

Midterm Review

CSCE 322

Name: _____

Instructions

Please solve the problems presented below. **Show your work to receive full credit; just an answer is not enough. No Approximations.**

Question 1 ()

Explain the connection between short-circuit Boolean expressions and normal-order evaluation in functional programming.

Question 2 ()

What does the following Haskell program compute?

mystery.hs

```
mystery :: Integer -> Integer
```

```
mystery 0 = 0
```

```
mystery n = 2 * n - 1 + (mystery (n-1))
```

Question 3 (points)

In the code below, the query `?- classmates(jane_doe, X)` will succeed three times: twice with `X = jane_doe`, and once with `X = ajit_chandra`. Show how to modify the `classmates(X, Y)` rule so that a student is not considered a classmate of him or herself.

```
takes(jane_doe, his201).  
takes(jane_doe, cs254).  
takes(ajit_chandra, art302).  
takes(ajit_chandra, cs254).  
classmates(X, Y) :- takes(X, Z), takes(Y, Z).
```

Question 4 (points)

What does the following Prolog program compute?

mystery01.pl

```
% ** is raising to a power
mystery(0,1).
mystery(A,B) :- 0 is mod(A,2) , C is A / 2 , mystery(C,D) , B is D ** 2 , !.
mystery(E,F) :- 1 is mod(E,2) , G is E - 1 , mystery(G,H) , F is H * 2 , !.
```

Question 5 (10 points)

Given the following code, what values of `Nebraska` will let the query
`?- mystery3([u,n,l],[u,n,o],Nebraska). succeed?`

mystery03.pl

```
mystery3([],A,A).  
mystery3([B|C],D,E):-  
    member(B,D), % member is true if B is in list D  
    !,  
    mystery3(C,D,E).  
mystery3([F|G],H,[F|J]):-  
    mystery3(G,H,J).
```

Question 6 (10 points)

Modify the code provided below so that the goal `path(X,Y)` for arbitrarily already-instantiated `X` and `Y` will succeed no more than once, even if there are multiple paths from `X` to `Y`.

```
edge(a,b).
```

```
edge(b,c).
```

```
edge(c,d).
```

```
edge(d,e).
```

```
edge(b,e).
```

```
edge(d,f).
```

```
path(X,X).
```

```
path(X,Y):-edge(Z,Y),path(X,Z).
```

Question 7 (10 points)

Given the following code, what was the input to `mystery` if the output was "phobiarebyc"?

mystery04.hs

```
mystery :: [Char] -> [Char]
mystery [] = []
mystery (b:bs) = helper (b:bs) 5

helper :: [Char] -> Int -> [Char]
helper c 0 = c
helper [] _ = []
helper (d:ds) e = (helper ds (e-1)) ++ [d]
```


Question 8 ()

Given this Haskell function, what input would produce the output (3,0,1,0,2)?

mystery02.hs

```
1  mystery :: Integer -> (Integer , Integer , Integer , Integer , Integer )
2  mystery n
3      | n >= 50 = (a+1,b,c,d,e)
4      | n >= 20 = (f,g+1,h,j,k)
5      | n >= 10 = (m,p,q+1,r,s)
6      | n >= 5  = (t,u,v,w+1,x)
7      | otherwise = (0,0,0,0,n)
8      where (a,b,c,d,e) = mystery (n-50)
9             (f,g,h,j,k) = mystery (n-20)
10            (m,p,q,r,s) = mystery (n-10)
11            (t,u,v,w,x) = mystery (n-5)
```

Question 9 ()

Given this Haskell function, and the input 5 [9,8,16,16,4,10,9,13], the result is **False**.
How many times is **d** evaluated during the computation of that result?

mystery03.hs

```
1 mystery2 :: Ord b => b -> [b] -> Bool
2 -- Ord is a type that defines ordinality (ordering).
3 -- Ord allows for <,>,etc.
4 -- Bool is the type for Boolean values
5 mystery2 _ [] = False
6 mystery2 a (x:xs)
7     | a == x = True
8     | a < x = mystery2 a c
9     | otherwise = mystery2 a d
10    where c = [c | c<-xs, c<x]
11          d = [d | d<-xs, d>x]
```

Question 10 (10 points)

Given this Haskell function, if the output of the function was 15, and the input to the function was $(D, 4)$, what was the value of D ?

mystery114501.hs

```
1 mystery :: (Integer,Integer) -> Integer
2 mystery (a,b)
3     | a == b = 1
4     | b == 1 = a
5     | otherwise = (mystery (a-1,b-1)) + (mystery (a-1,b))
```

Question 11 (13 points)

Given this Haskell function, provide one combinations of values for `input01` and `input02` that could cause `mystery 2 input01 input02` to return the output `Indinol`?

1145mysteryFinal.hs

```
1 mystery :: Int -> Char -> [Char] ->[Char]
2 mystery 0 - a = a
3 mystery - - [] = []
4 mystery b c (d:ds)
5         | c == d      = mystery (b-1) c ds
6         | otherwise   = [d] ++ (mystery b c ds)
```

Question 12 (13 points)

Given these Prolog predicates, `mystery([2,0,0,2]).` evaluates to `true`, `mystery([2,0,0,4]).` evaluates to `false` and `mystery([r,X,n,n,e,r]).` will evaluate to `true` when `X` is unified with `e`.

How many attempts are made to match `helper2` during the evaluation of `mystery([a,l,a,s,k,a]).`?

mystery1145final01.pl

```
1  mystery ([ ]): -!.
2  mystery (A): -
3      length (A,1) ,
4      !.
5  mystery ([B|C]): -
6      helper (C,B) ,
7      helper2 (C,D) ,
8      mystery (D) .
9
10 helper ([E] ,E): -!.
11 helper ([_ |F] ,G): -
12     helper (F,G) .
13
14 helper2 ([_] ,[]): -!.
15 helper2 ([J|K] ,[J|M]): -
16     helper2 (K,M) .
```

Question 13 (13 points)

Given these Prolog predicates, `repeat([a,b],2,[a,a,b,b]).` evaluates to `true`,
`repeat([a,b],1,[a,a,b,b]).` evaluates to `false` and
`repeat([[g,o],[b,i,g],[r,e,d]],3,Result).` will evaluate to `true` when `Result`
is unified with `[[g,o],[b,i,g],[r,e,d]],[g,o],[b,i,g],[r,e,d]],[g,o],[b,i,g],[r,e,d]]`.
How many times is `repeat([],_,[]).` successfully matched during the evaluation?

mystery1145final02.pl

```
1 repeat([],_,[]):-
2     writeln(repeat).
3 repeat(_,0,[]).
4 repeat([H|T],N,D):-
5     helper(H,N,A),
6     repeat(T,N,B),
7     append(A,B,D),
8     !.
9
10 helper([],_,[]).
11 helper(_,0,[]).
12 helper(C,E,F):-
13     G is E-1,
14     helper(C,G,H),
15     append([C],H,F).
```

Question 14 (10 points)

Given the following code, what would the output of `mystery` be if the input was
["New York","Los Angeles","Chicago","Dallas","Houston","Philadelphia","Los Alamos"]?

mystery114801.hs

```
mystery :: [[Char]] -> [[Char]]
mystery [] = []
mystery (a:b) = c ++ [a] ++ d
  where      c = mystery [e|e<=-b,(length e)<=(length a)]
            d = mystery [f|f<=-b,(length f)>(length a)]
```

Question 15 (10 points)

Given the following code, what would the input of `mystery02` be if the output was `["h","oooo","www","nn","b","r","c"]`?

mystery114802.hs

```
mystery02 :: Eq a => [a] -> [[a]]
mystery02 [] = []
mystery02 (b:c) = [(b:d)] ++ (mystery02 f)
  where d = [e | e<-c, e==b]
        f = [g | g<-c, g/=b]
```


Question 16 (12 points)

Given the following code, what is one value of `Input` will let the query
`?- mystery(Input,[s,a,s,a,t,c,e,w,a]).` succeed?

mystery1148.pl

```
num(4).
num(3).
num(2).
num(1).
```

```
mystery(In,Out):-
    num(Number),
    !,
    helper(In,Number,Out).
```

```
helper(A,N,B):-
    num(N),
    length(A,LA),
    LA < N,
    A = B,
    !.
```

```
helper(A,N,B):-
    num(N),
    P is N - 1,
    assistant(A,P,C),
    apprentice(A,N,D),
    helper(D,N,E),
    append(C,E,B).
```

```
assistant(_,0,[]):-!.
assistant([A|B],N,[A|C]):-
    num(N),
    P is N - 1,
    assistant(B,P,C).
```

```
apprentice(A,0,A):-!.
apprentice([_],1,[]):-!.
apprentice([_|A],N,B):-
    num(N),
    P is N - 1,
    apprentice(A,P,B).
```

Question 17 (10 points)

For the code below,

```

get ([Row|Rows] , Where , What):-
    length (Rows , RowsLength) ,
    0 is mod (RowsLength , 2) ,
    length (Row , Cols) ,
    getFRow (Row , RowWhere , What) ,
    Where is RowWhere + RowsLength * Cols .

get ([Row|Rows] , Where , What):-
    length (Rows , RowsLength) ,
    1 is mod (RowsLength , 2) ,
    writeln (executed) ,
    length (Row , Cols) ,
    getBRow (Row , RowWhere , What) ,
    Where is RowWhere + RowsLength * Cols .

get ([_ | Rows] , Where , What):-
    get (Rows , Where , What) .

getFRow ([What | _] , 1 , What) .
getFRow ([_ | Tail] , Where , What):-
    getFRow (Tail , TailWhere , What) ,
    Where is TailWhere + 1 .

getBRow ([What | T] , Where , What):-
    length ([What | T] , Where) .
getBRow ([_ | T] , Where , What):-
    getBRow (T , Where , What) .

```

For the query `?- get([[8,9,0],[7,8,5],[2,3,8]],Location,8).`, how many times is Line 10 executed?

Question 18 (12 points)

For the code below,

```

get ([Row|Rows] , Where , What):-
    length (Rows , RowsLength) ,
    0 is mod (RowsLength , 2) ,
    length (Row , Cols) ,
    getFRow (Row , RowWhere , What) ,
    Where is RowWhere + RowsLength * Cols .

get ([Row|Rows] , Where , What):-
    length (Rows , RowsLength) ,
    1 is mod (RowsLength , 2) ,
    writeln (executed) ,
    length (Row , Cols) ,
    getBRow (Row , RowWhere , What) ,
    Where is RowWhere + RowsLength * Cols .

get ([_ | Rows] , Where , What):-
    get (Rows , Where , What) .

getFRow ([What | _] , 1 , What) .
getFRow ([_ | Tail] , Where , What):-
    getFRow (Tail , TailWhere , What) ,
    Where is TailWhere + 1 .

getBRow ([What | T] , Where , What):-
    length ([What | T] , Where) .
getBRow ([_ | T] , Where , What):-
    getBRow (T , Where , What) .

```

For the query `?- get([[8,9,0],[7,8,5],[2,3,8]],Whe,Wha) .`, in what order are `Whe` and `Wha` given values if the user presses `;` to get all values that satisfy the query?