean;  
N = input('Digite um numero: ');  
SIT = 1; % true  
C = 2;  
while ( SIT ) & ( C < N )  
  if (mod( N , C ) == 0 )  
    SIT = 0; % false  
  end  
  C = C + 1;  
end  
if ( SIT )  
  fprintf(1, '%d eh primo!', N);  
else  
  fprintf(1, '%d nao eh primo!', N);  
end
```



- 39) Escreva um algoritmo em PORTUGOL que calcule os N-menores números primos.
Este número N deve ser lido do teclado.

```
algoritmo L3P039;  
var  
  inteiro: N, I, J, K;  
  lógico: SIT;  
início  
  leia(N);  
  K ← 2; { número primo candidato }  
  para I de 1 até N faça  
    SIT ← verdadeiro;  
    J ← 2;  
    enquanto ( SIT e J < K ) faça  
      se (K mod J = 0 ) então  
        SIT ← falso;  
      fim-se  
      J ← J + 1;  
    fim-enquanto  
    se ( SIT ) então  
      imprima(I, "-ésimo número primo é: ",K);  
    senão  
      I ← I + 1;  
    fim-se  
    K ← K + 1;  
  fim-para  
fim
```

```
program l3p039;  
var  
  N, I, J, K: integer ;  
  SIT: boolean;  
begin  
  write('Digite um numero: ');  
  read(N);  
  K := 2; { numero primo candidato }  
  I := 1;  
  while (I <= N) do  
    begin  
      SIT := true;  
      J := 2;  
      while ( SIT ) and ( J < K ) do  
        begin  
          if (K mod J = 0 ) then  
            SIT := false;  
          J := J + 1;  
        end;  
      if ( SIT ) then begin  
        writeln(I, '-esimo numero primo eh: ',K);  
        I := I + 1; end;  
      K := K + 1;  
    end;  
  end.  
end.
```



```
% N, I, J, K: integer ;
% SIT: boolean;
N = input('Digite um numero: ');
K = 2; % numero primo candidato
I = 1;
while (I <= N)
    SIT = 1; % true
    J = 2;
    while ( SIT ) & ( J < K )
        if ( mod( K , J ) == 0 )
            SIT = 0; % false
        end
        J = J + 1;
    end
    if ( SIT )
        fprintf(1, '%d-esimo numero primo eh: %d\n', I, K);
        I = I + 1;
    end
    K = K + 1;
end
```




- 40) Escreva um algoritmo em PORTUGOL que calcule o m.d.c. (máximo divisor comum) entre A e B (número inteiros e positivos). Esses dois valores são passados pelo usuário através do teclado.

```
algoritmo L3P040;  
var  
  inteiro: A, B, C, MDC;  
inicio  
  leia (A,B);  
  C ← 1;  
  enquanto ( C <= A e C <= B ) faça  
    se (A mod C = 0) e (B mod C = 0) então  
      MDC ← C;  
    fim-se  
    C ← C + 1;  
  fim-enquanto  
  imprima ("O m.d.c de A e B (m.d.c.(A,B)) é: ",MDC);  
fim
```

```
program l3p040;  
var  
  A, B, C, MDC: integer;  
begin  
  write('Digite um numero: ');  
  readLn(A);  
  write('Digite outro numero: ');  
  readLn(B);  
  C := 1;  
  while ( C <= A ) and ( C <= B ) do  
    begin  
      if (A mod C = 0) and (B mod C = 0) then  
        MDC := C;  
      C := C + 1;  
    end;  
  writeLn('O m.d.c. de A e B (m.d.c.(A,B)) eh: ',MDC);  
end.
```

```
A = input('Digite um numero: ');  
B = input('Digite outro numero: ');  
C = 1;  
while ( C <= A ) & ( C <= B )  
  if ( mod(A,C) == 0 ) & ( mod(B,C) == 0 )  
    MDC = C;  
  end  
  C = C + 1;  
end  
fprintf(1,'O m.d.c. de A e B (m.d.c.(A,B)) eh: %d\n',MDC);
```



41) A série de Fibonacci é formada pela sequência:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Escreva um algoritmo em PORTUGOL que gere a série de FIBONACCI até o N-ésimo termo.

```
algoritmo L3P041;  
var  
  inteiro: N, C, ATUAL, ANT1, ANT2;  
inicio  
  leia (N);  
  ANT2 ← 1;  
  ANT1 ← 1;  
  imprima (ANT2);  
  imprima (ANT1);  
  para C de 3 até N faça  
    ATUAL ← ANT1 + ANT2;  
    imprima (ATUAL);  
    ANT2 ← ANT1;  
    ANT1 ← ATUAL;  
  fim-para  
fim
```

```
program l3p041;  
var  
  N, C, ATUAL, ANT1, ANT2: integer;  
begin  
  write('Digite o numero de termos: ');  
  read(N);  
  ANT2 := 1;  
  ANT1 := 1;  
  writeln(ANT2);  
  writeln(ANT1);  
  for C := 3 to N do  
    begin  
      ATUAL := ANT1 + ANT2;  
      writeln(ATUAL);  
      ANT2 := ANT1;  
      ANT1 := ATUAL;  
    end;  
end.  
  
% N, C, ATUAL, ANT1, ANT2: integer;  
N = input('Digite o numero de termos: ');  
ANT2 = 1;  
ANT1 = 1;  
disp(ANT2);  
disp(ANT1);  
for C = 3 : N  
  ATUAL = ANT1 + ANT2;  
  disp(ATUAL);  
  ANT2 = ANT1;  
  ANT1 = ATUAL;  
end
```



- 42) A série de RICCI difere da série de FIBONACCI porque os dois primeiros termos são fornecidos pelo usuário. Os demais termos são gerados da mesma forma que a série de FIBONACCI. Criar um algoritmo em PORTUGOL que imprima os N primeiros termos da série de RICCI e a soma dos termos impressos, sabendo-se que para existir esta série serão necessários pelo menos três termos.

```
algoritmo L3P042;
var
  inteiro: N, C, ATUAL, ANT1, ANT2, SOMA;
início
  leia(N);
  leia(ANT2);
  leia(ANT1);
  imprima(ANT2);
  imprima(ANT1);
  SOMA ← ANT1 + ANT2;
  para C de 3 até N faça
    ATUAL ← ANT1 + ANT2;
    imprima(ATUAL);
    SOMA ← SOMA + ATUAL;
    ANT2 ← ANT1;
    ANT1 ← ATUAL;
  fim-para
  imprima("A soma dos ",N," termos é: ",SOMA);
fim

program l3p042;
var
  N, C, ATUAL, ANT1, ANT2, SOMA: integer;
begin
  write('Digite o numero de termos: ');
  readLn(N);
  write('Digite o primeiro termo: ');
  readLn(ANT2);
  write('Digite o segundo termo: ');
  readLn(ANT1);
  writeLn(ANT2);
  writeLn(ANT1);
  SOMA := ANT1 + ANT2;
  for C := 3 to N do
  begin
    ATUAL := ANT1 + ANT2;
    writeLn(ATUAL);
    SOMA := SOMA + ATUAL;
    ANT2 := ANT1;
    ANT1 := ATUAL;
  end;
  writeLn('A soma dos ',N,' termos eh: ',SOMA);
end.

% N, C, ATUAL, ANT1, ANT2, SOMA: integer;
N = input('Digite o numero de termos: ');
ANT2 = input('Digite o primeiro termo: ');
ANT1 = input('Digite o segundo termo: ');
disp(ANT2);
disp(ANT1);
SOMA = ANT1 + ANT2;
for C = 3 : N
  ATUAL = ANT1 + ANT2;
  disp(ATUAL);
  SOMA = SOMA + ATUAL;
  ANT2 = ANT1;
  ANT1 = ATUAL;
end
fprintf(1,'A soma dos %d termos eh: %d\n',N,SOMA);
```

- 43) A série de FETUCCINE é gerada da seguinte forma: os dois primeiros termos são fornecidos pelo usuário; a partir daí, os termos são gerados com a soma ou subtração dos dois termos anteriores, ou seja:

$$A_i = A_{i-1} + A_{i-2} \text{ para } i \text{ ímpar}$$

$$A_i = A_{i-1} - A_{i-2} \text{ para } i \text{ par}$$

Criar um algoritmo em PORTUGOL que imprima os N primeiros termos da série de FETUCCINE, sabendo-se que para existir esta série serão necessários pelo menos três termos.

```
algoritmo L3P043;  
var  
  inteiro: N, C, ATUAL, ANT1, ANT2;  
início  
  leia(N);  
  leia(ANT2);  
  leia(ANT1);  
  imprima(ANT2);  
  imprima(ANT1);  
  para C de 3 até N faça  
    se ( C mod 2 = 1 )  
      então ATUAL ← ANT1 + ANT2; { é ímpar }  
      senão ATUAL ← ANT1 - ANT2; { é par }  
    fim-se  
    imprima(ATUAL);  
    ANT2 ← ANT1;  
    ANT1 ← ATUAL;  
  fim-para  
fim
```

```
program l3p043;  
var  
  N, C, ATUAL, ANT1, ANT2: integer;  
begin  
  write('Digite o numero de termos: ');  
  readLn(N);  
  write('Digite o primeiro termo: ');  
  readLn(ANT2);  
  write('Digite o segundo termo: ');  
  readLn(ANT1);  
  writeLn(ANT2);  
  writeLn(ANT1);  
  for C := 3 to N do  
    begin  
      if ( C mod 2 = 1 ) then  
        ATUAL := ANT1 + ANT2 { eh impar }  
      else  
        ATUAL := ANT1 - ANT2; { eh par }  
      writeLn(ATUAL);  
      ANT2 := ANT1;  
      ANT1 := ATUAL;  
    end;  
  end.  
end.
```



```
% N, C, ATUAL, ANT1, ANT2: integer;
N = input('Digite o numero de termos: ');
ANT2 = input('Digite o primeiro termo: ');
ANT1 = input('Digite o segundo termo: ');
disp(ANT2);
disp(ANT1);
for C = 3 : N
    if ( mod(C,2) == 1 )
        ATUAL = ANT1 + ANT2; % eh impar
    else
        ATUAL = ANT1 - ANT2; % eh par
    end
    disp(ATUAL);
    ANT2 = ANT1;
    ANT1 = ATUAL;
end
```



44) Seja a seguinte série:

1, 4, 9, 16, 25, 36, ...

Escreva um algoritmo em PORTUGOL que gere esta série até o N-ésimo termo. Este N-ésimo termo é digitado pelo usuário.

algoritmo L3P044;

var

inteiro: N, C, QUAD;

início

leia(N);

para C **de** 1 **até** N **faça**

QUAD \leftarrow C * C;

imprima(QUAD);

fim-para

fim

program l3p044;

var

N, C, QUAD: integer;

begin

write('Digite o numero de termos: ');

read(N);

for C := 1 to N do

begin

QUAD := C * C;

writeln(QUAD);

end;

end.

% N, C, QUAD: integer;

N = input('Digite o numero de termos: ');

for C = 1 : N

QUAD = C * C;

disp(QUAD);

end



45) Seja a seguinte série:

1, 4, 4, 2, 5, 5, 3, 6, 6, 4, 7, 7, ...

Escreva um algoritmo em PORTUGOL que seja capaz de gerar os N termos dessa série. Esse número N deve ser lido do teclado.

```
algoritmo L3P045;
var
  inteiro: N, C;
início
  leia(N);
  para C de 1 até N faça
    se ( C mod 3 = 1 ) então
      imprima( ( C div 3 ) + 1 ); { 1°, 4°, 7° , ... }
    senão-se ( C mod 3 = 2 ) então
      imprima( ( C div 3 ) + 4 ); { 2°, 5°, 8° , ... }
    senão
      imprima( ( C div 3 ) + 3 ); { 3°, 6°, 9° , ... }
    fim-se
  fim-para
fim

program l3p045;
var
  N, C: integer;
begin
  write('Digite o numero de termos: ');
  read(N);
  for C := 1 to N do
    begin
      if ( C mod 3 = 1 ) then
        writeln( ( C div 3 ) + 1 )      { 1o, 4o, 7o , ... }
      else if ( C mod 3 = 2 ) then
        writeln( ( C div 3 ) + 4 )    { 2o, 5o, 8o , ... }
      else
        writeln( ( C div 3 ) + 3 ); { 3o, 6o, 9o , ... }
      end;
    end;
  end.

N = input('Digite o numero de termos: ');
for C = 1 : N
  if ( mod(C,3) == 1 )
    fprintf(1,'%d\n', floor( C / 3 ) + 1 ) % 1o, 4o, 7o , ...
  elseif ( mod(C,3) == 2 )
    fprintf(1,'%d\n', floor( C / 3 ) + 4 ); % 2o, 5o, 8o , ...
  else
    fprintf(1,'%d\n', floor( C / 3 ) + 3 ); % 3o, 6o, 9o , ...
  end
end
end
```



46) Sendo $H = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$, faça um algoritmo em PORTUGOL para gerar o número H. O número N é lido do teclado.

algoritmo L3P046;

var

inteiro: N, C;

real: H;

início

H ← 0;

leia(N);

para C **de** 1 **até** N **faça**

H ← H + 1 / C;

fim-para

imprima("H = ",H);

fim

program l3p046;

var

N, C: integer;

H: real;

begin

write('Digite um numero: ');

readLn(N);

H := 0;

for C := 1 to N do

H := H + 1 / C;

writeln('H = ',H:5:4);

end.

% N, C: integer;

% H: real;

N = input('Digite um numero: ');

H = 0;

for C = 1 : N

H = H + 1 / C;

end

fprintf(1,'H = %f\n',H);



47) Sendo $H = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{N}$, faça um algoritmo em PORTUGOL para gerar o número H. O número N é lido do teclado.

algoritmo L3P047;

var

inteiro: N, C;

real: H;

início

H ← 0;

leia (N);

para C **de** 1 **até** N **faça**

se (C mod 2 = 1) **então**

H ← H + 1 / C; { termo ímpar }

senão

H ← H - 1 / C; { termo par }

fim-se

fim-para

imprima ("H = ",H);

fim

program l3p047;

var

N, C: integer;

H: real;

begin

write('Digite um numero: ');

readLn(N);

H := 0;

for C := 1 to N do

begin

if (C mod 2 = 1) then

H := H + 1 / C { termo ímpar }

else

H := H - 1 / C; { termo par }

end;

writeln('H = ',H:5:4);

end.

% N, C: integer;

% H: real;

N = input('Digite um numero: ');

H = 0;

for C = 1 : N

if (mod(C,2) == 1)

H = H + 1 / C; % termo ímpar

else

H = H - 1 / C; % termo par

end

end

fprintf(1,'H = %f\n',H);



48) Faça um algoritmo em PORTUGOL para calcular o valor de S, dado por:

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

sendo N fornecido pelo teclado.

algoritmo L3P048;

var

inteiro: N, C;

real: S;

início

 S ← 0;

leia(N);

para C **de** 1 **até** N **faça**

 S ← S + C / (N - C + 1);

fim-para

imprima("S = ",S);

fim

program l3p048;

var

 N, C: integer;

 S: real;

begin

 write('Digite um numero: ');

 read(N);

 S := 0;

 for C := 1 to N do

 S := S + C / (N - C + 1);

 writeln('S = ',S:5:4);

end.

%N, C: integer;

%S: real;

N = input('Digite um numero: ');

S = 0;

for C = 1 : N

 S = S + C / (N - C + 1);

end

fprintf(1,'S = %f\n',S);



49) Sendo $S = 1 + \frac{1}{2^2} + \frac{1}{3^3} + \frac{1}{4^4} + \frac{1}{5^5} + \dots + \frac{1}{N^N}$, um somatório de N (definido pelo usuário) termos, escreva algoritmo em PORTUGOL para calcular S para um número N.

algoritmo L3P049;

var

inteiro: N, C;

real: S;

início

S ← 0;

leia(N);

para C **de** 1 **até** N **faça**

S ← S + 1 / (C ** C);

fim-para

imprima("S = ",S);

fim

program l3p049;

var

N, C: integer;

S: real;

begin

write('Digite um numero: ');

read(N);

S := 0;

for C := 1 to N do

S := S + 1 / (Exp(C * Ln(C)));

writeln('S = ',S:5:4);

end.

%N, C: integer;

%S: real;

N = input('Digite um numero: ');

S = 0;

for C = 1 : N

S = S + 1 / C ^ C;

end

fprintf(1,'S = %f\n',S);



50) Faça um algoritmo que:

- leia um número real X do teclado;
- determine e imprima o seguinte somatório:

$$S = X - \frac{X}{1!} + \frac{X}{2!} - \frac{X}{3!} + \frac{X}{4!} + \dots$$

usando os 20 primeiros termos da série.

```
algoritmo L3P050;  
var  
  inteiro: I, J, FAT;  
  real: X, S;  
início  
  S ← 0;  
  leia (X);  
  para I de 1 até 20 faça  
    FAT ← 1;  
    para J de 2 até I-1 faça  
      FAT ← FAT * J;  
    fim-para  
    se ( I mod 2 = 0 ) então  
      S ← S - X / FAT; { termo par }  
    senão  
      S ← S + X / FAT; { termo ímpar }  
    fim-se  
  fim-para  
  imprima("S = ",S);  
fim
```

```
algoritmo L3P050B;  
var  
  inteiro: I, J, FAT;  
  real: X, S;  
início  
  leia (X);  
  S <- 0;  
  FAT <- 1;  
  para I de 1 até 20 faça  
    se ( I mod 2 = 0 ) então  
      S <- S - X / FAT; { termo par }  
    senão  
      S <- S + X / FAT; { termo ímpar }  
    fim-se  
    FAT <- FAT * I;  
  fim-para  
  imprima("S = ",S);  
fim
```



```
program l3p050;  
var  
  I, J: integer;  
  X, FAT, S: real;  
begin  
  write('Digite X: ');  
  readLn(X);  
  S := 0;  
  for I := 1 to 20 do  
    begin  
      FAT := 1;  
      for J := 2 to I-1 do  
        FAT := FAT * J;  
      if ( I mod 2 = 0 ) then  
        S := S - X / FAT { termo par }  
      else  
        S := S + X / FAT; { termo ímpar }  
      end;  
      writeLn('S = ',S:5:4);  
    end.  
end.
```

```
program l3p050b;  
var  
  I: integer;  
  X, FAT, S: real;  
begin  
  write('Digite X: ');  
  readLn(X);  
  S := 0;  
  FAT := 1;  
  for I := 1 to 20 do  
    begin  
      if ( I mod 2 = 0 ) then  
        S := S - X / FAT { termo par }  
      else  
        S := S + X / FAT; { termo ímpar }  
      FAT := FAT * I;  
    end;  
    writeLn('S = ',S:5:4);  
  end.  
end.
```

```
%I, J: integer;  
%X, FAT, S: real;  
X = input('Digite X: ');  
S = 0;  
for I = 1 : 20  
  FAT = 1;  
  for J = 2 : I - 1  
    FAT = FAT * J;  
  end  
  if ( mod(I,2) == 0 )  
    S = S - X / FAT; % termo par  
  else  
    S = S + X / FAT; % termo ímpar  
  end  
end  
fprintf(1,'S = %f\n',S);
```



```
%I: integer;
%X, FAT, S: real;
X = input('Digite X: ');
S = 0;
FAT = 1;
for I = 1 : 20
    if ( mod(I,2) == 0 )
        S = S - X / FAT; % termo par
    else
        S = S + X / FAT; % termo ímpar
    end
    FAT = FAT * I;
end
fprintf(1,'S = %f\n',S);
```

51) O valor aproximado do número π pode ser calculado usando-se a série

$$S = 1 - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \frac{1}{9^3} \dots$$

sendo $\pi = \sqrt[3]{S \times 32}$. Faça um algoritmo em PORTUGOL que calcule e imprima o valor de π usando os 51 primeiros termos da série acima.

```

algoritmo L3P051;
var
  inteiro: I;
  real: S, PI;
início
  S ← 0;
  para I de 1 até 51 faça
    se ( I mod 2 = 0 ) então
      S ← S - 1 / ( ( 2 * I - 1 ) ** 3 ); { termo par }
    senão
      S ← S + 1 / ( ( 2 * I - 1 ) ** 3 ); { termo ímpar }
    fim-se
  fim-para
  PI ← ( S * 32 ) ** ( 1 / 3 ); { raiz cúbica através da potenciação }
  PI ← raiz( S * 32, 3 ); { raiz cúbica através da função raiz }
  imprima( "PI = ", PI );
fim

program l3p051;
var
  I: integer;
  S, PI: real;
begin
  S := 0;
  for I := 1 to 51 do
    if ( I mod 2 = 0 ) then
      S := S - 1 / Exp( 3 * Ln( 2 * I - 1 ) ) { termo par }
    else
      S := S + 1 / Exp( 3 * Ln( 2 * I - 1 ) ); { termo ímpar }
    PI := Exp( 1/3 * Ln( S * 32 ) ); { raiz cúbica através da potenciação }
    writeLn( 'PI = ', PI );
  end.

% I: integer;
% S, PI: real;
S = 0;
for I = 1 : 51
  if ( mod(I,2) == 0 )
    S = S - 1 / ( 2 * I - 1 ) ^ 3; % termo par
  else
    S = S + 1 / ( 2 * I - 1 ) ^ 3; % termo ímpar
  end
end
PI = ( S * 32 ) ^ ( 1 / 3 ); % raiz cúbica através da potenciação
fprintf(1, 'PI = %f\n', PI);
fprintf(1, 'Pi = %f\n', pi);

```



52) Implementar um algoritmo em PORTUGOL para calcular o $\text{sen}(X)$. O valor de X deverá ser digitado em graus. O valor do seno de X será calculado pela soma dos 15 primeiros termos da série a seguir:

$$\text{sen}(X) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$

algoritmo L3P052;

var

inteiro: I, J, FAT;

real: X, SEN;

início

SEN ← 0;

leia(X);

para I **de** 1 **até** 15 **faça**

FAT ← 1;

para J **de** 2 **até** 2*I - 1 **faça**

FAT ← FAT * J;

fim-para

se (I **mod** 2 = 0) **então**

SEN ← SEN - (X ** (2 * I - 1)) / FAT; { termo par }

senão

SEN ← SEN + (X ** (2 * I - 1)) / FAT; { termo ímpar }

fim-se

fim-para

imprima("SEN(",X,") = ",SEN);

fim

algoritmo L3P052B;

var

inteiro: I, FAT;

real: X, SEN;

início

FAT ← 1;

SEN ← 0;

leia(X);

para I **de** 1 **até** 15 **faça**

se (X ≠ 0) **então**

se (I **mod** 2 = 0)

então SEN ← SEN - (X ** (2 * I - 1)) / FAT; { termo par }

senão SEN ← SEN + (X ** (2 * I - 1)) / FAT; { termo ímpar }

fim-se

fim-se

FAT ← FAT * (2 * I) * (2 * I + 1);

fim-para

imprima("SEN(",X,") = ",SEN);

fim



```
program l3p052;
var
  I, J: integer;
  X, SN, FAT, XE: real;
begin
  writeln('Sen(x)');
  write('Digite um valor para X: ');
  read(X);
  X := X * 0.01745329252; {converte de graus para radianos}
  SN := 0;
  XE := X;
  for I := 1 to 15 do
  begin
    FAT := 1;
    for J := 2 to 2 * I - 1 do
      FAT := FAT * J;
    if ( I mod 2 = 0 ) then
      SN := SN - XE / FAT { termo par }
    else
      SN := SN + XE / FAT; { termo impar }
    XE := XE * X * X;
  end;
  writeln('Sen(',X:5:4,') = ',SN:9:8);
  writeln('Sin(',X:5:4,') = ',Sin(X):9:8);
end.

program l3p052b;
var
  I: integer;
  X, SN, FAT: real;
begin
  writeln('Sen(x)');
  write('Digite um valor para X: ');
  read(X);
  X := X * 0.01745329252; {converte de graus para radianos}
  FAT := 1;
  SN := 0;
  for I := 1 to 15 do
  begin
    if ( X <> 0 ) then
      if ( I mod 2 = 0 ) then
        SN := SN - Exp( ( 2 * I - 1 ) * Ln(X) ) / FAT { termo par }
      else
        SN := SN + Exp( ( 2 * I - 1 ) * Ln(X) ) / FAT; { termo ímpar }
      FAT := FAT * ( 2 * I ) * ( 2 * I + 1 );
    end;
    writeln('Sen(',X:5:4,') = ',SN:9:8);
    writeln('Sin(',X:5:4,') = ',Sin(X):9:8);
  end.

%I, J: integer;
%X, SN, FAT: real;
disp('Sen(x)');
X = input('Digite um valor para X: ');
X = X * pi/180;
SN = 0;
for I = 1 : 15
  FAT = 1;
  for J = 2 : 2 * I - 1
    FAT = FAT * J;
  end
  if ( mod(I,2) == 0 )
    SN = SN - X ^ ( 2 * I - 1 ) / FAT; % termo par
  else
    SN = SN + X ^ ( 2 * I - 1 ) / FAT; % termo impar
  end
end
fprintf(1,'Sen(%d) = %f\n',X,SN);
fprintf(1,'Sin(%d) = %f\n',X,sin(X));
```



```
fprintf(1,'Sen(x)');
X = input('Digite um valor para X: ');
X = X * pi/180;
FAT = 1;
SN = 0;
for I = 1 : 15
    if ( X ~= 0 )
        if ( mod(I, 2) == 0 )
            SN = SN - X ^ ( 2 * I - 1 ) / FAT; % termo par
        else
            SN = SN + X ^ ( 2 * I - 1 ) / FAT; % termo ímpar
        end
    end
    FAT = FAT * ( 2 * I ) * ( 2 * I + 1 );
end
fprintf(1,'Sen(%d) = %f\n',X,SN);
fprintf(1,'Sin(%d) = %f\n',X,sin(X));
```

- 53) Implementar um algoritmo em PORTUGOL para calcular o $\cos(X)$. O valor de X deverá ser digitado em graus. O valor do cosseno de X será calculado pela soma dos 15 primeiros termos da série a seguir:

$$\cos(X) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

```
algoritmo L3P053;  
var  
  inteiro: I, J, FAT;  
  real: X, COS;  
início  
  COS ← 0;  
  leia(X);  
  para I de 1 até 15 faça  
    FAT ← 1;  
    para J de 2 até 2*I - 2 faça  
      FAT ← FAT * J;  
    fim-para  
    se ( I mod 2 = 0 ) então  
      COS ← COS - ( X ** ( 2 * I - 2 ) ) / FAT; { termo par }  
    senão  
      COS ← COS + ( X ** ( 2 * I - 2 ) ) / FAT; { termo ímpar }  
    fim-se  
  fim-para  
  imprima("COS(",X,") = ",COS);  
fim  
  
algoritmo L3P053B;  
var  
  inteiro: I, FAT;  
  real: X, COS;  
início  
  FAT ← 1;  
  COS ← 0;  
  leia(X);  
  para I de 1 até 15 faça  
    se (X ≠ 0) então  
      se ( I mod 2 = 0 ) então  
        COS ← COS - ( X ** ( 2 * I - 2 ) ) / FAT; { termo par }  
      senão  
        COS ← COS + ( X ** ( 2 * I - 2 ) ) / FAT; { termo ímpar }  
      fim-se  
    fim-se  
    FAT ← FAT * ( 2 * I - 1 ) * ( 2 * I );  
  fim-para  
  imprima("COS(",X,") = ",COS);  
fim
```



```
program l3p053;
var
  I, J: integer;
  X, CS, FAT, XE: real;
begin
  writeln('Cos(x)');
  write('Digite um valor para X: ');
  read(X);
  X := X * 0.01745329252; {converte de graus para radianos}
  CS := 0;
  XE := 1;
  for I := 1 to 15 do
  begin
    FAT := 1;
    for J := 2 to 2 * I - 2 do
      FAT := FAT * J;
    if ( I mod 2 = 0 ) then
      CS := CS - XE / FAT { termo par }
    else
      CS := CS + XE / FAT; { termo impar }
    XE := XE * X * X;
  end;
  writeln('Cos(',X:5:4,') = ',CS:9:8);
  writeln('Cos(',X:5:4,') = ',Cos(X):9:8);
end.

program l3p053b;
var
  I: integer;
  X, CS, FAT: real;
begin
  writeln('Cos(x)');
  write('Digite um valor para X: ');
  read(X);
  X := X * 0.01745329252; {converte de graus para radianos}
  FAT := 2; {2!}
  CS := 1;
  for I := 2 to 15 do
  begin
    if ( X <> 0 ) then
      if ( I mod 2 = 0 ) then
        CS := CS - Exp( ( 2 * I - 2 ) * Ln(X) ) / FAT { termo par }
      else
        CS := CS + Exp( ( 2 * I - 2 ) * Ln(X) ) / FAT; { termo impar }
      FAT := FAT * ( 2 * I - 1 ) * ( 2 * I );
    end;
    writeln('Cos(',X:5:4,') = ',CS:9:8);
    writeln('Cos(',X:5:4,') = ',Cos(X):9:8);
  end.

% I, J: integer;
% X, CS, FAT: real;
disp('Cos(x)');
X = input('Digite um valor para X: ');
X = X * pi/180;
CS = 0;
for I = 1 : 15
  FAT = 1;
  for J = 2 : 2 * I - 2
    FAT = FAT * J;
  end
  if ( mod(I,2) == 0 )
    CS = CS - X ^ ( 2 * I - 2 ) / FAT; % termo par
  else
    CS = CS + X ^ ( 2 * I - 2 ) / FAT; % termo impar
  end
end
fprintf(1,'Cos(%d) = %f\n',X,CS);
fprintf(1,'Cos(%d) = %f\n',X,cos(X));
```



```
% I: integer;
% X, CS, FAT: real;
fprintf(1,'Cos(x)');
X = input('Digite um valor para X: ');
X = X * pi/180;
FAT = 2; % 2!
CS = 1;
for I = 2 : 15
    if ( X ~= 0 )
        if ( mod(I, 2) == 0 )
            CS = CS - X ^ ( 2 * I - 2 ) / FAT; % termo par
        else
            CS = CS + X ^ ( 2 * I - 2 ) / FAT; % termo ímpar
        end
    end
    FAT = FAT * ( 2 * I - 1 ) * ( 2 * I );
end
fprintf(1,'Cos(%d) = %f\n',X,CS);
fprintf(1,'Cos(%d) = %f\n',X,cos(X));
```

- 54) Implementar um algoritmo para calcular o valor de e^x . O valor de X deverá ser digitado. O valor de e^x será calculado pela soma dos 15 primeiros termos da série a seguir:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

```

algoritmo L3P054;
var
  inteiro: I, J, FAT;
  real: X, E;
início
  E ← 0;
  leia(X);
  para I de 1 até 15 faça
    FAT ← 1;
    para J de 2 até I - 1 faça
      FAT ← FAT * J;
    fim-para
    E ← E + ( X ** ( I - 1 ) ) / FAT;
  fim-para
  imprima("E^",X," = ",E);
fim
  
```

```

algoritmo L3P054B;
var
  inteiro: I, FAT;
  real: X, E;
início
  FAT ← 1;
  COS ← 0;
  leia(X);
  para I de 1 até 15 faça
    E ← E - ( X ** ( I - 1 ) ) / FAT;
    FAT ← FAT * I;
  fim-para
  imprima("E^",X," = ",E);
fim
  
```

```

program l3p054;
var
  I, J: integer;
  X, E, FAT, XE: real;
begin
  writeln('Exp(x)');
  write('Digite um valor para X: ');
  read(X);
  E := 0;
  XE := 1;
  for I := 1 to 15 do
  begin
    FAT := 1;
    for J := 2 to I - 1 do
      FAT := FAT * J;
    E := E + XE / FAT;
    XE := XE * X;
  end;
  writeln('Exp(',X:5:4,') = ',E:9:8);
  writeln('Exp(',X:5:4,') = ',Exp(X):9:8);
end.
  
```



```
program l3p054b;
var
  I: integer;
  X, E, FAT: real;
begin
  writeln('Exp(x)');
  write('Digite um valor para X: ');
  read(X);
  FAT := 1;
  E := 1;
  for I := 2 to 15 do
    begin
      if ( X <> 0 ) then
        E := E + Exp( ( I - 1 ) * Ln(X) ) / FAT;
        FAT := FAT * I;
      end;
      writeln('Exp(',X:5:4,') = ',E:9:8);
      writeln('Exp(',X:5:4,') = ',Exp(X):9:8);
    end.
end.
```

```
% I, J: integer;
% X, E, FAT: real;
disp('Exp(x)');
X = input('Digite um valor para X: ');
E = 0;
for I = 1 : 15
  FAT = 1;
  for J = 2 : I - 1
    FAT = FAT * J;
  end
  E = E + X ^ ( I - 1 ) / FAT;
end
fprintf(1,'Exp(%d) = %f\n',X,E);
fprintf(1,'Exp(%d) = %f\n',X,exp(X));
```

```
% I: integer;
% X, E, FAT: real;
fprintf(1,'Exp(x)');
X = input('Digite um valor para X: ');
FAT = 1;
E = 1;
for I = 2 : 15
  if ( X ~= 0 )
    E = E + X ^ ( I - 1 ) / FAT;
  end
  FAT = FAT * I;
end
fprintf(1,'Exp(%d) = %f\n',X,E);
fprintf(1,'Exp(%d) = %f\n',X,exp(X));
```



55) Criar um algoritmo que leia o valor de N, imprima a sequência a seguir e o resultado.

$$N! / 0! - (N-1)! / 2! + (N-2)! / 4! - (N-3)! / 6! + \dots 0! / (2N)!$$

```
algoritmo L3P055;  
var  
  inteiro: I, J, N, FATN, FATD;  
  real: SOMA, R;  
início  
  leia(N);  
  SOMA ← 0;  
  para I de 0 até N faça  
    FATN ← 1;  
    para J de 2 até N - I faça  
      FATN ← FATN * J;  
    fim-para  
    FATD ← 1;  
    para J ← 2 até 2*I faça  
      FATD ← FATD * J;  
    fim-para  
    R ← FATN / FATD;  
    se (I mod 2 = 1) então  
      imprima (I+1, 'o. Termo = ', R);  
      SOMA ← SOMA + R;  
    senão  
      imprima (I+1, 'o. Termo = -', R);  
      SOMA ← SOMA - R;  
    fim-se  
  fim-para  
  imprima ("Soma = ", SOMA);  
fim  
  
algoritmo L3P055B;  
var  
  inteiro: I, J, FATN, FATD;  
  real: SOMA, R;  
início  
  SOMA ← 0;  
  FATN ← 1;  
  FATD ← 1;  
  leia(N);  
  para J de 2 até N faça  
    FATN ← FATN * J;  
  fim-para  
  para I de 0 até N faça  
    R ← FATN / FATD;  
    se (I mod 2 = 1) então  
      imprima (I+1, 'o. Termo = ', R);  
      SOMA ← SOMA + R;  
    senão  
      imprima (I+1, 'o. Termo = -', R);  
      SOMA ← SOMA - R;  
    fim-se  
    se (I ≠ N)  
      então FATN ← FATN / (N - I);  
    fim-se  
    FATD ← FATD * ( 2*I + 1 ) * ( 2*I + 2 );  
  fim-para  
  imprima ("Soma = ", SOMA);  
fim
```




```
program l3p055;
var
  I, J, N: integer;
  SOMA, R, FATN, FATD: real;
begin
  write('Entre com o valor de N: ');
  read(N);
  SOMA := 0;
  for I := 0 to N do
    begin
      FATN := 1;
      for J := 2 to N - I do
        FATN := FATN * J;
      FATD := 1;
      for J := 2 to (2 * I) do
        FATD := FATD * J;
      R := FATN / FATD ;
      if (I mod 2 = 0) then
        begin
          writeln(I + 1, 'o. Termo = ', R:17:16);
          SOMA := SOMA + R;
        end
      else
        begin
          writeln(I + 1, 'o. Termo = -', R:17:16);
          SOMA := SOMA - R;
        end;
      end;
    end;
  writeln('Soma = ', SOMA:17:16);
end.
```

```
program l3p055b;
var
  I, J, N: integer;
  SOMA, R, FATN, FATD: real;
begin
  write('Entre com o valor de N: ');
  read(N);
  SOMA := 0;
  FATN := 1;
  FATD := 1;
  for J := 2 to N do
    FATN := FATN * J;
  for I := 0 to N do
    begin
      R := FATN / FATD;
      if (I mod 2 = 0) then
        begin
          writeln(I + 1, 'o. Termo = ', R:17:16);
          SOMA := SOMA + R;
        end
      else
        begin
          writeln(I + 1, 'o. Termo = -', R:17:16);
          SOMA := SOMA - R;
        end;
      end;
      if (I <> N) then
        FATN := FATN / (N - I);
        FATD := FATD * (2 * I + 1) * (2 * I + 2);
      end;
    end;
  writeln('Soma = ', SOMA:17:16);
end.
```



```
N = input('Entre com o valor de N: ');
SOMA = 0;
for I = 0 : N
    FATN = 1;
    for J = 2 : N - I
        FATN = FATN * J;
    end
    FATD = 1;
    for J = 2 : (2 * I)
        FATD = FATD * J;
    end
    R = FATN / FATD;
    if (mod(I, 2) == 0)
        fprintf(1, '%do. Termo = %4.10f\n', I+1, R);
        SOMA = SOMA + R;
    else
        fprintf(1, '%do. Termo = - %4.10f\n', I+1, R);
        SOMA = SOMA - R;
    end
end
fprintf(1, 'Soma = %f\n', SOMA);

N = input('Entre com o valor de N: ');
SOMA = 0;
FATN = 1;
FATD = 1;
for J = 2 : N
    FATN = FATN * J;
end
for I = 0 : N
    R = FATN / FATD;
    if (mod(I, 2) == 0)
        fprintf(1, '%do. Termo = %4.10f\n', I+1, R);
        SOMA = SOMA + R;
    else
        fprintf(1, '%do. Termo = - %4.10f\n', I+1, R);
        SOMA = SOMA - R;
    end
    if (I ~= N)
        FATN = FATN / (N - I);
    end
    FATD = FATD * ( 2*I + 1 ) * ( 2*I + 2 );
end
fprintf(1, 'Soma = %f\n', SOMA);
```



56) Escreva um algoritmo em PORTUGOL, que leia um conjunto de 50 fichas, cada uma contendo, a altura e o código do sexo de uma pessoa (código = 1 se for masculino e 2 se for feminino), e calcule e imprima:

- a maior e a menor altura da turma;
- a média de altura das mulheres;
- a média de altura da turma.

```
algoritmo L3P056;  
var  
  inteiro: C, CODSEXO, NMULHER;  
  real: ALTURA, MAIOR, MENOR;  
  real: SOMAMULHER, MEDIAMULHER;  
  real: SOMATURMA, MEDIATURMA;  
início  
  NMULHER ← 0;  
  SOMAMULHER ← 0;  
  SOMATURMA ← 0;  
  para C de 1 até 50 faça  
    leia (ALTURA);  
    leia (CODSEXO);  
    se ( C = 1 ) então  
      MAIOR ← ALTURA; { primeira leitura }  
      MENOR ← ALTURA;  
    senão  
      se ( ALTURA > MAIOR ) { leituras posteriores } então  
        MAIOR ← ALTURA;  
      senão  
        se ( ALTURA < MENOR ) então  
          MENOR ← ALTURA;  
      fim-se  
    fim-se  
  fim-se  
  se CODSEXO = 2  
    então NMULHER ← NMULHER + 1;  
    SOMAMULHER ← SOMAMULHER + ALTURA;  
  fim-se  
  SOMATURMA ← SOMATURMA + ALTURA;  
fim-para  
  MEDIAMULHER ← SOMAMULHER / NMULHER;  
  MEDIATURMA ← SOMATURMA / 50;  
  imprima ("Maior altura da turma: ", MAIOR);  
  imprima ("Menor altura da turma: ", MENOR);  
  imprima ("Media de altura das mulheres: ", MEDIAMULHER);  
  imprima ("Media de altura da turma: ", MEDIATURMA );  
fim
```



```
program l3p056;
var
  C, CODSEXO, NMULHER: integer;
  ALTURA, MAIOR, MENOR: real;
  SOMAMULHER, MEDIAMULHER: real;
  SOMATURMA, MEDIATURMA: real;
begin
  NMULHER := 0;
  SOMAMULHER := 0;
  SOMATURMA := 0;
  for C := 1 to 50 do
  begin
    write('Entre com a altura: ');
    readln(ALTURA);
    write('Entre com o código do sexo (1(M),2(F)): ');
    readln(CODSEXO);
    if ( C = 1) then begin
      MAIOR := ALTURA; { primeira leitura }
      MENOR := ALTURA; end
    else
      if ALTURA > MAIOR then { leituras posteriores }
        MAIOR := ALTURA
      else
        if ALTURA < MENOR then
          MENOR := ALTURA;
    if ( CODSEXO = 2 ) then begin
      NMULHER := NMULHER + 1;
      SOMAMULHER := SOMAMULHER + ALTURA; end;
      SOMATURMA := SOMATURMA + ALTURA;
    end;
    MEDIAMULHER := SOMAMULHER / NMULHER;
    MEDIATURMA := SOMATURMA / 50;
    writeln('Maior altura da turma: ', MAIOR:3:2);
    writeln('Menor altura da turma: ', MENOR:3:2);
    writeln('Media de altura das mulheres: ', MEDIAMULHER:3:2);
    writeln('Media de altura da turma: ', MEDIATURMA :3:2);
  end.
end.
```



```
%C, CODSEXO, NMULHER: integer;
%ALTURA, MAIOR, MENOR: real;
%SOMAMULHER, MEDIAMULHER: real;
%SOMATURMA, MEDIATURMA: real;
NMULHER = 0;
SOMAMULHER = 0;
SOMATURMA = 0;
for C = 1 : 50
    ALTURA = input('Entre com a altura: ');
    CODSEX = input('Entre com o código do sexo (1(M),2(F)): ');
    if ( C == 1 ) % primeira leitura
        MAIOR = ALTURA;
        MENOR = ALTURA;
    else
        if ( ALTURA > MAIOR ) % leituras posteriores
            MAIOR = ALTURA;
        else
            if ( ALTURA < MENOR )
                MENOR = ALTURA;
            end
        end
    end
    if ( CODSEX == 2 )
        NMULHER = NMULHER + 1;
        SOMAMULHER = SOMAMULHER + ALTURA;
    end
    SOMATURMA = SOMATURMA + ALTURA;
end
MEDIAMULHER = SOMAMULHER / NMULHER;
MEDIATURMA = SOMATURMA / 50;
fprintf(1, 'Maior altura da turma: %f\n', MAIOR);
fprintf(1, 'Menor altura da turma: %f\n', MENOR);
fprintf(1, 'Media de altura das mulheres: %f\n', MEDIAMULHER);
fprintf(1, 'Media de altura da turma: %f\n', MEDIATURMA);
```



57) Escreva um algoritmo em PORTUGOL que:

- leia 100 fichas, onde cada ficha contém o número de matrícula e a nota de cada aluno de um determinado curso;
- determine e imprima as duas maiores notas, juntamente com o número de matrícula do aluno que obteve cada uma delas;
- Suponha que não exista dois ou mais alunos com a mesma nota.

```
algoritmo L3P057;  
var  
  inteiro: C, NMAT, MNMAT1, MNMAT2;  
  real:    NOTA, MNOTA1, MNOTA2;  
início  
  para C de 1 até 100 faça  
    leia(NMAT);  
    leia(NOTA);  
    se ( C = 1 ) ou ( NOTA > MNOTA1 ) então  
      MNOTA2 ← MNOTA1;  
      MNMAT2 ← MNMAT1;  
      MNOTA1 ← NOTA;  
      MNMAT1 ← NMAT;  
    senão  
      se ( C = 2 ) ou ( NOTA > MNOTA2 ) então  
        MNOTA2 ← NOTA;  
        MNMAT2 ← NMAT;  
    fim-se  
  fim-se  
  imprima("Número de Matrícula: ", MNMAT1, " tem a maior nota: ", MNOTA1);  
  imprima("Número de Matrícula: ", MNMAT2, " tem a segunda maior nota: ", MNOTA2);  
fim
```



```
algoritmo L3P057B;  
var  
  inteiro: C, NMA1,MNMA1,MNMA2;  
  real:    NTA,MNTA1,MNTA2;  
início  
  leia(NMA1);  
  leia(NTA);  
  MNTA1 ← NTA;  
  MNMA1 ← NMA1;  
  leia(NMA1);  
  leia(NTA);  
  se ( NTA > MNTA1 ) então  
    MNTA2 ← MNTA1;  
    MNMA2 ← MNMA1;  
    MNTA1 ← NTA;  
    MNMA1 ← NMA1;  
  senão  
    MNTA2 ← NTA;  
    MNMA2 ← NMA1;  
  fim-se  
  para C de 3 até 100 faça  
    leia(NMA1);  
    leia(NTA);  
    se ( NTA > MNTA1 ) então  
      MNTA2 ← MNTA1;  
      MNMA2 ← MNMA1;  
      MNTA1 ← NTA;  
      MNMA1 ← NMA1;  
    senão  
      se ( NTA > MNTA2 ) então  
        MNTA2 ← NTA;  
        MNMA2 ← NMA1;  
      fim-se  
    fim-se  
  fim-para  
  imprima("Número de Matrícula: ",MNMA1," tem a maior nota: ",MNTA1);  
  imprima("Número de Matrícula: ",MNMA2," tem a segunda maior nota: ",MNTA2);  
fim
```

```
program l3p057;  
var  
  C, NMA1,MNMA1,MNMA2: integer;  
  NTA,MNTA1,MNTA2: real;  
begin  
  for C := 1 to 100 do  
    begin  
      write('Entre com o numero de matricula: ');  
      read(NMA1);  
      write('Entre com a nota: ');  
      read(NTA);  
      if ( C = 1 ) or ( NTA > MNTA1) then begin  
        MNTA2 := MNTA1;  
        MNMA2 := MNMA1;  
        MNTA1 := NTA;  
        MNMA1 := NMA1; end  
      else  
        if ( C = 2 ) or ( NTA > MNTA2 ) then begin  
          MNTA2 := NTA;  
          MNMA2 := NMA1; end;  
        end;  
      writeLn('Numero de Matricula: ',MNMA1,' tem a maior nota: ',MNTA1:5:1);  
      writeLn('Numero de Matricula: ',MNMA2,' tem a segunda maior nota: ',MNTA2:5:1);  
    end.  
end.
```



```
program l3p057b;
var
  C, NMAT,MNMAT1,MNMAT2: integer;
  NOTA,MNOTA1,MNOTA2: real;
begin
  write('Entre com o numero de matricula: ');
  read(NMAT);
  write('Entre com a nota: ');
  read(NOTA);
  MNOTA1 := NOTA;
  MNMAT1 := NMAT;
  write('Entre com o numero de matricula: ');
  read(NMAT);
  write('Entre com a nota: ');
  read(NOTA);
  if ( NOTA > MNOTA1 ) then begin
    MNOTA2 := MNOTA1;
    MNMAT2 := MNMAT1;
    MNOTA1 := NOTA;
    MNMAT1 := NMAT; end
  else begin
    MNOTA2 := NOTA;
    MNMAT2 := NMAT; end;
  for C := 3 to 100 do
  begin
    write('Entre com o numero de matricula: ');
    read(NMAT);
    write('Entre com a nota: ');
    read(NOTA);
    if ( NOTA > MNOTA1 ) then begin
      MNOTA2 := MNOTA1;
      MNMAT2 := MNMAT1;
      MNOTA1 := NOTA;
      MNMAT1 := NMAT; end
    else if ( NOTA > MNOTA2 ) then begin
      MNOTA2 := NOTA;
      MNMAT2 := NMAT; end;
    end;
    writeln('Numero de Matricula: ',MNMAT1,' tem a maior nota: ',MNOTA1);
    writeln('Numero de Matricula: ',MNMAT2,' tem a segunda maior nota: ',MNOTA2);
  readln(c);
  end.

%C, NMAT,MNMAT1,MNMAT2: integer;
%NOTA,MNOTA1,MNOTA2: real;
MNOTA1 = 0;
MNOTA2 = 0;
MNMAT1 = 0;
MNMAT2 = 0;
for C = 1 : 100
  NMAT = input('Entre com o numero de matricula: ');
  NOTA = input('Entre com a nota: ');
  if ( C == 1 ) | ( NOTA > MNOTA1 )
    MNOTA2 = MNOTA1;
    MNMAT2 = MNMAT1;
    MNOTA1 = NOTA;
    MNMAT1 = NMAT;
  else
    if ( C == 2 ) | ( NOTA > MNOTA2 )
      MNOTA2 = NOTA;
      MNMAT2 = NMAT;
    end
  end
end
end
fprintf(1,'Numero de Matricula: %d tem a maior nota: %f\n',MNMAT1,MNOTA1);
fprintf(1,'Numero de Matricula: %d tem a segunda maior nota: %f\n',MNMAT2,MNOTA2);
```




```
%C, NMAT,MNMAT1,MNMAT2: integer;
% NOTA,MNOTA1,MNOTA2: real;
NMAT = input('Entre com o numero de matricula: ');
NOTA = input('Entre com a nota: ');
MNOTA1 = NOTA;
MNMAT1 = NMAT;
NMAT = input('Entre com o numero de matricula: ');
NOTA = input('Entre com a nota: ');
if ( NOTA > MNOTA1 )
    MNOTA2 = MNOTA1;
    MNMAT2 = MNMAT1;
    MNOTA1 = NOTA;
    MNMAT1 = NMAT;
else
    MNOTA2 = NOTA;
    MNMAT2 = NMAT;
end
for C = 3 : 100
    NMAT = input('Entre com o numero de matricula: ');
    NOTA = input('Entre com a nota: ');
    if ( NOTA > MNOTA1 )
        MNOTA2 = MNOTA1;
        MNMAT2 = MNMAT1;
        MNOTA1 = NOTA;
        MNMAT1 = NMAT;
    elseif ( NOTA > MNOTA2 )
        MNOTA2 = NOTA;
        MNMAT2 = NMAT;
    end
end
fprintf(1,'Numero de Matricula: %d tem a maior nota: %f\n',MNMAT1,MNOTA1);
fprintf(1,'Numero de Matricula: %d tem a segunda maior nota: %f\n',MNMAT2,MNOTA2);
```



58) Um cinema possui capacidade de 100 lugares e está sempre com ocupação total. Certo dia, cada espectador respondeu a um questionário, no qual constava:

- sua idade;
- sua opinião em relação ao filme, segundo as seguintes notas:

Nota	Significado
A	Ótimo
B	Bom
C	Regular
D	Ruim
E	Péssimo

Elabore um algoritmo que, lendo estes dados, calcule e imprima:

- a quantidade de respostas ótimo;
- a diferença percentual entre respostas bom e regular;
- a média de idade das pessoas que responderam ruim;
- a percentagem de respostas péssimo e a maior idade que utilizou esta opção;
- a diferença de idade entre a maior idade que respondeu ótimo e a maior idade que respondeu ruim.



```
algoritmo L3P058;
var
  inteiro: C, IDADE, NOTIMO, NRUIM, NPESSIMO, NBOM, NREGULAR;
  inteiro: SOMAIDADERUIM, MAIORIDADEPESSIMO, MAIORIDADEOTIMO, MAIORIDADERUIM;
  real: MEDIAIDADERUIM;
  caractere: OPINIAO;
início
  NOTIMO ← 0;
  NBOM ← 0;
  NREGULAR ← 0;
  NRUIM ← 0;
  NPESSIMO ← 0;
  SOMAIDADERUIM ← 0;
  para C de 1 até 100 faça
    leia(OPINIAO);
    leia(IDADE);
    escolha(OPINIAO)
    se ( OPINIAO = "A" ) então
      NOTIMO ← NOTIMO + 1;
      se ( NOTIMO = 1 ) ou ( IDADE > MAIORIDADEOTIMO ) então
        MAIORIDADEOTIMO ← IDADE;
      fim-se
    senão-se ( OPINIAO = "B" ) então
      NBOM ← NBOM + 1;
    senão-se ( OPINIAO = "C" ) então
      NREGULAR ← NREGULAR + 1;
    senão-se ( OPINIAO = "D" ) então
      NRUIM ← NRUIM + 1;
      SOMAIDADERUIM ← SOMAIDADERUIM + IDADE;
      se ( NRUIM = 1 ) ou ( IDADE > MAIORIDADERUIM ) então
        MAIORIDADERUIM ← IDADE;
      fim-se
    senão-se ( OPINIAO = "E" ) então
      NPESSIMO ← NPESSIMO + 1;
      se ( NPESSIMO = 1 ) ou ( IDADE > MAIORIDADEPESSIMO ) então
        MAIORIDADEPESSIMO ← IDADE;
      fim-se
    senão
      imprima("Opção de opinião inválida!");
    fim-se
  fim-para
  imprima("A quantidade de respostas ótimo é: ", NOTIMO);
  imprima("A diferença percentual entre repostas bom e regular: ", 100*(NBOM-NREGULAR)/100 );
  se ( NRUIM ≠ 0 ) { evitando divisão por zero } então
    MEDIAIDADERUIM ← SOMAIDADERUIM / NRUIM;
    imprima("A média de idade de pessoas que responderam ruim é: ", MEDIAIDADERUIM);
  senão
    imprima("Nenhuma pessoa respondeu ruim");
  fim-se
  imprima("A percentagem de pessoas que responderam péssimo: ", NPESSIMO);
  imprima("A maior idade que respondeu péssimo é: ", MAIORIDADEPESSIMO );
  DIF ← MAIORIDADEOTIMO - MAIORIDADERUIM;
  imprima("A diferença entre a maior idade de resposta ótimo e ruim é: ", DIF);
fim
```



```
program l3p058;
var
  C, IDADE, NOTIMO, NRUIM, NPRESSIMO, NBOM, NREGULAR: integer;
  SOMAIDADERUIM, MAIORIDADEPESSIMO, MAIORIDADEOTIMO, MAIORIDADERUIM, DIF: integer;
  MEDIAIDADERUIM: real;
  OPINIAO: string;
begin
  NOTIMO := 0;
  NBOM := 0;
  NREGULAR := 0;
  NRUIM := 0;
  NPRESSIMO := 0;
  SOMAIDADERUIM := 0;
  MAIORIDADEOTIMO := 0;
  MAIORIDADERUIM := 0;
  MAIORIDADEPESSIMO := 0;
  for C := 1 to 100 do
  begin
    write('Digite a idade: ');
    readLn(IDADE);
    write('Digite a Opinioao (A/B/C/D/E): ');
    readLn(OPINIAO);
    if (OPINIAO = 'A' ) then begin
      NOTIMO := NOTIMO + 1;
      if ( NOTIMO = 1 ) or ( IDADE > MAIORIDADEOTIMO ) then
        MAIORIDADEOTIMO := IDADE; end
    else if ( OPINIAO = 'B' ) then
      NBOM := NBOM + 1
    else if ( OPINIAO = 'C' ) then
      NREGULAR := NREGULAR + 1
    else if ( OPINIAO = 'D' ) then begin
      NRUIM := NRUIM + 1;
      SOMAIDADERUIM := SOMAIDADERUIM + IDADE;
      if ( NRUIM = 1 ) or ( IDADE > MAIORIDADERUIM ) then
        MAIORIDADERUIM := IDADE; end
    else if ( OPINIAO = 'E' ) then begin
      NPRESSIMO := NPRESSIMO + 1;
      if ( NPRESSIMO = 1 ) or ( IDADE > MAIORIDADEPESSIMO ) then
        MAIORIDADEPESSIMO := IDADE; end
    else
      writeLn('Opcao de opiniao invalida!');
    end;
    writeLn('A quantidade de respostas otimo eh: ', NOTIMO);
    writeLn('A diferenca percentual entre repostas bom e regular: ', (NBOM-NREGULAR), '% ');
    if ( NRUIM <> 0 ) then { evitando divisao por zero } begin
      MEDIAIDADERUIM := SOMAIDADERUIM / NRUIM;
      writeLn('A media de idade de pessoas que responderam ruim eh: ', MEDIAIDADERUIM:4:1); end
    else
      writeLn('Nenhuma pessoa respondeu ruim');
    writeLn('A percentagem de pessoas que responderam pessimo: ', NPRESSIMO, '% ');
    writeLn('A maior idade que respondeu pessimo eh: ', MAIORIDADEPESSIMO );
    DIF := MAIORIDADEOTIMO - MAIORIDADERUIM;
    writeLn('A diferenca entre a maior idade de resposta otimo e ruim eh: ', DIF);
  end.
end.
```



```
% C, IDADE,NOTIMO,NRUIM,NPESSIMO,NBOM,NREGULAR: integer;
% SOMAIDADERUIM,MAIORIDADEPESSIMO,MAIORIDADEOTIMO,MAIORIDADERUIM,DIF: integer;
% MEDIAIDADERUIM: real;
% OPINIAO: string;
NOTIMO = 0;
NBOM = 0;
NREGULAR = 0;
NRUIM = 0;
NPESSIMO = 0;
SOMAIDADERUIM = 0;
MAIORIDADEOTIMO = 0;
MAIORIDADERUIM = 0;
MAIORIDADEPESSIMO = 0;
for C = 1 : 100
    IDADE = input('Digite a idade: ');
    OPINIAO = input('Digite a Opinioao (A/B/C/D/E): ');
    if (OPINIAO == 'A' )
        NOTIMO = NOTIMO + 1;
        if ( NOTIMO == 1 ) | ( IDADE > MAIORIDADEOTIMO )
            MAIORIDADEOTIMO = IDADE;
        end
    elseif ( OPINIAO == 'B' )
        NBOM = NBOM + 1;
    elseif ( OPINIAO == 'C' )
        NREGULAR = NREGULAR + 1;
    elseif ( OPINIAO == 'D' )
        NRUIM = NRUIM + 1;
        SOMAIDADERUIM = SOMAIDADERUIM + IDADE;
        if ( NRUIM == 1 ) | ( IDADE > MAIORIDADERUIM )
            MAIORIDADERUIM = IDADE;
        end
    elseif ( OPINIAO == 'E' )
        NPESSIMO = NPESSIMO + 1;
        if ( NPESSIMO == 1 ) | ( IDADE > MAIORIDADEPESSIMO )
            MAIORIDADEPESSIMO = IDADE;
        end
    else
        fprintf(1,'Opcao de opiniao invalida!');
    end
end
fprintf(1,'A quantidade de respostas otimo eh: %d\n',NOTIMO);
fprintf(1,'A diferenca percentual entre repostas bom e regular: %.2f\n',(NBOM-NREGULAR));
if ( NRUIM ~= 0 ) % evitando divisao por zero
    MEDIAIDADERUIM = SOMAIDADERUIM / NRUIM;
    fprintf(1,'A media de idade de pessoas que responderam ruim eh: %f\n',MEDIAIDADERUIM);
else
    fprintf(1,'Nenhuma pessoa respondeu ruim \n');
end
fprintf(1,'A percentagem de pessoas que responderam pessimo: %f\n',NPESSIMO );
fprintf(1,'A maior idade que respondeu pessimo eh: %d\n',MAIORIDADEPESSIMO );
DIF = MAIORIDADEOTIMO - MAIORIDADERUIM;
fprintf(1,'A diferenca entre a maior idade de resposta otimo e ruim eh: %f\n',DIF);
```



59) Em um prédio, com 50 moradores, há três elevadores denominados A, B e C. Para otimizar o sistema de controle dos elevadores, foi realizado um levantamento no qual cada usuário respondia:

- o elevador que utilizava com mais frequência;
- o período que utilizava o elevador, entre:
 - “M” = matutino;
 - “V” = vespertino;
 - “N” = noturno.

Construa um algoritmo que calcule e imprima:

- qual é o elevador mais freqüentado e em que período se concentra o maior fluxo;
- qual o período mais usado de todos e a que elevador pertence;
- qual a diferença percentual entre o mais usado dos horários e o menos usado;
- qual a porcentagem sobre o total de serviços prestados do elevador de média utilização.



```
algoritmo L3P059;
var
  inteiro: C, NAM, NAV, NAN, NBM, NBV, NBN, NCM, NCV, NCN;
  caractere: ELEVADOR, PERIODO;
início
  NAM ← 0; NAV ← 0; NAN ← 0;
  NBM ← 0; NBV ← 0; NBN ← 0;
  NCM ← 0; NCV ← 0; NCN ← 0;
  para C de 1 até 50 faça
    leia(ELEVADOR);
    leia(PERIODO);
    se ( ELEVADOR = "A" ) então
      se ( PERIODO = "M" ) então
        NAM ← NAM + 1;
      senão-se ( PERIODO = "V" ) então
        NAV ← NAV + 1;
      senão
        NAN ← NAN + 1;
    fim-se
    senão-se ( ELEVADOR = "B" ) então
      se ( PERIODO = "M" ) então
        NBM ← NBM + 1;
      senão-se ( PERIODO = "V" ) então
        NBV ← NBV + 1;
      senão
        NBN ← NBN + 1;
    fim-se
    senão-se ( ELEVADOR = "C" ) então
      se ( PERIODO = "M" ) então
        NCM ← NCM + 1;
      senão-se ( PERIODO = "V" ) então
        NCV ← NCV + 1;
      senão
        NCN ← NCN + 1;
    fim-se
  fim-se
  fim-para
  se ( (NAM+NAV+NAN) > (NBM+NBV+NBN) ) e ( (NAM+NAV+NAN) > (NCM+NCV+NCN) ) então
    imprima("A é o elevador mais freqüentado, no período ");
    se ( (NAM > NAV) e (NAM > NAN) ) então
      imprima("Matutino!");
    senão-se ( (NAV > NAM) e (NAV > NAN) ) então
      imprima("Vespertino!");
    senão
      imprima("Noturno!");
    fim-se
  senão-se ( (NBM+NBV+NBN) > (NAM+NAV+NAN) ) e ( (NBM+NBV+NBN) > (NCM+NCV+NCN) ) então
    imprima("B é o elevador mais freqüentado, no período ");
    se ( (NBM > NBV) e (NBM > NBN) ) então
      imprima("Matutino!");
    senão-se ( (NBV > NBM) e (NBV > NBN) ) então
      imprima("Vespertino!");
    senão
      imprima("Noturno!");
    fim-se
  senão-se ( (NCM+NCV+NCN) > (NAM+NAV+NAN) ) e ( (NCM+NCV+NCN) > (NBM+NBV+NBN) ) então
    imprima("C é o elevador mais freqüentado, no período ");
    se ( (NCM > NCV) e (NCM > NCN) ) então
      imprima("Matutino!");
    senão-se ( (NCV > NCM) e (NCV > NCN) ) então
      imprima("Vespertino!");
    senão
      imprima("Noturno!");
    fim-se
  fim-se
  { continua }
fim
```



```
program l3p059;
var
  C, NAM, NAV, NAN, NBM, NBV, NBN, NCM, NCV, NCN: integer;
  ELEVADOR, PERIODO: string;
begin
  NAM := 0; NAV := 0; NAN := 0;
  NBM := 0; NBV := 0; NBN := 0;
  NCM := 0; NCV := 0; NCN := 0;
  for C := 1 to 50 do
    begin
      write('Elevador: ');
      readLn(ELEVADOR);
      write('Período: ');
      readLn(PERIODO);
      if ( ELEVADOR = 'A' ) then
        if ( PERIODO = 'M' ) then
          NAM := NAM + 1
        else if ( PERIODO = 'V' ) then
          NAV := NAV + 1
        else
          NAN := NAN + 1
      else if ( ELEVADOR = 'B' ) then
        if ( PERIODO = 'M' ) then
          NBM := NBM + 1
        else if ( PERIODO = 'V' ) then
          NBV := NBV + 1
        else
          NBN := NBN + 1
      else if ( ELEVADOR = 'C' ) then
        if ( PERIODO = 'M' ) then
          NCM := NCM + 1
        else if ( PERIODO = 'V' ) then
          NCV := NCV + 1
        else
          NCN := NCN + 1;
    end;
    if ( (NAM+NAV+NAN) > (NBM+NBV+NBN) ) and ( (NAM+NAV+NAN) > (NCM+NCV+NCN) ) then
      begin
        write('A eh o elevador mais frequentado, no período ');
        if ( (NAM > NAV) and (NAM > NAN) ) then
          writeLn('Matutino!')
        else if ( (NAV > NAM) and (NAV > NAN) ) then
          writeLn('Vespertino!')
        else
          writeLn('Noturno!');
      end
    else if ( (NBM+NBV+NBN) > (NAM+NAV+NAN) ) and ( (NBM+NBV+NBN) > (NCM+NCV+NCN) ) then
      begin
        write('B eh o elevador mais frequentado, no período ');
        if ( (NBM > NBV) and (NBM > NBN) ) then
          writeLn('Matutino!')
        else if ( (NBV > NBM) and (NBV > NBN) ) then
          writeLn('Vespertino!')
        else
          writeLn('Noturno!')
      end
    else if ( (NCM+NCV+NCN) > (NAM+NAV+NAN) ) and ( (NCM+NCV+NCN) > (NBM+NBV+NBN) ) then
      begin
        write('C eh o elevador mais frequentado, no período ');
        if ( (NCM > NCV) and (NCM > NCN) ) then
          writeLn('Matutino!')
        else if ( (NCV > NCM) and (NCV > NCN) ) then
          writeLn('Vespertino!')
        else
          writeLn('Noturno!');
      end;
    end;
  { continua }
end.
```




```
% C, NAM,NAV,NAN,NBM,NBV,NBN,NCM,NCV,NCN: integer;
% ELEVADOR,PERIODO: string;
NAM = 0;NAV = 0;NAN = 0;
NBM = 0;NBV = 0;NBN = 0;
NCM = 0;NCV = 0;NCN = 0;
for C = 1 : 50
    ELEVADOR = input('Elevador: ');
    PERIODO = input('Período: ');
    if ( ELEVADOR == 'A' )
        if ( PERIODO == 'M' )
            NAM = NAM + 1;
        elseif ( PERIODO == 'V' )
            NAV = NAV + 1;
        else
            NAN = NAN + 1;
        end
    elseif ( ELEVADOR == 'B' )
        if ( PERIODO == 'M' )
            NBM = NBM + 1;
        elseif ( PERIODO == 'V' )
            NBV = NBV + 1;
        else
            NBN = NBN + 1;
        end
    elseif ( ELEVADOR == 'C' )
        if ( PERIODO == 'M' )
            NCM = NCM + 1;
        elseif ( PERIODO == 'V' )
            NCV = NCV + 1;
        else
            NCN = NCN + 1;
        end
    end
end
if ( NAM+NAV+NAN > NBM+NBV+NBN ) & ( NAM+NAV+NAN > NCM+NCV+NCN )
    disp('A eh o elevador mais frequentado, no período ');
    if ( NAM > NAV ) & ( NAM > NAN )
        disp('Matutino!');
    elseif ( NAV > NAM ) & ( NAV > NAN )
        disp('Vespertino!');
    else
        disp('Noturno!');
    end
elseif ( NBM+NBV+NBN > (NAM+NAV+NAN) & (NBM+NBV+NBN) > (NCM+NCV+NCN)
    disp('B eh o elevador mais frequentado, no período ');
    if ( NBM > NBV ) & ( NBM > NBN )
        disp('Matutino!');
    elseif ( NBV > NBM ) & ( NBV > NBN )
        disp('Vespertino!');
    else
        disp('Noturno!');
    end
elseif ( NCM+NCV+NCN > NAM+NAV+NAN ) & ( NCM+NCV+NCN > NBM+NBV+NBN )
    disp('C eh o elevador mais frequentado, no período ');
    if ( NCM > NCV ) & ( NCM > NCN )
        disp('Matutino!');
    elseif ( NCV > NCM ) & ( NCV > NCN )
        disp('Vespertino!');
    else
        disp('Noturno!');
    end
end
end
% continua
```

- 60) Calcule o imposto de renda de um grupo de 10 contribuintes, considerando que os dados de cada contribuinte, número do CPF, número de dependentes e renda mensal são valores fornecidos pelo usuário. Para cada contribuinte será feito um desconto de 5% do salário mínimo por dependente.

Os valores da alíquota para cálculo do imposto são:

Renda Líquida	Alíquota
Até 2 salários mínimos	Isento
2 a 3 salários mínimos	5%
3 a 5 salários mínimos	10%
5 a 7 salários mínimos	15%
Acima de 7 salários mínimos	20%

Observe que deve ser fornecido o valor atual do salário mínimo para que o algoritmo calcule os valores corretamente.

```

algoritmo L3P060;
var
  inteiro:    C, CPF, NDEP;
  real:      RENDA, SALMIN, IMP, NSALMIN;
início
  leia (SALMIN);
  para C de 1 até 10 faça
    leia (CPF);
    leia (NDEP);
    leia (RENDA);
    NSALMIN ← RENDA / SALMIN;
    se ( NSALMIN < 2 ) então
      IMP ← 0;
    senão-se ( NSALMIN < 3 ) então
      IMP ← RENDA*5/100;
    senão-se ( NSALMIN < 5 ) então
      IMP ← RENDA*10/100;
    senão-se ( NSALMIN < 7 ) então
      IMP ← RENDA*15/100;
    senão
      IMP ← RENDA*20/100;
  fim-se
  IMP ← IMP + NDEP * (SALMIN*5/100);
  se ( IMP > 0 ) então
    imprima ("Imposto a ser pago: ", IMP);
  senão
    imprima ("Imposto a ser restituído: ", -IMP);
  fim-se
fim-para
fim
  
```



```
program l3p060;
var
  C, CPF, NDEP: integer;
  RENDA, SALMIN, IMP, NSALMIN: real;
begin
  write('Informe o salario minimo: ');
  readLn(SALMIN);
  for C := 1 to 10 do
    begin
      write('CPF: ');
      readLn(CPF);
      write('N Dep: ');
      readLn(NDEP);
      write('Renda: ');
      readLn(RENDA);
      NSALMIN := RENDA / SALMIN;
      if ( NSALMIN < 2 ) then
        IMP := 0
      else if ( NSALMIN < 3 ) then
        IMP := RENDA*5/100
      else if ( NSALMIN < 5 ) then
        IMP := RENDA*10/100
      else if ( NSALMIN < 7 ) then
        IMP := RENDA*15/100
      else
        IMP := RENDA*20/100;
      IMP := IMP - NDEP * (SALMIN*5/100);
      if ( IMP > 0 ) then
        writeLn('Imposto a ser pago: ',IMP:3:2)
      else
        writeLn('Imposto a ser restituído: ',-IMP:3:2);
    end;
  end.

SALMIN = input('Informe o salario minimo: ');
for C = 1 : 10
  CPF = input('CPF: ');
  NDEP = input('N Dep: ');
  RENDA = input('Renda: ');
  NSALMIN = RENDA / SALMIN;
  if ( NSALMIN < 2 )
    IMP = 0;
  elseif ( NSALMIN < 3 )
    IMP = RENDA*5/100;
  elseif ( NSALMIN < 5 )
    IMP = RENDA*10/100;
  elseif ( NSALMIN < 7 )
    IMP = RENDA*15/100;
  else
    IMP = RENDA*20/100;
  end
  IMP = IMP - NDEP * (SALMIN*5/100);
  if ( IMP > 0 )
    fprintf(1,'Imposto a ser pago: %.2f\n',IMP);
  else
    fprintf(1,'Imposto a ser restituído: %.2f\n',-IMP);
  end
end
```