Midterm Examination Topics

This exam will cover material from the first four homework assignments. I realize that we have learned many functions, so this document will prioritize the functions learned into three tiers. Tier 1 consists of functions that you should know inside and out (including syntax) and be able to explain what they do. You should be so familiar with these functions that you could walk up to mathematica and use them without having to look them up. Tier 3 consists of functions that we have used at some point to do something interesting, but I don't expect to test you on them without giving you a reminder about what they do and how they work. Tier 2 falls somewhere in between tier 1 and tier 3. Here are those functions:

1 Tier 1 functions

PrimeQ	Sin	Cos	Tan	Sqrt	Head
Log	Exp	FactorInteger	Fibonacci	Flatten	Level
Range	Table	Map(/@)	Select	Length	List
Apply(@@)	Sum	Product	Union	Complement	Intersection
FullForm	TreeForm	Plus	Times	Rational	Power
Inner	Outer	Equal(==)	And(&&)	Or($\|\|$)	Not(!)
If	Which	Point	Line	Graphics	Graphics3D
Do	While	For	EvenQ	OddQ	RandomInteger

2 Tier 2 functions

N	ArcSin	ArcCos	ArcTan	Simplify	FullSimplify
Mod	Quotient	Expand	Factor	Together	Apart
Random	LCM	Divisible	First	Last	Drop
Take	Rest	Most	Print	IntegerDigits	Reverse
FromDigits	Divisors	Prime	Binomial	Join	Factorial
Floor	Ceiling	Min	Max	Append	AppendTo
\[Infinity](∞)	Increment(++)	+=, *=	Nest	NestList	Sequence
Fold	FoldList				

3 Tier 3 functions

| RootReduce	PrimePi	TrigExpand	Slider	Manipulate	Dynamic
Thread	TableForm	\[Element]($\in$)	ForAll	Exist	Rule
Implies	Tally	Piecewise	CoefficientList	NSum	NProduct
Input	NestWhile	NestWhileList	FixedPointList	LengthWhile	TakeWhile
Accumulate					

4   Types of problems

The midterm exam will look a lot like the quizzes. The following types of problems could be asked.

1. Produce an input that will calculate a given quantity.

2. Produce an input that will produce a specified output.

3. Produce an output associated with a specified input.

4. Match a set of inputs with a set of outputs.

5. Produce an input that will render a particular graphic.

6. Match a set of `Graphics` (or `Graphics3D`) inputs with a set of graphical outputs.

7. Given a function (pure or defined), one should be able to:

   (a) evaluate it at specified inputs.
   (b) explain what the output represents (in English) for a given input.

8. Given a function that returns `True` or `False`, one should be able to:

   (a) evaluate it at specified inputs.
   (b) determine precisely which inputs return `True`.

9. Write a function (using correct syntax) that returns a particular output.

5 Sample problems

1. Define a function called `Staircase` (using correct syntax) that takes a positive integer n as an argument and renders a staircase (beginning with an up-step and ending in a right-step) connecting $(0,0)$ to (n,n) by completing the expression below. Also shown below is the desired output for `Staircase[20]`.


```
Staircase[n_]:=
```
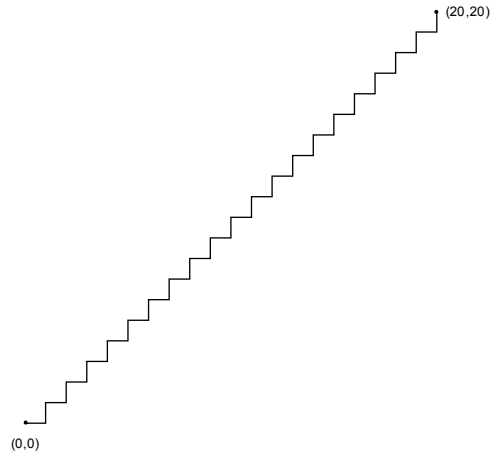
2. Suppose `j` is a previously defined variable and `j//TreeForm` outputs the following:
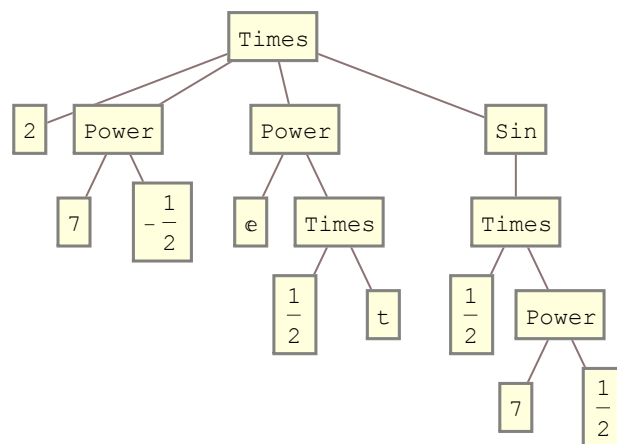
(a) Using standard mathematical notation, write what the variable `j` represents in Mathematica.

(b) Write down the output associated with following inputs involving `j`.

i. `FullForm[j]`

ii. `Length[j]`

iii. `Head[j]`

iv. `Take[j,3]`

v. `Level[j, {2}]`

vi. `Select[j, Length[#] == 1 &]`

vii. `Apply[Plus,j,1]`

viii. `Apply[Plus,j,2]`

ix. `Apply[Plus,j,3]`

x. `Apply[Plus,j,{1}]`

xi. `Apply[Plus,j,{2}]`

xii. `Apply[Plus,j,{3}]`

3. The following anonymous functions return either `True` or `False`. For each function, determine precisely which integer inputs n return `True`.

   (a) `And[# > 100, # < 102] &`

   (b) `Or @@ IntegerQ /@ Sqrt[Divisors[#]] &`

   (c) `OddQ[Plus @@ Divisors[#]/Length[Divisors[#]]] &`

4. The following function `f` is a function of a positive integer n and returns a list of integers.

```
f[n_] := First /@ Select[FactorInteger[n], EvenQ@Last[#] &]
```

   (a) Determine `f[40]`.

   (b) Determine `f[50]`.

   (c) Determine `f[100]`.

   (d) Determine `f[1800]`.

   (e) For a given positive integer input n, what does the output represent?

5. Write down the output associated with following inputs (without reference to Mathematica! this is a test of your familiarity with the functions).

   (a) `Sqrt[{144,169}]`

   (b) `Log[E]`

   (c) `Sin[Pi/6]`

   (d) `Binomial[5,2]`

   (e) `FactorInteger[40]`

   (f) `Quotient[14234, 1123]*1123 + Mod[14234, 1123]`

   (g) `FoldList[#1 + 2 #2 &, 5, {4, 3, 2, 1}]`

   (h) `Fold[#1 + 2 #2 &, 5, {4, 3, 2, 1}]`

   (i) `Length[Select[Range[100], EvenQ]]`

   (j) `Union @@ Outer[Plus, Range[5], Reverse[Range[5]]]`

   (k) `Plus @@ Inner[List, Range[5], 2*Range[5] - 1, List]`

6. Sketch the output from the following call to `Graphics`.

```
l = {{0, 0}};
For[i = 1, i <= 12, i++, AppendTo[l,
  Which[
    Mod[i, 4] == 0, Last[l] + {0, i},
    Mod[i, 4] == 1, Last[l] + {i, 0},
    Mod[i, 4] == 2, Last[l] - {0, i},
    Mod[i, 4] == 3, Last[l] - {i, 0}
    ]
  ]
 ]
Graphics[Line[l]]
```

7. Write down a simple input (that is, a function of less than 50 characters) that produces the following output when evaluated.

   (a) {1,8,27,64,125,216,343,512,729,1000,1331,1728,2197,2744,3375}

   (b) 1234567891011121314151617181920212223242526272829 30
       (Hint: Look for a pattern in the digits of this integer.)

   (c) {25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10}

   (d) {1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10}

   (e) {1,3,7,9,11,13,17,19,21,23,27,29,31,33,37,39,41,43,47,49}

   (f) {12,14,16,18,32,34,36,38,52,54,56,58,72,74,76,78,92,94,96,98}

8. `AbsolutePointSize[d]` is a graphics directive which specifies that points which follow are to be shown if possible as a circular regions with absolute diameter `d`. Write an input to mathematica which creates a graphic consisting of 20 points. Specifically, the i-th point is located at $(i, \cos i)$ and has point size $20/i$. The graphic is shown below.

9. Below is an attempt to implement compute 15!. Fill in the `While` loop so that the print statement prints 15!=1307674368000.

   ```
   n=15;
   i=1;
   While[                      ,                              ]
   Print[i];
   ```