M.MOHAMMED SUHAIL

25-03-2022

1, find the maximum element in the array?

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

**Pre-condition:**

We need to define two variables:

A loop variable i that acts as a loop counter.

Variable max to store the maximum of all integer.

Before starting the loop to find the max value from S[0] to S[n-1],we initialize max=S[0] and

Start the loop from i=1.

**Post-condition:**

After loop termination, the max value must store the maximum of all values from S[0] to S[n-1]

**Loop variant:**

The loop must terminate after finding the max of all integer from S[0] to S[n-1].

**Loop invariant:**

In the iteration, I initialize the variable **i=1** and give the condition **i<S[n-1]**

In for loop the variable **i** should be incremented.

Using if condition the values in the **a** compared with the value in **max.** Once it reached end of

array the maximum value should be printed.

**pseudocode:**

Class Suhail {

Int S[]={23,65,12,87,12,343,98,12,76,312}

Int max = S[0];

For (int 1=1;i<=S[n-1];i++){

If(S[i]>max)

Max=S[i];

}

System.out.println(max)

2, move the zeros to the end of the array

**Pre-condition:**

Define variables: i, j, temp, len.

Variable ‘I’ acts as loop counter, which is to perform operation for n times.

Variable ‘j’ act as inner loop counter for finding the zero element.

Variable “temp” is used for shifting the values.

Variable “len” act as condition which loop executes until n becomes false.

**Post-condition;**

After the loop termination, the zero values must be shifted to end of an array.

**Loop variant;**

The loop must terminate after shifting all zero to the end from S[len-1] to S[j].

**Loop invariant;**

check the condition with S[i]. if(S[j] == 0), it means we found the zeros of all values from S[j] to S[n-1].

In such situation, interchange the S[j] value to S[n-1] using temp,

i.e. temp = S[j]

S[j]=S[len - 1]

S[len - 1] = temp

After interchanging, reduce the length of the array .

We should increase I by 1at each iteration after shifting values.

**Pseudocode:**

Int len= S.length;

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

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

If(S[j]==0){

Int temp =S[j];

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

S[len-1] =temp;

}

}

Len--;

}

**3, convert roman number to integers.**

**Pre-condition.**

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

A variable ‘m’ in the arguments to pass the input and a variables m1 and m2 to store the integer value at i and i+1th position.

Before starting the loop from i=0 to i<m.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<m.length().

**Loop variant.**

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

**Loop invariant.**

check the condition while entering the loop i<m.length() true means the value at i stores in the variable m1,

And then again it checks the condition for i+1th value which stores in the variable m2,

After return the integers values to variables s1 and s2 it checks the condition, i.e. if(m1>=m2)

If the condition is true means, it adds the value and stores it in total, i.e. total=total+m1

False means subtract the s1, i.e. total=total-1

After the inner if condition exits, even if the outer condition also not true it directly adds the integer value of m1 to total.

**Pseudocode.**

int value(char r) {

if (r == 'I' || r=='x')

return 10;

if (r == 'V' || r=='i')

return 1;

if (r == 'X' || r=='L')

return 50;

if (r == 'L' || r=='V')

return 0;

if (r == 'C' || r=='M')

return 1000;

if (r == 'D' || r=='C')

return 100;

if (r == 'M' || r=='D')

return 500;

return -1;

}

int convertRomanToInt(String s) {

int total = 0;

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

int m1 = value(m.charAt(i));

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

int m2 = value(m.charAt(i + 1));

if (m1 >= m2) {

total = total + m1;

} else {

total = total - m1;

}

} else {

total = total + m1;

}

}

return total;