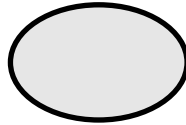


Total / 30



Please PRINT using keyboard letters :

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. **Scheme :** The following function by filius Bonacci takes $O(2^n)$ time and $O(n)$ stack. Recode it using a helper function so that it takes $O(n)$ time and $O(1)$ stack. **[2✓]**
`(define (f n) (if (< n 2) n (+ (f (- n 1)) (f (- n 2)))))`

2. **Scheme :** Define the function `foldl`, which has three arguments in this order : a function, a unit, and a list. A single result is returned such that the function is applied to the unit and the `car` of the list, sending that down as the next unit, and the `cdr` of the list is sent in on each recursive call. It is tail recursive. **[2✓]**

Using `foldl`, define the function `length`, which returns the length of a list. **[1✓]**

3. **Scheme :** Without using any higher-order functions, define the function `map` whose first argument is a function, which is applied to each element of its second argument, a list. The result returned as a new list. **[2✓]**

```
> (map (lambda (x) (* x x)) '(1 2 3 4 5))
(1 4 9 16 25)
```

4. **C++ :** Define a function `print` whose argument is a `vector<string>`, and which uses an iterator to print out all of the elements, one per line. **[2✓]**

Instead, for an extra point, make it a template function whose type can be anything for which `operator<<` is properly defined. **[1✓]**

5. **Scheme**: Assuming you have a hash table named `hash`, which contains pointers to binary functions, Write code to evaluate an expression tree. The tree is represented as three element lists, where `car` is the function, `cadr` is the left operand, and `caddr` is the right operand. Leaf nodes are symbols whose values are in the table `syntab`, also a hash. You need not do error checking: all variables and operators are valid, all leaf nodes are symbols, and all interior nodes have exactly three elements. **[3✓]**

```
(define (evaltree tree)
```

6. **Perl**: Define a function `fold`, which uses a loop to join all elements of an array into a single value. Its first argument is a reference to a binary function, its second argument is a unit value, and its third argument is a reference to an array. **[2✓]**

```
print fold sub{$_[$_]+$_[1]}, 0, [1, 2, 3, 4, 5, 6];  
21
```

7. **Scheme**: Define a function `split` that takes a single list argument and returns a list of two elements, with the `car` of the result containing the odd numbered elements (counting from 1) `cdr` the even numbered elements. The elements remain in the same relative order. **[3✓]**

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

8. **Scheme**: Without using any higher order functions, code `reverse`, which reverses a list. Hint: Think about two stacks. Use tail recursive form and a helper function that uses accumulator style. **[2✓]**

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		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total $c = \max(a - b, 0)$	11		$= c$

- What expression will print (3) ?
(A) `(caar '(1 2 3))`
(B) `(cadr '(1 2 3))`
(C) `(cdar '(1 2 3))`
(D) `(cddr '(1 2 3))`
- What variable in a command in a **Makefile** has as its value the target of the recipe ?
(A) `$*`
(B) `$<`
(C) `$?`
(D) `$@`
- What might cause a dangling pointer in C++?
(A) `catch`
(B) `delete`
(C) `new`
(D) `throw`
- In Ocaml, what is the type of the function (+) ?
(A) `int * int * int`
(B) `int * int -> int`
(C) `int -> int * int`
(D) `int -> int -> int`
- Fold right (`foldr`) uses up how much stack space on a list of length n ?
(A) $O(1)$
(B) $O(2^n)$
(C) $O(\log_2 n)$
(D) $O(n)$
- In the following statement, `continue` goes to what part ?

```

for (i = 0; i < n; ++i) {
    f (); continue; g ();
} h ();

```


(A) `++i`
(B) `h ();`
(C) `i < n`
(D) `i = 0`
- What declares a variable `m` of type `map` whose keys are `strings` and whose values are `ints` ?
(A) `m: (string, int) map;`
(B) `map <string, int> m;`
(C) `map [int, string] m;`
(D) `map m <string, int>;`
- Both Perl and Scheme have type systems which are :
(A) strong and dynamic.
(B) strong and static.
(C) weak and dynamic.
(D) weak and static.
- For which function will C++ say: *Warning: Returning a reference to a local variable or temporary.*
(A) `int &f (int &i) {int j; return j; }`
(B) `int f (int &i) {int j; return j; }`
(C) `int f (int i) {int j; return j; }`
(D) `int f (int i) {int &j; return j; }`
- In C++, which of these operators is “lazy” ?
(A) `&&`
(B) `++`
(C) `<<`
(D) `==`
- In Perl, the default variable acted on when nothing is specified, e.g., for `chomp` or pattern matching, is :
(A) `$!`
(B) `$0`
(C) `$_`
(D) `@_`