Assignment 2

Name:_____

Write or type solutions on a separate paper. If written, write legibly.

1. Determine the truth value of each predicate wff and state the truth set and falsity set of the wff.

   a) Given $P(x) \Rightarrow x/2$ is odd and $Q(x) \Rightarrow 4$ divides x, for

   $$\big(\forall x\big)\big[P(x) \rightarrow Q(x)\big]$$

   on $DOI = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$.

   b) If $G(x, y) \Rightarrow x < y$, for

   $$\big(\forall x\big)\big(\exists y\big)\big[G(x, y) \vee G(y, x)\big]$$

   on $DOI = D_x \times D_y = \{\{1, 3, 5\} \times \{2, 4, 6\}\}$

2. Write a program that evaluates a predicate of the form $\big(\exists x\big)P(x)$. Your program should define the following function

Function Name:	`eWff()`
Parameter(s):	*doi*: string *(*P)(int)*: bool
Return:	bool

   where *doi* is the name of the file to open for reading and *(*P)(int)* is a function pointer of a bool function that takes an int as a parameter. The function `eWff()` should read in and evaluate each element of the file with *(*P)(int)*; and then, it should return the truth value of $\big(\exists x\big)P(x)$ with the file as the domain of interpretation. Along with the above function, your program should define

Function Name:	`isEven()`
Parameter(s):	x: int
Return:	bool

Function Name:	`isMult3()`
Parameter(s):	x: int
Return:	bool

where `isEven()` returns true only if the absolute value of its parameter is even, and `isMult3()` returns true only if the absolute value of its parameter is a multiple of 3. Use the following main function

```
int main()
{
 string files[2] = {"data1.txt","data2.txt"};
 cout << boolalpha;

 for(int i = 0;i < 2;i += 1)
 {
  cout << "Predicate wff evaluates to " << eWff(files[i],isEven) << " for DOI " << (i +1)
  << " and predicate isEven()\n";
  cout << "Predicate wff evaluates to " << eWff(files[i],isMult3) << " for DOI " << (i + 1)
  << " and predicate isMult3()\n\n";
 }
 return 0;
}
```

3. Translate the English statements into a predicate wff. Indicate what each predicate stands for.

   a) Everyone laughs, but no one knows why.

   b) Rational numbers are fractions. Some fractions are integers. Therefore, some rational numbers are integers.

4. Prove that the following predicate arguments are valid.

   a) $\big(\exists x\big)[A(x) \wedge B(x)] \vee \big(\forall x\big)A(x) \wedge \big(\forall x\big)\big[A(x) \rightarrow B(x)\big] \rightarrow \big(\exists x\big)B(x)$

   b) $\big(\forall x\big)\big[\big(\exists y\big)P(x,y)\big]' \rightarrow \big(\forall y\big)\big(\forall x\big)P(x,y)'$

5. Write a program that evaluates a predicate of the form $\big(\forall x\big)P(x)$. Your program should define the following function

Function Name:	`uWff()`
Parameter(s):	*doi*: string *(*P)(int)*: bool
Return:	bool

where *doi* is the name of the file to open for reading and *(*P)(int)* is a function pointer of a bool function that takes an int as a parameter. The function `uWff()` should read in and evaluate each element of the file with *(*P)(int)*; and then, it should return the truth value of $(\forall x)P(x)$ with the file as the domain of interpretation. Along with the above function, your program should define

Function Name:	`isOdd()`
Parameter(s):	x: int
Return:	bool

Function Name:	`isMult5()`
Parameter(s):	x: int
Return:	bool

where `isOdd()` returns true only if the absolute value of its parameter is odd, and `isMult5()` returns true only if the absolute value of its parameter is a multiple of 5. Use the following main function

```
int main()
{
 string files[2] = {"data1.txt","data2.txt"};
 cout << boolalpha;

 for(int i = 0;i < 2;i += 1)
 {
  cout << "Predicate wff evaluates to " << eWff(files[i],isOdd) << " for DOI " << (i +1)
  << " and predicate isOdd()\n";
  cout << "Predicate wff evaluates to " << eWff(files[i],isMult5) << " for DOI " << (i + 1)
  << " and predicate isMult5()\n\n";
 }
 return 0;
}
```

Extra Credit Translate the following verbal argument into predicate form and prove that the argument is valid. You must provide the definition of each predicate.

> Every computer science student works harder than somebody, and everyone who works harder than any other person gets less sleep than than person. Maria is a computer science student. Therefore, Maria gets less sleep than someone else.