

# Técnicas de solução de problemas

## Objetivos:

- Apresentar alguns problemas e discutir soluções algorítmicas para eles
- Implementar códigos em Pascal para estes algoritmos

Quantos números temos aqui?

8 4 3 7 2

E aqui?

18 65 17 34 22 27 42 48 50

E agora???

12 89 66 28 24 82 13 64 52  
37 57 78 70 38 30 7 53 70 26  
77 13 25 20 11 48 51 18 23  
61 62 52 37 21 32 54 72 30  
29 78 74 81 17 16 77 67 45  
29 48 58 40 81 40 54 58 42  
83 46 57 78 26 35 50 67 19

Agora vamos ver vários números, até encontrarmos a palavra “FIM”.

A pergunta é a mesma: **quantos números temos?**

18

65

17



34

22

27

42

48

50

FIM

Próximo problema



Qual é a soma destes números?

8 4 3 7 2

E a soma destes outros?

18 65 17 34 22 27 42 48 50

E agora???

12 89 66 28 24 82 13 64 52  
37 57 78 70 38 30 7 53 70 26  
77 13 25 20 11 48 51 18 23  
61 62 52 37 21 32 54 72 30  
29 78 74 81 17 16 77 67 45  
29 48 58 40 81 40 54 58 42  
83 46 57 78 26 35 50 67 19

Agora vamos ver vários números, até encontrarmos a palavra “FIM”.

A pergunta é a mesma: qual é a soma?

18

65

17

34



22

27

42

48

50

FIM

# Algoritmo para problema de contar

Begin

Cont := 0;

Read (n);

While n <> 0 do

Begin

Cont := Cont + 1;

Read (n);

End;

Write (Cont);

End.

# Algoritmo para problema de contar

Begin

Cont := 0;

Read (n);

While n <> 0 do

Begin

Cont := Cont + 1;

Read (n);

End;

Write (Cont);

End.



# Algoritmo base

Begin

    Read (n);

    While n <> 0 do

        Begin

            Read (n);

        End;

End.

# O que faz o algoritmo base?

Begin

Read (n);

While n <> 0 do

Begin

Read (n);

End;

End.

# Modificando o algoritmo base

Begin

Soma := 0;

Read (n);

While n <> 0 do

Begin

Soma := Soma + n;

Read (n);

End;

Write (Soma);

End.

# O que este algoritmo faz?

Begin

    Soma := 0;

    Read (n);

    While n <> 0 do

        Begin

            Soma := Soma + n;

            Read (n);

        End;

    Write (Soma);

End.

# Quais são as diferenças?

Begin

Cont := 0;

Read (n);

While n <> 0 do

Begin

Cont := Cont + 1;

Read (n);

End;

Write (Cont);

End.

Begin

Soma := 0;

Read (n);

While n <> 0 do

Begin

Soma := Soma + n;

Read (n);

End;

Write (Soma);

End.

# A verdadeira diferença! E que diferença!!!

```
Begin
    Cont := 0;
    Read (n);
    While n <> 0 do
        Begin
            Cont := Cont + 1;
            Read (n);
        End;
    Write (Cont);
End.
```

```
Begin
    Soma := 0;
    Read (n);
    While n <> 0 do
        Begin
            Soma := Soma + n;
            Read (n);
        End;
    Write (Soma);
End.
```

# Este conta números

```
Begin
  Cont := 0;
  Read (n);
  While n <> 0 do
    Begin
      Cont := Cont + 1;
      Read (n);
    End;
  Write (Cont);
End.
```

```
Begin
  Soma := 0;
  Read (n);
  While n <> 0 do
    Begin
      Soma := Soma + n;
      Read (n);
    End;
  Write (Soma);
End.
```

# Este soma números

```
Begin
    Cont := 0;
    Read (n);
    While n <> 0 do
        Begin
            Cont := Cont + 1;
            Read (n);
        End;
    Write (Cont);
End.
```

```
Begin
    Soma := 0;
    Read (n);
    While n <> 0 do
        Begin
            Soma := Soma + n;
            Read (n);
        End;
    Write (Soma);
End.
```



# Propriedades do algoritmo base

```
Begin
```

```
    Read (n);
```

```
    While  $n \neq 0$  do
```

```
        Begin
```

```
            Read (n);
```

```
        End;
```

```
End.
```

- a variável  $n$  controla o laço.

# Propriedades do algoritmo base

```
Begin
```

```
    Read (n);
```

```
    While n <> 0 do
```

```
        Begin
```

```
            Read (n);
```

```
        End;
```

```
End.
```

- a variável n controla o laço.
- portanto, deve ser modificada dentro do laço!

# Propriedades do algoritmo base

```
Begin
```

```
    Read (n);
```

```
    While  $n \neq 0$  do
```

```
        Begin
```

```
            Read (n);
```

```
        End;
```

```
End.
```

- a variável  $n$  controla o laço.
- portanto, deve ser modificada dentro do laço!
- senão, os comandos do laço serão executados infinitas vezes.

# Propriedades do algoritmo base

```
Begin
```

```
    Read (n);
```

```
    While  $n \neq 0$  do
```

```
        Begin
```

```
            Read (n);
```

```
        End;
```

```
End.
```

- a variável  $n$  controla o laço.
- portanto, deve ser modificada dentro do laço!
- senão, os comandos do laço serão executados infinitas vezes.

# Algoritmo base

Begin

(\* inicialização de variáveis \*)

Read (n);

While n <> 0 do

Begin

(\* alguma operação \*)

Read (n);

End;

End.

# Algoritmo base

Begin

(\* inicialização de variáveis \*)

Read (n);

While n <> 0 do

Begin

(\* alguma operação \*)

Read (n);

End;

End.

- A inicialização das variáveis depende da operação a ser feita.

# Algoritmo base

Begin

(\* inicialização de variáveis \*)

Read (n);

While n <> 0 do

Begin

(\* alguma operação \*)

Read (n);

End;

End.

- A inicialização das variáveis depende da operação a ser feita.
- A operação pode ser tão complexa quanto se queira (ou quanto se precise!).

# Algoritmo base

Begin

(\* inicialização de variáveis \*)

Read (n);

While n <> 0 do

Begin

(\* alguma operação \*)

Read (n);

End;

End.



# O que afinal importa?

Begin

(\* inicialização \*)

Read (n);

While n <> 0 do

Begin

(\* operações em n \*)

Read (n);

End;

End.

Qualquer algoritmo que tenha como entrada uma sequência de números que deve ser processada de alguma maneira pode usar este código base.