1. 编写一个Perl程序（程序名：tphw1.pl），构建一个名为%geneticCodes哈 希数据结构，键（key）和值（value）对应如下： “UUU”对应“Phe”， “UUA”对应“Leu”， “GGA”对应“Gly” 。最终遍历这个哈希，输出一个名 为geneticHash.txt的文件1。

use strict;

use warnings;

my %geneticCodes = ("UUU","Phe","UUA","Leu","GGA","Gly");

my @codes = keys %geneticCodes;

open (OUTPUT, ">D:/geneticHash.txt")||die "can't open";

foreach my $code (@codes)

{

print OUTPUT "the $code is ", "$geneticCodes{$code}", "\n";

}

close (OUTPUT);

1. 编写一个Perl程序（程序名：tphw2.pl ) , 根据密码子哈希表 %codonTable ，统计并打印输出每个氨基酸有多少个密码子编 码到aminoAcidNum.txt文件。 提示：可用values获得哈希表values的数组、构建一个新的哈希 表（如:%aminoAcidTable）去统计每个氨基酸的数目。 如：$ aminoAcidTable{„F‟}++, 表示„F„的数目加１。利用 foreach和keys去输出%aminoAcidTable的结果。 输入： %codonTable 输出：氨基酸和对应的密码子数目到 aminoAcidNum.txt文件。

my %codonTable = (

"TTT" => 'F',

"TTC" => 'F',

"TTA" => 'L',

"TTG" => 'L',

"TCT" => 'S',

"TCC" => 'S',

"TCA" => 'S',

"TCG" => 'S',

"TGG" => 'W');

my @vals =values %codonTable;

my %aminoAcidTable =();

foreach my $val (@vals)

{

$aminoAcidTable{$val}++;

}

foreach my $ key(keys %aminoAcidTable)

{

print $key, "\n",$aminoAcidTable{$key} , "\t";

}

1. 根据存有名字的数组 @name数 组和存有电话号码的@phone数组（两个数组一一对应），去构建一个名为 %phoneBook的哈希数据结构，键（key）为名字，值（value）为电话号码。 然后在电话号码本里查找“zhaoliu”的电话号码，如果存在则输出名字和电 话号码 。 提示：可用for循环同时遍历名字和电话号码数组，然后把对应的名和号码存 到phoneBook。最后用exists判断zhaoliu是否存在。 输入：@name和@phone数组 输出：输出zhaoliu的电话号码 （ zhaoliu ： 134234568379 ）。

@name = ("zhangsan", "lisi", "wangwu", "zhaoliu", "sunqi", "zhouba");

@phone = ("135234567369", "134234567368", "135235567367", "134234568379","135244567969", "135234567369");

my % phoneBook =();

for (my $name = 0; $name < 6; $i++)

{

$phoneBook{$name[$i]}=$phonr[i];

}

if(exist $phoneBook{"zhaoliu"})

{

print $phoneBook{"zhaoliu"},"\n"

}

my$seq ="AUGAUGGCCAUGGCGCCCAGAACUGAGAUCAAUAGUACCCGUAUUAACGGGUGA";

my $seqlen =length($seq);

my $i=0;

my $rnalen=3;

for (my $i= 0; $i<$seqlen-$rnalen;$i+=$rnalen)

{

my $str=substr($seq,$i,3);

print $str,"\n";

}

my %codonTable = (

"UUU" => 'F',

"UUC" => 'F',

"UUA" => 'L',

"UUG" => 'L',

"UCU" => 'S',

"UCC" => 'S',

"UCA" => 'S',

"UCG" => 'S',

"UAU" => 'Y',

"UAC" => 'Y',

"UAA" => '0',

"UAG" => '0',

"UGU" => 'C',

"UGC" => 'C',

"UGA" => '0',

"UGG" => 'W',

"CUU" => 'L',

"CUC" => 'L',

"CUA" => 'L',

"CUG" => 'L',

"CCU" => 'P',

"CCC" => 'P',

"CCA" => 'P',

"CCG" => 'P',

"CAU" => 'H',

"CAC" => 'H',

"CAA" => 'Q',

"CAG" => 'Q',

"CGU" => 'R',

"CGC" => 'R',

"CGA" => 'R',

"CGG" => 'R',

"AUU" => 'I',

"AUC" => 'I',

"AUA" => 'I',

"AUG" => 'M',

"ACU" => 'T',

"ACC" => 'T',

"ACA" => 'T',

"ACG" => 'T',

"AAU" => 'N',

"AAC" => 'N',

"AAA" => 'K',

"AAG" => 'K',

"AGU" => 'S',

"AGC" => 'S',

"AGA" => 'R',

"AGG" => 'R',

"GUU" => 'V',

"GUC" => 'V',

"GUA" => 'V',

"GUG" => 'V',

"GCU" => 'A',

"GCC" => 'A',

"GCA" => 'A',

"GCG" => 'A',

"GAU" => 'D',

"GAC" => 'D',

"GAA" => 'E',

"GAG" => 'E',

"GGU" => 'G',

"GGC" => 'G',

"GGA" => 'G',

"GGG" => 'G',

);

print $codonTable{my$str};

my @codes = keys % codonTable