COSC2436: Recursion

Recursion

What is the output of fun(4)?

```
void fun(int x){
   if(x >= 1){
      fun(--x);
      --x;
      fun(x-2);
      cout << x << endl;
   }
}
```

Recursion

What is the output of fun(4)?

```
void fun(int x){
   if(x >= 1){
      fun(--x);
      --x;
      fun(x-2);
      cout << x << endl;
   }
}
```

OUTPUT:

-1
0
1
2

Recursion

For the function below, what values of *n* will cause an infinite loop?

```
int fun(int n){
   if(n % 2 == 1)
      return n;
   return fun(n/2);
}
```

Recursion

For the function below, what values of *n* will cause an infinite loop?

```
int fun(int n){
   if(n % 2 == 1)
      return n;
   return fun(n/2);
}
```

ANSWER: When n <= 0

Recursion

Write a recursive function to find the sum of the first n natural numbers. Example: Given 4, the function would return 10 since 1+2+3+4 = 10.

```
int getSum(int n){



}
```

Recursion

Write a recursive function to find the sum of the first n natural numbers. Example: Given 4, the function would return 10 since 1+2+3+4 = 10.

```
int getSum(int n){
   if(n == 1)
      return 1;
   return n + getSum(n-1);
}
```

Recursion

Write the function *factorial* which returns the factorial for a given number *n*.

```
int factorial(int n){



}
```

Recursion

Write the function *factorial* which returns the factorial for a given number *n*.

```
int factorial(int n){
   if(n == 0 || n == 1)
      return 1;
   return n * factorial(n-1);
}
```

Recursion

Write the function *fibonacci* which returns the *n*th fibonacci of a given number *n*.

```
int fibonacci(int n){



}
```

Recursion

Write the function *fibonacci* which returns the *n*th fibonacci of a given number *n*.

```
int fibonacci(int n){
   if(n <= 1)
      return n;
   return fibonacci(n-1) + fibonacci(n-2);
}
```

Recursion

Given a positive integer, write the function *sumOfDigits* which uses recursion to find the sum of the digits of a given integer *x*.

```
int sumOfDigits(int x){




}
```

Recursion

Given a positive integer, write the function *sumOfDigits* which uses recursion to find the sum of the digits of a given integer *x*.

```
int sumOfDigits(int x){
   if(n < 10)
      return n;
   return n % 10 + sumOfDigits(n/10);
}
```

Recursion

You are climbing a staircase that takes *n* steps to reach the top. You can either climb 1 or 2 steps at a time. Write the function *climbStairs* which returns the number of distinct ways in which you can climb to the top of a staircase with *n* stairs.

```
int climbStairs(int n){



}
```

Recursion

You are climbing a staircase that takes *n* steps to reach the top. You can either climb 1 or 2 steps at a time. Write the function *climbStairs* which returns the number of distinct ways in which you can climb to the top of a staircase with *n* stairs.

```
int climbStairs(int n){
   if(n <= 0)
      return 1;
   return climbStairs(n-1) + climbStairs(n-2);
}
```

Recursion

Write the function *printBackwards* which prints a given string *str* backwards.

```
void printBackwards(string str, int size){




}
```

Recursion

Write the function *printBackwards* which prints a given string *str* backwards.

```
void printBackwards(string str, int size){
   if(size == 0)
      return;
   cout << str[size-1];
   printBackwards(str, size-1);
}
```

Recursion

Write a recursive function to find the minimum element of an array.

```
int getMin(int arr[ ], int size){




}
```

Recursion

Write a recursive function to find the minimum element of an array.

```
int getMin(int arr[ ], int size){
   if(size == 1)
      return arr[0];
   else if(arr[size-1] < getMin(arr, size-1))
      return arr[size-1];
   return getMin(arr, size-1);
}
```