ASSIGNMENT #1

SUBMITTED BY :

NAME : SAI KIRAN TADURI

NET ID : sxt161730

1) Compute $\log_2 2048$ using only log base 10.

Since, $\log_A B = \log_c B / \log_c A$;

$A, B, C > 0$,

$A \neq 1$;

$$\log_2 2048 = \log_{10} 2048 / \log_{10} 2$$

$$= \log_{10} 2^{11} / \log_{10} 2$$

$$= 11 \log_{10} 2 / \log_{10} 2$$

$$\boxed{\log_2 2048 = 11}$$

2) Express the following summation in closed form.

$3 + 5 + 7 + 9 + \dots + 2k + 1$

We can write the above summation as,

$$2(1 + 2 + 3 + \dots + k) + (1 + 1 + 1 + \dots + 1)$$

$$= 2(k(k+1)/2) + k$$

$$= k^2 + k + k = k(k+2)$$

3) Proof by counterexample

Prove that the following statement is false,

$n^3 > 2^n$ for any $n >= 1$.

Consider $n = 1$,

$\Rightarrow (1)^3 \overset{?}{} 2^{(1)}$

$\Rightarrow 1 < 2$.

Therefore, the statement is false,

Now Consider, $n = 10$

$\Rightarrow (10)^3 \ ? \ 2^{10}$

$\Rightarrow 1000 < 1024$

Therefore, the statement is false.

The given statement $n^3 > 2^n$ is false for $n = 1$ and $n \geq 10$.

4) Proof by Contradiction.

Prove that the following statement is true:

the square of an even number is also even.

Assume the above statement is false.

That is, given even x, then x^2 is odd.

If x^2 is odd, then $x^2 = 2c + 1$

But $x = 2a$, means $(2a)^2 = 2c+1$

$\Rightarrow 4a^2 = 2c+1$, but this says

$\Rightarrow 2(2a^2) = 2c+1$

an even number equals an odd number, which is impossible.

Therefore, the square of an even number is also even.

5) a) Prove by Induction:

$$\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

Base case: $n=1$, the sum is $(1)^3 = 1$ and

$(1)^2(1+1)^2/4 = 4/4 = 1$

So it is true for $n=1$

Inductive step:

Assume true for k

$$\sum_{i=1}^{k} i^3 = k^2(k+1)^2/4$$

Show true for $k+1$:

$$\sum_{i=1}^{k+1} i^3 = \sum_{i=1}^{k} i^3 + (k+1)^3$$

$$= \frac{k^2(k+1)^2}{4} + (k+1)^3$$

$$= (k+1)^2 \cdot \left[\frac{k^2}{4} + k + 1\right]$$

$$= (k+1)^2 \cdot \left[\frac{k^2 + 4k + 4}{4}\right]$$

$$= \frac{(k+1)^2 \cdot \left(\left[k + 1 + 1\right]\right)^2}{4}$$

$$= \frac{(k+1)^2 \cdot ((k+1)+1)^2}{4}$$

Conclusion: by induction the statement holds true for all $n >= 1$.

(b) Prove by induction:

$n^2 - n$ is even for any $n >= 1$.

Base case: $n = 1$, $n^2 - n = 2k$. where $k \geq 0$

$\Rightarrow (1)^2 - (1) = 2k.$

$\Rightarrow 2k = 0, \Rightarrow 0 = 0$

So it is true for $n = 1$ and $k = 0$

Inductive Step:

Assume true for m.

$$m^2 - m = 2k. \longrightarrow (1)$$

Show true for m+1.

$$(m+1)^2 - (m+1)$$

$$= m^2 + 2m + 1 - m - 1$$

$$= m^2 + m$$

$$= 2k + m + m \quad (\because m^2 = 2k + m \text{ from } (1))$$

$$= 2(k+m). \quad (\because \text{Anything multiplied by 2 is even})$$

Conclusion: by induction the statement holds true for all $n >= 1$.

6) Recursion.

a) Write a recursive function that when passed a value n displays

$n(n-1)(n-2)(n-3)\ldots\ldots 0 \ldots\ldots (n-3)(n-2)(n-1)n$

```
import java.util.Scanner;

public class NumberDisplay
{
    private static void print(int n)
    {
        if (n==0)                      // base case
        {   System.out.print(n);
            return;
        }
        System.out.print(n);
        print(n-1);      // recursive call
        System.out.print(n);
    }

    public static void main (String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter any number");
```

```java
        int number = scanner.nextInt();
        print (number);
    }
}
```

(6) b) Write a recursive function that receives an array of integers and a position as parameters and returns the count of odd numbers in the allay.

```java
import java.util.Scanner;
public class OddCountRecursive
{
    static int count= 0;
    private static void oddCount (int[] arr, int pos)
    {
        if (pos >= arr.length)    // Base Case
            return;
        if (arr[pos] % 2 != 0)
            count++;
        oddCount (arr, pos+1);  // Recursive Call
    }
```

```java
public static void main(String[] args)
{
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter array size");
    int size = scanner.nextInt();
    int[] arr = new int[size];
    System.out.println("Enter Numbers");
    For (int i=0; i<arr.length; i++)
        arr[i] = scanner.nextInt();
    oddCount(arr, 0);
    System.out.println("Count of odd numbers: "
                        + count);
  }
}
```

(7) Suppose there exists a generic Java class named Pair with type parameter T that stores two objects with get and set methods for each.

The statements necessary to create an object of type Pair with String as its type parameter is,

```
Pair<String> p= new Pair<>();
```

The set methods to set the two strings. are,

```
p. setObj1 (" Object1");
p. setObj2 (" Object2");
```

The get methods to retrieve them for printing.

```
String temp1 = p.getObj1();
String temp2 = p. getObj2();
System.out.println ( temp1 + " " + temp2);
```