Task 3: Write a letter sequence, that is next on the series. Try to determine the general form of the series: a) bce , bbcde , bbbcdde , _____

The general form: b) bcace , bcacacace , bcacacacacace , _____

The general form: c) _____ , abcccdd , abbccccdd , abbbccccccdd , … The general form:

Ans:-

a)

b^n.c^1.d^(n-1).e^1

=b^n.c.d^(n-1).e

b) b^1.ca(2n-1).ce

c) a.b^(n-1).c^(n+1).dd

Task 4: Your job is to sum up 50 numbers and at the end report the sum and the count of numbers that were positive numbers (>0). Describe how you would solve the problem or write in a pseudo-code.

int i, num, s=0

i=0

do

input num

s=s+num

if(num>0)

{

i=i++

}

till the value of i<=50

print s as the sum of numbers.

print i as the number of positive number.

Task 5: Your job is to deduce which of the given numbers is the smallest. How many comparisons you have to make if you have: a) 3 numbers b) 6 numbers c) n numbers _____ (n>=1)

a)2

b)5

c)n-1

part 2:If you have 4987 participants in a knock out ping pong championship . How many matches would it take to determine the winner. Number of games played by winner by the time winner is determined.

Ans:-

4986 matches would take to determine winner

13 match palyed by winner

Part 3: Computer Science Foundation

Task 1: Define Binary searching.

Ans:- A binary search is a quick and efficient method of finding a specific target value from a set of ordered items. By starting in the middle of the sorted list, it can effectively cut the search space in half by determining whether to ascend or descend the list based on the median value compared to the target value.

Binary search is a fast search algorithm with run-time complexity of O(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form.

Task 2: Write a algorithm or pseudo-code to find common elements from two different array.

Ans:-

#Algorithm

declare array and array lenght as integer arr1[],arr2[],len1,len2

for i=0 till i<len1

for j=0 till j<len2

if arr[i]==arr2[2]

print as common

increment j and i

#code

```
Public Class ArrayComonElements {
   Public static void (string[] args) {
            int[] array1=(1,2,3,4,5);
             int[] array2=(3,4,5);
//Iterating Array1
  for(int i=0;i<array1.length;i++){
//Iterating array2
   for(int j=0;j<array2.length;j++)
{
        if(array1[i]==array2[j])
            {
                system.out.println(ayrray1[i]);
               }
}
}
```

Task 3: Calculate the space time complexity of your solution to prior question.

space time complexity is O(mn). Where m and n are the number of elements in arr1[] and arr2[].