Name:Group6 Discussion 5 CSI 311

For answer to the following questions please refer to the following documentation:

Sleep programming language

http://sleep.dashnine.org/manual/

1. The "What is sleep?" main page says "Sleep is a Java-based scripting language heavily inspired by Perl". (5points each)
   1.1. What features of Perl do you see in the sleep programming language?

   Sleep variables are scalars. Scalar variable names always begin with the dollar sign. Perl expects variable names to begin with a dollar sign as well.

   Perl is useful for text and data processing and sleep excels at taking input and extracting info fr om it.

   Like the Perl language, it has more convenient programming elements such as generic variables, dynamic arrays, and hash tables provided directly.

   1.2. What are some of the differences from perl?

   Perl does not allow Spaces between morphemes, but sleep must adhere strictly to the specification.

   1.3. What are the advantages of making Sleep a scripting language over a compiled language?

   Compiled languages are used to build applications from scratch. Whereas, scripting languages c an combine existing modules or components, so, scripting languages allows programs to be accesse d and utilized by other programs.

   Sleep is simple, easy to learn, easy to use, and easy to combine with the program. Its language s yntax is similar to Perl scripts. Sleep scripts can access, create, and query Java objects, and parsed sc ripts can be serialized.

   The syntax is simpler, not translated into machine language, but directly executed by the script language environment.

2. What is the purpose of the statement .= in sleep? (10 point)

   Connect two or more strings. It splices the right parameter to the back of the left parameter an d assigns the value to the left parameter.

3. Please describe an instance of where you might be able to use a Sleep script to accomplish some task that you wouldn't want to do by hand? (5 points each)
   3.1. What is the task?

   I will use SLEEP language to do some simple automation work. For instance, count the total number of characters in a text file and find whether a particular character or string appears and how many times it appears.

   3.2. What are the benefits of using Sleep?

   As we know, Sleep is a scripting language. So it has the advantages that come from that of scripting languages.

   First, it's easy to code. Scripting languages provide higher flexibility, so does Sleep. So I'll not care about the complex static type checking when I try to code the algorithm. I can care more about

the higher-level details. In addition, Sleep doesn't have compilation, linking, etc. It's easier for me to debug the program.

Second, Sleep is more lightweight. This feature allows me to embed it to other programs or applications easily. That's useful to add some specific automation functionality to the current system. Like the task I mentioned above, It's a specific automation functionality. I can embed it to the system that is written in another language easily.

Finally, Sleep can access to the Java class library. And, I also have the option of generating the Java Doc API's for the sleep language. So it would be helpful to interact with Java-written Programs and get some advantages from Java language.

3.3. Please write the algorithm to accomplish the task.

For the algorithm to finish this task, I defined several steps to implement the algorithm.

First, used file system to get the target file. I called the lof() function to get the size of the file.

Second, I read the file content. I used an array to hold the string of the txt file content.

Then, I used a loop to traverse the array to search the specified character or string and maintained 2 variables to check the existence and the count times respectively.

4. Write a sleep program that opens a binary file. The binary file is stored as 4 byte integers on a line. The program should read contents of the file and covert it to hexadecimal values. The program should write the hexadecimal values out to new file. (20 points)

4.1. (10 points) Compile the program and run it. Take a screen shot of the output of your program.

```
C:\Users\lenovo>java -jar sleep.jar BinaryConvertion.txt
Binary: 00000000000000000000101000000101
Hexadecimal: a05
```


```
$handle = openf("test.txt");

$binary = readb($handle, 32);

closef($handle);
```

```
$handle = openf("> $^ test2.txt");

$hexadecimal = formatNumber($binary, 2, 16);

writeb($handle,$hexadecimal);

closef($handle);


println("Binary: $binary");

print("Hexadecimal: $hexadecimal");
```

5. Write a sleep program that does the following: (20 points)

The program will first prompt the user for a range of numbers. Then the game will generate a random number in that range. The program will then allow the user to guess 3 times. Each time the person takes a guess, if it is not the number then the game should tell them to guess higher or lower. They only get 3 guesses. If they have not guessed the number in three tries then the game should tell them the computer won and thank you for playing and ask them if they would like to play again. If the user wins then the game should congratulate them and ask them if they would like to play again. Each time the program repeats you should store the random number generated into an array. When the program is done the program should display the all of the random numbers back to the user.

5.1. (10 points) Compile the program and run it. Take a screen shot of the output of your program.

```
$n = 0;
srand(ticks());
# srand(): Set random seek
# ticks(): Return the current time in milliseconds.
while(true)
{
 println("Guess a number between 0 to 9");
 println("You have 3 chance to guess");

 $number = rand(9);
 for($i = 3;$i > 0;$i--)
 {
  $flag = 0;
  $guess = readln();
  if ($guess == $number)
```

```
  {
    println("You got it!! You must be psychic");
    $flag = 1;   # The player won
    break;
  }
  else if ($guess < $number)
  {
    println("Too low... you lose!");
  }
  else if ($guess > $number)
  {
    println("Too high, you still lose");
  }
 }

 @array[$n] = $number;
 $n = $n + 1;
 if($flag == 0)
 {
  println("The computer won and thank you for playing!");
 }
 else
 {
  println("You won the game and thank you for playing!");
 }
 println("Do you want to play again?");
 print("[y/n]: ");
 $input = readln();
 if($input eq "n")
 {
  break;
 }
}
println("The answer for guessing the number is: ");
println(@array);
```

```
C:\Users\Admin\Desktop>java -jar sleep.jar guessGame.txt
Guess a number between 0 to 9
You have 3 chance to guess
5
Too high, you still lose
2
Too high, you still lose
0
You got it!! You must be psychic
You won the game and thank you for playing!
Do you want to play again?
[y/n]: y
Guess a number between 0 to 9
You have 3 chance to guess
5
Too low... you lose!
7
Too high, you still lose
6
You got it!! You must be psychic
You won the game and thank you for playing!
Do you want to play again?
[y/n]: y
Guess a number between 0 to 9
You have 3 chance to guess
5
Too high, you still lose
3
Too high, you still lose
1
Too high, you still lose
The computer won and thank you for playing!
Do you want to play again?
[y/n]: n
The answer for guessing the number is:
@(0, 6, 0)
```