Krystal -> Week 3 : Recursion -> Ques 1, 6, 11, 18a, 18b, 21a, 21b (pg 1)

1. Demonstrate this function is recursive by listing the criteria of a recursive solution and stating how this function meets each criterion.

1.Base Case
// it checks for a case in which the function will not return via recursion
ie (if x[n-1] == desiredValue) and
if (n <= 0), return 0;
count = 1; //ie it finds the value without another call to function
2. Calls itself-> Recursive case

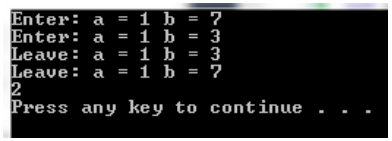
6. Describe the problem with the following recursive function:

```
void printNum(int n)
{ cout << n << endl;
  print Num(n - 1);
  }// end printNum;
```

The function doesn't have a base case.
What if n is negative? What if n is infinity (theoretically)?

11. What output does the following program produce?

Enter: a = 1 b = 7
Enter: a = 1 b = 3
Leave: a = 1 b = 3
Leave: a = 1 b = 7
2



Krystal -> Week 3 : Recursion -> Ques 1, 6, 11, 18a, 18b, 21a, 21b (pg 2)

18a. Write an iteractive function power1 to compute x^n
for some n > 0
x^ 0 = 1
x^n = x * x^(n-1) if n > 0

if (n == 0)
{ cout << 1 << endl;}
else
{ for (int i = 0; i < n; i++){
product = product * (pow(x, i));}
cout << product;
}

18b. Write a recursive function power2
to compute x^n for some n > 0
x ^0 = 1
x^n = x * x^(n-1) if n > 0

```cpp
#include <math.h>

using namespace std;


int main()
{
    int n;
    int x;
    int product = 1;
    cout << "Please enter a value for x: ";
    cin >> x;
    cout << "Please enter a value for n: ";
    cin >> n;

    if (n == 0){
        cout << " Total is 1 " << endl;
    }

    else{
        for (int i = 0; i < n; i++){

            product = product * (pow(x, i));
        }

        cout << "Total is: " << product;



    }


    // x = 2, n = 2 = total should be => 2^0 * 2^1 => 2
    // x = 3, n = 3 => total should be => 3^0 * 3^1 * 3^2 = 1 * 3 * 9 = 27

    system("PAUSE");
    return 0;
}
```

```cpp
#include <string>
#include <iostream>
#include <vector>
#include <math.h>

using namespace std;

int product(int x, int n);

int main()
{
  int a, b;

  cout << "Please enter a value for a: ";
  cin >> a;
  cout << "Please enter a value for b: ";
  cin >> b;

  cout << "Total is: " << product(a, b) << endl;
  return 0;
}

  int product(int x, int n)
  {
    if (n == 0)
      return 1;
    else if (n == 1)
      return x;
    else
      return x * product(x, n-1);

  }


  // x = 2, n = 2 = total should be => 2^0 * 2^1 => 2
  // x = 3, n = 3 => total should be => 3^0 * 3^1 * 3^2 = 1 * 3 * 9 = 27

```

Krystal -> Week 3 : Recursion -> Ques 1, 6, 11, 18a, 18b, 21a, 21b (pg 3)

kSmall (iterative)

```cpp
#include <string>
#include <iostream>
#include <vector>
#include <math.h>
#include <string>

using namespace std;


int main()
{

 int arr[6] = {4, 2, 5, 6, 10, 6};
 int size = (sizeof(arr)/ sizeof(arr[0]));
 cout << size << endl;
 int temp = arr[0];

 for (int i = 0; i < size; i++){

   if (arr[i] < temp){
     temp = arr[i];
   }
 }
 cout << "Smallest: " << temp << endl;

  return 0;
}
```

Factorial (iterative)

```cpp
#include <string>
#include <iostream>
#include <vector>
#include <math.h>

using namespace std;


int main()
{
  int x;
  int fact = 1;

  cout << "What is the factorial of the number you'd like to calculate? ";
  cin >> x;

  if (x <= 0){
    cout << "Please choose a positive number. " << endl;
  }
  else{
    for (int i=1; i <= x; i++){
      fact = fact * i;
    }
    cout << x << "! is " << fact << endl;
  }



  return 0;
}
```

21a. Write iterative versions of the following recursive functions:
(i). fact,
(ii) writeBackward,
(iii) binarySearch
(iv) kSmall

Krystal -> Week 3 : Recursion -> Ques 1, 6, 11, 18a, 18b, 21a, 21b (pg 4)

writeBackward (iterative)

```cpp
#include <string>
#include <iostream>
#include <vector>
#include <math.h>
#include <string>

using namespace std;


int main()
{

 string word;
 cout << "Please enter a word " << endl;
 cin >> word;
 cout << "Your word is " << word << endl;
  for (int i = word.size(); i >=0; i--){
    cout << word[i]; // reverse
  }
  return 0;
}
```

21a. Write iterative versions of the following recursive functions:
(i). fact,
(ii) writeBackward,
(iii) binarySearch
(iv) kSmall

binarySearch (iterative)

share save run

```cpp

using namespace std;


int main()
{
  // make a sorted vector!
  vector <int> vec1 = {2, 4, 6, 8, 10, 12, 14};
  int size = vec1.size(); // 7
  int num;
  bool found;
  cout << "Please enter the number you wish to find " << endl;
  cin >> num;
  cout << "You chose " << num << endl;

  int low = 0;
  int high = size - 1;
  int mid;

  while (low <= high)
  {
    mid = (low + high) / 2;
    if (num == vec1[mid]){
      cout << "Item found at position " << (mid + 1);
      exit(0);
    }
    else if (num > vec1[mid]){
      low = mid + 1;
    }
    else{
      high = mid - 1;
    }
  }

  cout << "Number not found." << endl;
  return 0;
}

```

input clear

```
gcc version 4.6.3
>
Please enter the number you wish to find
 3
You chose 3
Number not found.
>
```