**NAME:KISHORE.S**

**DATE:26-03-2023**

**ASSIGNMENT-01**

**1: find the max element in an array**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **23** | **65** | **12** | **87** | **12** | **343** | **098** | **12** | **76** | **312** |

**Pre-condition** :

Define two variables, a variable i that act as a loop counter.

a variable max to store the maximum element.

Before start loop find max element from a[0] to a[10-1], so at the start initialise max=0 and i=1.

**Post-condition:**

After the loop termination the value of max element stores the maximum of all elements form a[0] to a[10-1].

**Loop variant :**

the loop will terminate after finding the max element from a[0] to a[10-1].

loop invariant

Let's assume invariant is true after (i-1)th iteration

i.e., max stores the maximum value of all elements from a[0] to a10-1]

we need to design instruction so that the variant must be true after ith iterations of the loop

i.e. max must be equal to the max of all integers from a[0] to a[i]

**Pseudocode**

int a[10] = {23,65,12,87,12,343,098,12,76,312};

{

int max = a[0];

for(i = 1;i < n ; i = i++)

{

if(max < a[i])

max = a[i];

}

return max;

}

**2. Convert roman numbers into integers**

**Pre-condition:**

Defining variables:

A variable ‘i’ that acts as a loop counter, and a variable ‘total’ store the sum of integer values of char r.

A variable ‘s’ in the arguments to pass the input and a variables s1 and s2 to store the integer value at i and i+1th position.

Before starting the loop from i=0 to i<s.length(), we need to initialize total=0 and starts the loop.

This precondition is true, enters the first iteration of loop.

**Post-Condition**

After the loop termination, the final sum value of total stores the integer for the roman letters from i=0 to i<s.length( ).

**Loop variant**

The loop must terminate after sum of all integers values of Roman Letter and returns the total.

In other words, loop should not terminate until the integer values of all roman letters sum.

i.e. i<=length()-1 or i<s.length( )

**Loop invariant**

Let us assume invariant is true after (i-1)th iteration.

i.e. total stores the integers value of all charAt(i) from i to i-1

We need to design instruction so that the invariant must be after ith iteration of the loop.

i.e. total must be equal to total integer values of all charAt(i) from i=0 to i=n-1

**pseudocode**

int convertromantojnt(String s)

{

int total = 0;

for (int i = 0; i < s. length(); i++) {

int s1 = value(s.charAt(i));

if (i+1 < s. length()) {

int s2 = value(s. charAt(i + 1));

if (s1 >= s2) {

total = total + s1;

} else {

total = total - s1;

}

} else {

total = total + s1;

}

}

return total;

**3 : Move zeros to end of an array**

**Pre-condition**

Define four variables, len, i, j, temp

A variable 'i' that acts as a loop counter which is to perform operation for n times

variable 'j' acts as loop counter for finding the zero elements, that executes until matches found

variable "len" acts as condition which loop executes until n becomes false

variable "temp" is used for shifting or interchanging the values

**Post-Condition**

After the loop termination, the zero values must be shifted to end of an array from a[n-1] to a[j]

**Loop variant**

The loop must terminate after shifting all zeros to the end from a[len-1] to a[j]

In other words, loop should not terminate until the shifting process ends

i.e. i<=len-1 or i<n and j<len

**Loop invariant**

Let us assume loop invariant is true after (i-1)th and jth iteration.

i.e. a[j]==0 for the jth iteration

we need design the instruction so that the invariant must be true for jth iteration and ith iteration of loop

i.e. a[n-1]=a[j] inter-changes the value for each iteration len--

**Pseudocode**

int len = a.length;

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

for (int j = i; j < len; j++) {

if (a[j] == 0) {

int temp = a[j];

a[j] = a[len - 1];

a[len - 1] = temp;

}

}

len--;

}