

Introdução à Programação de Computadores para Biologia

Estruturas de controle repetitiva (loops)

Aula 07

<https://ttdorres.github.io/introprog2024/>

ESTRUTURAS DE CONTROLE

O que é verdadeiro/falso em Perl?

Não verdadeiro:

- zero
- string vazia
- lista vazia
- undefined (undef)

ESTRUTURAS DE CONTROLE

Condicionais

1. *if*

```
if (expressao) {  
    bloco de comandos  
}
```

2. *else e elsif*

```
if (expressao) {  
    bloco de comandos  
} else {  
    bloco de comandos  
}
```

3. *boolean*

```
(expressao) && (expressao)  
(expressao) || (expressao)
```

ESTRUTURAS DE CONTROLE

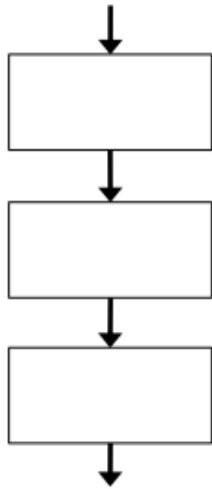
Repetição

Problema:

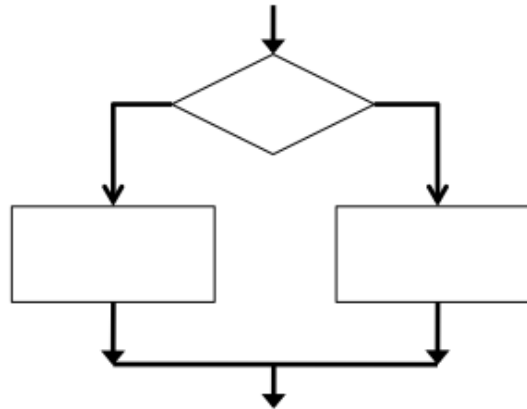
Calcular o tamanho médio de sequências armazenadas em um array.

Imprimir todos os elementos de um hash em ordem alfabética das chaves com seus valores correspondentes.

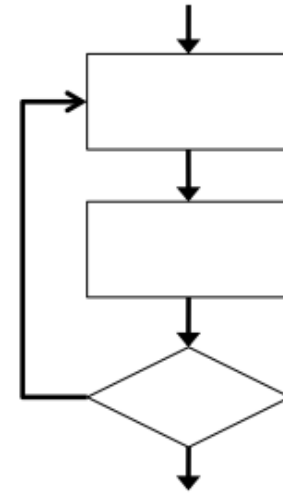
ESTRUTURAS DE CONTROLE



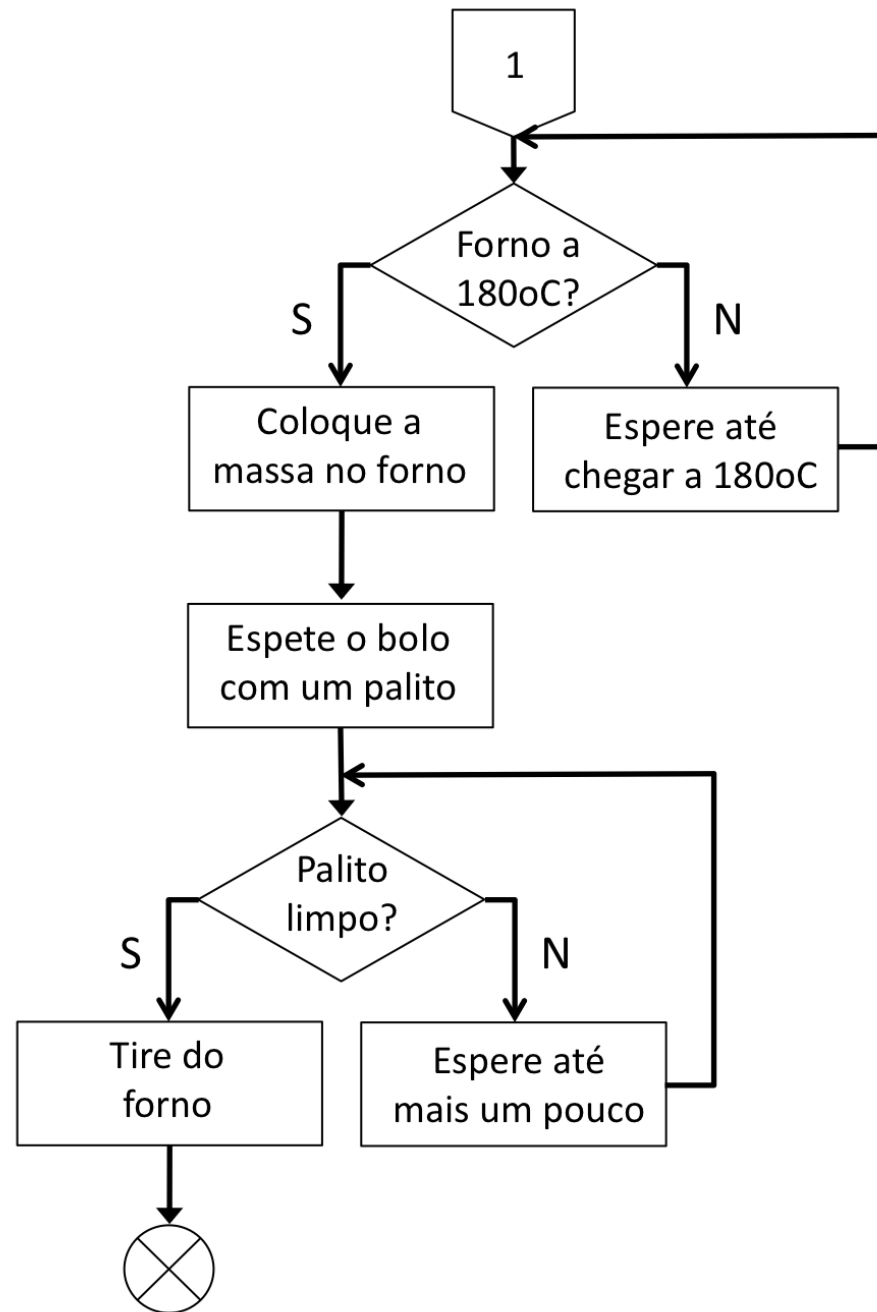
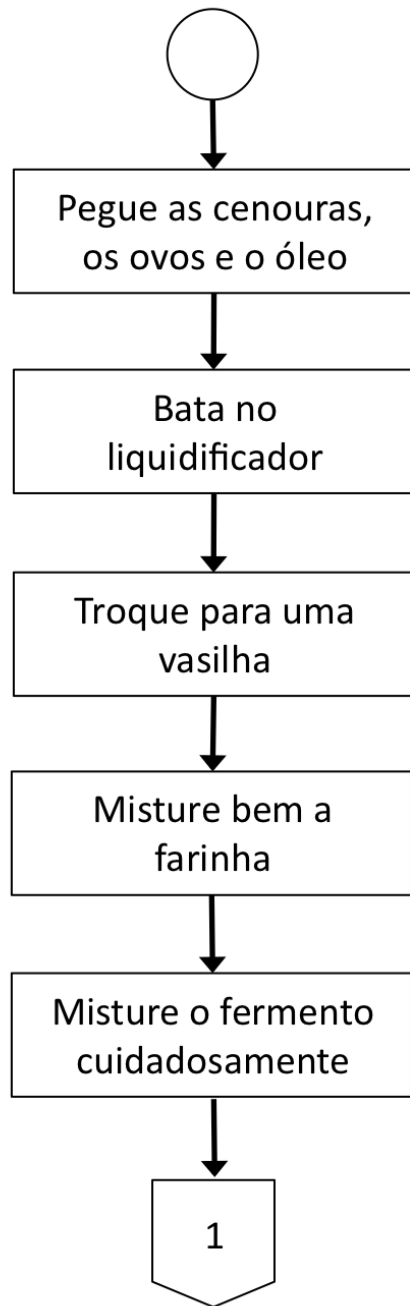
Estrutura de controle
sequencial



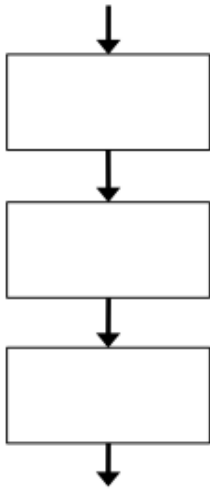
Estrutura de controle
condicional



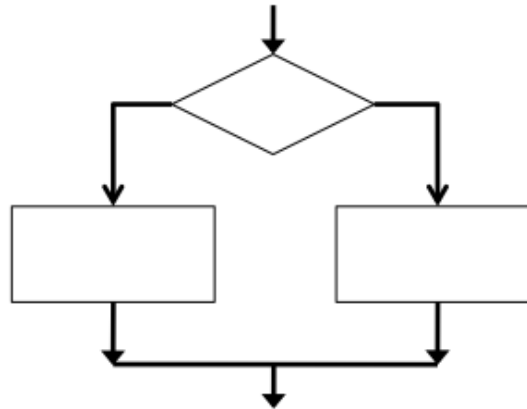
Estrutura de controle
repetitiva



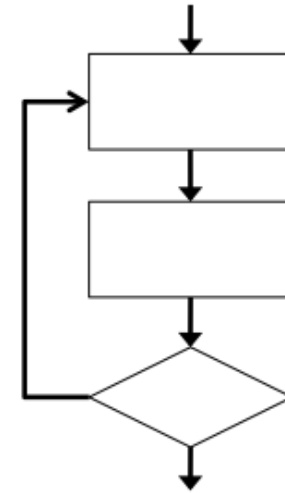
ESTRUTURAS DE CONTROLE



Estrutura de controle
sequencial



Estrutura de controle
condicional



Estrutura de controle
repetitiva

ESTRUTURAS DE CONTROLE

Repetição

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [loops.pl](#)
4. Copiar **exemplo01** da página da disciplina.
5. Continuar o script, calculando o tamanho médio das sequências.
Usar o comando *length()*.

ESTRUTURAS DE CONTROLE

Repetição

Script: [loops.pl](#)

Calcular média, exemplo01:

```
# calcular a media de tamanho das sequencias em um array

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

# Passo 1: acessar elementos do array, calcular e armazenar tamanhos

$ tamanho = # completar com os comandos;

# Passo 2: calcular a media

$media = # completar com os comandos;

# Passo 3: Imprimir resultado

print "\nExemplo 01:\n";
print "Media de tamanho = $media\n";

exit;
```

ESTRUTURAS DE CONTROLE

Repetição

Script: [loops.pl](#)

Calcular média, exemplo01:

```
# calcular a media de tamanho das sequencias em um array

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

# Passo 1: acessar elementos do array, calcular e armazenar comprimento

$tamanho = length($seqs[0]) + length($seqs[1]) + length($seqs[2]);

# Passo 2: calcular a media

$seqs = @seqs; #numero de elementos do array
$media = $tamanho/$seqs;

# Passo 3: Imprimir resultado

print "\nExemplo 01:\n";
print "Media de tamanho = $media\n";

exit;
```

ESTRUTURAS DE CONTROLE

Repetição

Problema:

Calcular o tamanho médio de sequências armazenadas em um array com milhares de elementos.

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. *for*

2. *foreach*

3. *while*

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

3. *while*

ESTRUTURAS DE CONTROLE - LOOPS

FOR

Script: [loops.pl](#)

Exemplo 02 (implementar):

```
# script para testar loop com for  
# exemplo02  
print "\nExemplo 02:\n";  
for ($i = 0; $i < 5; $i++) {  
    print "$i\n";  
}  
exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR

Script: [loops.pl](#)

Exemplo 03, copiar da página:

```
# for() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

$seqs = @seqs; #numero de elementos do array

print "\nExemplo 03:\n";

print "Numero de sequencias = $seqs\n";

for ( $i = 0; $i < $seqs; $i++ ) {

    print ">Gene$i\n$seqs[$i]\n\n" ;

}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR

Script: [loops.pl](#)

Exemplo 04, copiar o bloco "for" e modificar para calcular a média de tamanho

```
@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");  
$seqs = @seqs; #numero de elementos do array  
  
print "\nExemplo 04:\n";  
  
print "Numero de sequencias = $seqs\n";  
  
for ( $i = 0; $i < $seqs; $i++ ) {  
    # bloco para calcular tamanho  
}  
  
$media = #calcular media  
  
print "Media de tamanho = $media\n";  
  
exit;
```


ESTRUTURAS DE CONTROLE - LOOPS

FOR

Script: [loops.pl](#)

Exemplo 04, copiar o bloco "for" e modificar para calcular a média de tamanho

```
@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");  
$seqs = @seqs; #numero de elementos do array  
  
print "\nExemplo 04:\n";  
  
print "Numero de sequencias = $seqs\n";  
  
for ( $i = 0; $i < $seqs; $i++ ) {  
    $tamanho += length($seqs[$i]);  
}  
  
$media = $tamanho/$seqs;  
  
print "Media de tamanho = $media\n";  
  
exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

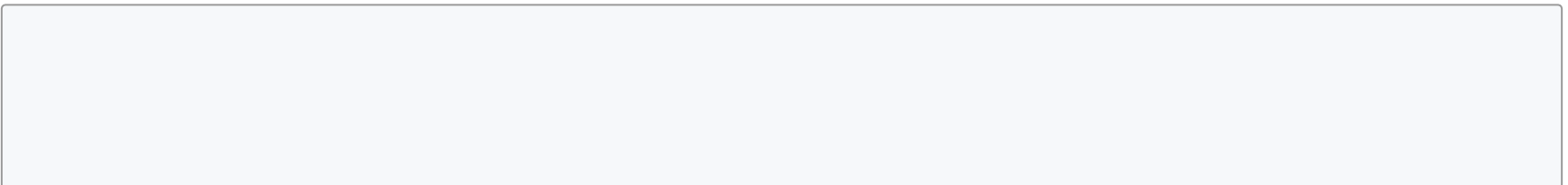
1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

3. *while*



ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](https://www.perl.com/doc/perl/loops.pl)

Exemplo 05, copiar da página

```
# foreach() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

###

print "\nExemplo 05:\n";

foreach $seq(@seqs) {
    print "$seq\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 06, modificar o exemplo05 para imprimir ID sequencial para cada sequencia("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCCGAT", "GATAGAG");

###

print "\nExemplo 06:\n";

foreach $seq(@seqs) {
    print "$seq\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 06, modificar o exemplo05 para imprimir ID sequencial para cada sequencia ("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays

print "\nExemplo 06:\n";

$i = 1;

foreach $seq(@seqs) {
    print ">Gene$i\n$seq\n\n";
    ++$i;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH = FOR

Script: [loops.pl](#)

Exemplo 06, modificar o exemplo05 para imprimir ID sequencial para cada sequencia ("Gene1","Gene2","Genen")

```
# foreach() para imprimir conteudo de arrays

print "\nExemplo 06:\n";

$i = 1;

for $seq(@seqs) {

    print ">Gene$i\n$seq\n\n";
    ++$i;

}

exit;
```

ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimi-las?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 07, copiar da página

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 07:\n";

#comandos

exit;
```


ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 07

```
# foreach() para imprimir conteudo de hashes

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 07:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como imprimir as sequências de forma ORDENADA?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 08, modificar exemplo 07

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 08:\n";

@genes = keys(%seqs);

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 08, modificar exemplo 07

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 08:\n";

@genes = sort(keys(%seqs));

foreach $gene(@genes) {
    print ">$gene\n$seqs{$gene}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

A variável `$_`

```
# foreach() para imprimir conteudo de hashes ordenado

%seqs = ("Gene25", "ATGGCGTAGATCG",
         "Gene49", "TAAGCCCCGGTATATTTGACCCCGAT",
         "Gene03", "GATAGAG");

print "\nExemplo 08:\n";

@genes = sort(keys(%seqs));

foreach (@genes) {
    print ">$_\n$seqs{$_}\n\n" ;
}

exit;
```

ESTRUTURAS DE CONTROLE

Hashes

Problema:

Se as sequências estiverem armazenadas em hashes (ID => sequência), como encontrar uma sequência específica? Por exemplo, a sequência *GATAGAG* está presente no hash?

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

A sequência GATAGAG está presente no hash?

```
# foreach() para encontrar um valor específico em hashes  
print "\nExemplo 09:\n";  
$seq_alvo = "GATAGAG"; # esta presente?  
foreach $gene (keys(%seqs)) {  
    #comandos  
}  
exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOREACH

Script: [loops.pl](#)

Exemplo 09, copiar da página

```
# foreach() para encontrar um valor especifico em hashes

print "\nExemplo 09:\n";

$seq_alvo = "GATAGAG"; # esta presente?

foreach $gene (keys(%seqs)) {
    if ($seqs{$gene} eq $seq_alvo) {
        print "Sequencia encontrada: $gene.\n\n";
    }
}
exit;
```


ESTRUTURAS DE CONTROLE - LOOPS

for, foreach, while

1. *for*

```
for (inicializacao, teste, modificacao) {  
    bloco de comandos  
}
```

2. *foreach*

```
foreach $escalar(@array) {  
    bloco de comandos  
}
```

3. *while*

```
while (teste) {  
    bloco de comandos  
}
```

ESTRUTURAS DE CONTROLE - LOOPS

WHILE

Script: [loops.pl](#)

Exemplo 10, copiar da página

```
$contagem = 10;

while($contagem > 0){
    print("$contagem\n");

    # contagem regressiva
    $contagem--;

    # pausar o script por 1s
    sleep(1);

    if($contagem == 0){
        print("Feliz Ano Novo!\n");
    }
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

WHILE

Script: [loops.pl](#)

Exemplo 10, copiar da página

```
$contagem = 10;

while($contagem){
    print("$contagem\n");

    # contagem regressiva
    $contagem--;

    # pausar o script por 1s
    sleep(1);

    if($contagem == 0){
        print("Feliz Ano Novo!\n");
    }
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

WHILE

Script: [loops.pl](#)

Exemplo 10, copiar da página

```
$contagem = 10;

while(1){
    print("$contagem\n");

    # contagem regressiva
    $contagem--;

    # pausar o script por 1s
    sleep(1);

    if($contagem == 0){
        print("Feliz Ano Novo!\n");
    }
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

WHILE ou UNTIL?

Script: [loops.pl](#)

Exemplo 11, adivinhar o número secreto

```
$n_secreto = 9;
$palpite;

# Loop 'until' ate adivinhar o numero secreto
until ($palpite == $n_secreto) {
    print "Digite o seu palpite: ";
    $palpite = <STDIN>; # Ler a entrada
    chomp($palpite);    # Remover newline

    # Verificar se o palpite está correto
    if ($palpite == $n_secreto) {
        print "Parabéns! Você adivinhou o número secreto.\n";
    } else {
        print "Errado! Tente novamente.\n";
    }
}
exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR vs WHILE

Script: [loops.pl](#)

Exemplo 12, **for** e **while** para a mesma tarefa

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

FOR vs WHILE

Script: [loops.pl](#)

Exemplo 12, **for** e **while** para a mesma tarefa

```
# for vs. while

# for
for ($i = 0; $i < 5; $i++) {
    print "$i\n";
}

# while
$i = 0;

while ($i < 5) {
    print "$i\n";
    ++$i;
}

exit;
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 13, copiar da página

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# for
$seqs = @seqs;

print "\nExemplo 13 (for):\n";

for ( $i = 0; $i < $seqs; $i++ ) {
    print "$seqs[$i]\n" ;
}
```


ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 14, continuar

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# foreach

print "\nExemplo 14 (foreach):\n";

foreach ... {
    ... ;
}
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 14, continuar

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# foreach

print "\nExemplo 14 (foreach):\n";

foreach $seq(@seqs) {
    print "$seq\n" ;
}
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 15, continuar

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# while
$n_seqs = @seqs;

$i = 0;

print "\nExemplo 15 (while):\n";

while ... {
    ...;
}
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 15, continuar

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# while
$n_seqs = @seqs;

$i = 0;

print "\nExemplo 15 (while):\n";

while ( $i < $n_seqs ) {
    print "$seqs[$i]\n";
}
```

ESTRUTURAS DE CONTROLE - LOOPS

Acessando elementos de um array

Script: [loops.pl](#)

Exemplo 15, continuar

```
# formas de imprimir arrays

@seqs = ("ATGGCGTAGATCG", "TAAGCCCCGGTATATTTGACCCT", "GATA

# while
$n_seqs = @seqs;

$i = 0;

print "\nExemplo 15 (while):\n";

while ( $i < $n_seqs ) {
    print "$seqs[$i]\n";
    ++$i;
}
```

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Usando loop em Perl, crie um array com os números de 10 a 900, contando de 10 em 10.

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Crie um script que vai contar o número de bases G em uma sequência de DNA.

Solução 01

```
#!/usr/bin/perl

#Script para contar a frequencia da base G em uma sequencia de DNA

$DNA = "GATTACACGAT";

$countG = 0;

#criar um array com $DNA
@sequence = split('', $DNA);

#Para cada letra da sequencia, checar se é igual a G
#caso 'verdadeiro' aumentar $countG
foreach(@sequence){
    if($_ eq "G"){
        $countG++;
    }
}

#imprimir o número de Gs
print "Ha $countG bases G\n";
exit;
```


Solução 02

```
#!/usr/bin/perl

#Script para contar a frequencia da base G em uma sequencia de DNA

$DNA = "GATTACACGAT";

$countG = 0;
$currentPos = 0;

#descobrir o tamanho do $DNA
$DNAlength = length($DNA);

#Para cada letra da sequencia, checar se é igual a G
#caso 'verdadeiro' aumentar $countG
while($currentPos < $DNAlength){
    $base = substr($DNA,$currentPos,1);
    if($base eq "G"){
        $countG++;
    }
    $currentPos++;
}

#imprimir o número de Gs
print "Ha $countG bases G\n";
exit;
```

ESTRUTURAS DE CONTROLE

for, foreach, while

Exercício:

Crie um script que vai contar o número de bases G ou C em uma sequência de DNA.