

1_assignment1

Assignment #1

Due Date: Saturday, September 5 at 11:59pm

Submit: eLearning

Late Policy: -10 points per hour late

Instructions: This is an individual assignment. Answers should be your own work.

Chapter 1

10 points

1) Suppose your calculator only did base 10 logarithms.

Write an expression to compute log base 2 of 2048 using only log base 10.

solution:

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

10 points

2) Express the following summation in closed form (an expression that can be directly computed from k).

(Refer to slide 11)

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

solution:

$$2(1+2+3+4+\dots+k \text{ times}) + (1+1+1+\dots+k \text{ times})$$

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

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

10 points

3) Proof by counterexample

Prove that the following statement is false: $n^3 > 2^n$ for any $n \geq 1$

solution:

consider the case where $n=1$

$$1^3 = 1$$

and $2^1 = 2$

hence at case $n=1$ it is not satisfying

This $n^3 > 2^n$ for any $n \geq 1$ is false.

10 points

4) Proof by contradiction

Prove that the following statement is true: the square of an even number is also even

solution:

let us assume it to be false : The square of an even number is odd

If the even number is $2a$ then its square is odd that is $(2a)^2 = (2x) + 1$

but even number square is $(2a)^2 = 2^2 * a^2 = 4 (a^2) = 2 (2 a^2)$

so $2 (2 a^2) = 2x + 1$ but this says an even number is equal an odd number, which is impossible.

Therefore the square of an even number is also even.

30 points

5) Induction proofs.

a. Prove by induction:

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

Note: "sum" is the summation symbol, and ^ is for an exponent

solution:

Base case : $n=1$

so LHS equals

1_assignment1

$$1^3 = 1$$

and RHS equals

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

so it is true for $n = 1$.

Inductive step :

Assume it to be true for $n=k$, then

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

to show it to be true for $(k+1)$:

$$\begin{aligned} \sum_{i=1}^{k+1} (k+1)^3 &= \sum_{i=1}^k (k)^3 + (k+1)^3 \\ &= [k^2][(k+1)^2]/4 + (k+1)^3 \\ &= \{ [k^2][(k+1)^2] + 4 [(k+1)^3] \} / 4 \\ &= \{ [(k+1)^2] [k^2 + 4 [k+1]] \} / 4 \\ &= \{ [(k+1)^2] [k^2 + 4k + 4] \} / 4 \\ &= \{ [(k+1)^2] [(k+2)^2] \} / 4 \\ &= \{ [(k+1)^2] [(k+1) + 1]^2 \} / 4 = \end{aligned}$$

Conclusion: by induction the statement holds true for all $n \geq 1$

b. Prove by induction:

$$n^2 - n \text{ is even for any } n \geq 1$$

solution:

Base case : $n=2$

so LHS equals

1_assignment1

$$2^2 - 2 = 2$$

since the resultant 2 is even

it is true for $n = 1$.

Inductive Step :

Assume it to be true for $n=k$, then

$k^2 - k$ is even that is $k^2 - k = 2x$

to show it to be true for $(k+1)$:

$$\begin{aligned}(k+1)^2 - (k+1) &= (k^2 + 2k + 1) - (k + 1) \\ &= k^2 + k \\ &= (k^2 - k) + 2k = 2x + 2k = 2(x+k) \\ &= \text{even number}\end{aligned}$$

Conclusion: by induction the statement holds true for all $n \geq 1$

20 points

6) Recursion.

Note: You can use Java or pseudocode for these. If pseudocode then the logic must be complete and easy to understand.

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

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

for example, if passed 5 displays

5 4 3 2 1 0 1 2 3 4 5

Solution:

```
import java.util.Scanner;

public class numbers_pattern
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
```

```

                                1_assignment1
System.out.println("Enter any number:");
int x = scanner.nextInt();
System.out.println("The number is "+ x);
Display(x);
}

public static void Display(int x)
{
if(x==0)
    System.out.print("0 ");
else
    {
        System.out.print(x+" ");
        Display(x-1);
        System.out.print(x+" ");
    }
}
}

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 array. Let each recursive call consider the
    next integer in the array.

```

Solution:

```

import java.util.Scanner;

public class odd_number_array
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        int[] a= new int[30];
        System.out.println("Enter number of items in the array:");
        int x = scanner.nextInt();
        for(int i=0;i<x;i++)
        {
            a[i]=scanner.nextInt();
        }

        //int[] a={1,2,3,4,5,6};

        int o=odd_count(a,0);
        System.out.println("The number of odd numbers in the array is "+o);
    }
}

```

1_assignment1

```
}

public static int odd_count(int a[], int i)
{
    if (i == a.length-1)
        return 0;
    else
    {
        if((a[i])%2 != 0 )
            return 1 + odd_count(a,i+1);
        else
            return 0 + odd_count(a,i+1);
    }
}
}
```

10 points

- 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. Write the statements necessary to create an object of type Pair with String as its type parameter, and use the set methods to set the two strings, then the get methods to retrieve them for printing. Note that you do not need to write the Pair class itself.

Solution:

```
import java.util.*;

public class Pair<T>
{
    public void seta(T x)
    {
        a=x;
    }

    public T geta()
    {
        return a;
    }
}
```

1_assignment1

```
public void setb(T x)
{
    b=x;
}

public T getb()
{
    return b;
}

private T a;
private T b;

public static void main (String args[])
{
    Pair<String> p = new Pair<String>();
    System.out.println("Enter the value of p.a:");
    Scanner scanner = new Scanner(System.in);
    String sa=scanner.nextLine();
    p.seta(sa);

    System.out.println("Enter the value of p.b:");
    String sb=scanner.nextLine();
    p.setb(sb);

    System.out.println("the values of p.a and p.b are : " + p.geta() +" and "+
p.getb());
}
}
```

Submit to eLearning:

assignment1.xxx where xxx is txt, doc, docx, pdf, or jpg

You may scan handwritten work if you wish.