page 1 page 2 page 3 Total / 31

Please print clearly :

Name :	
Login :	@ucsc.edu

No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Define a function **`grep`** in Scheme whose first argument is a predicate and second argument a list. It returns a new list of all of the elements of the argument list for which the predicate is true. Do not use a higher-order function. **[2✓]**

2. Define a function **`positives`** which uses **`grep`** (above) to accept a list of numbers and return only the positive ones. Use a lambda expression, not a function name, as one of its arguments. **[1✓]**

3. Define a function **`sum`** which returns the sum of all elements of a list. Do not use any higher-order functions. Use constant ($O(1)$) stack space. **[2✓]**

4. Define the same function **`sum`** using the higher-order function **`foldleft`**, whose arguments are a unit value, a binary function, and a list. **[1✓]**

5. Name the two kinds of ***universal polymorphism*** and give an example of each. **[2✓]**

6. Name the two kinds of ***ad hoc polymorphism*** and give an example of each. **[2✓]**

7. Write a complete program in Perl that reads in words from all of the filenames specified in `@ARGV`, and after end of file of the last file, prints out each word, one per line, sorted lexicographically, with each word followed by a count of the number of times it occurred. Exit status is 0 for success and 1 if any file could not be opened. Print suitable error messaages for files that could not be opened. If no files are specified, read the standard input. Do not use the `<>` operator. Use `/\w+/` to match words or `/\W+/` to split delimiters. If `@ARGV` is empty, use `"-"` (standard input). **[5✔]**

```
#!/usr/bin/perl
use strict;
use warnings;
$0 =~ s|.*/||;
```

8. Define the function `zip` in Scheme. It takes a function and a pair of lists and returns a list having pairwise applied the function to corresponding pairs of elements in the list. If the lists are of unequal length, the function stops when the shorter list runs out. **[2✔]**
   Example: `(zip + '(1 2 3) '(4 5 6 7))`
   Returns: `(5 7 9)`

9. Define the same function in Perl. It takes a reference to a function of two arguments, followed by ***references*** to two arrays and applies the function pairwise to the arrays, returning an array of the shorter length. **[2✔]**
   Example: `@y = zip sub{$_[0]+$_[1]}, [1, 2, 3], [4, 5, 6, 7]; print "@y\n";`
   Prints: `5 7 9`

10. Write a program in Perl which prints out all of the command line arguments (but not the program name) on one line, each separated from the next with one space, but does not print a space in the front or at the end. It finishes by printing a newline character. **[1✔]**

```
#!/usr/bin/perl
```

Multiple choice. To the ***left*** of each question, write the letter that indicates your answer. Write ***Z*** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✔]**

number of correct answers		× 1 =	= a
number of wrong answers		× ½ =	= b
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		= c

1. Which of these languages evaluates expressions in a lazy manner by default ?
   (A) Haskell
   (B) Lisp
   (C) Perl
   (D) Scheme

2. Which of the following C operators is "lazy" ?
   (A) `&&`
   (B) `++`
   (C) `==`
   (D) `>>`

3. What is the running time of the following function ?

```
(define (f n)
   (if (< n 2) n
       (+ (f (- n 1)) (f (- n 2)))))
```

   (A) $O(n)$
   (B) $O(\log_2 n)$
   (C) $O(n^2)$
   (D) $O(2^n)$

4. What amount of stack space is taken up by evaluating that same function ?
   (A) $O(n)$
   (B) $O(\log_2 n)$
   (C) $O(n^2)$
   (D) $O(2^n)$

5. What is the value of :

```
(cadr '((1 2 3) (4 5 6) (7 8 9)))
```

   (A) `1`
   (B) `(4 5 6)`
   (C) `(2 3)`
   (D) `((7 8 9))`

6. In C, when is a `static` local variable bound to a particular virtual address ?
   (A) When the program is compiled.
   (B) When the program is linked.
   (C) When the program is loaded into execution by the `exec` system call.
   (D) When the function in which it is declared is called.

7. Data types in Perl and Scheme are :
   (A) strong and dynamic.
   (B) strong and static.
   (C) weak and dynamic.
   (D) weak and static.

8. A static link in a stack frame points at :
   (A) the address of the call instruction that activated this function.
   (B) the stack frame of the calling function.
   (C) the stack frame of the function in which this function is nested.
   (D) the top of the function call stack.

9. Assuming only pure Java code with no sneaky tricks written in C, If M = memory leaks, D = dangling references, and U = unsafe type conversions or casting, which of the following are possible in Java ?
   (A) all of them.
   (B) none of them.
   (C) only D, but neither M nor U.
   (D) only M, but neither D nor U.

10. Given the C declaration `char **p;`, which expression is invalid ?
    (A) `&(&p)`
    (B) `&(*p)`
    (C) `*(&p)`
    (D) `*(*p)`

11. John Backus, 1953, IBM 704.
    (A) `ALGOL`
    (B) `COBOL`
    (C) `FORTRAN`
    (D) `LISP`