**BÁO CÁO THỰC HÀNH**

|  |  |  |
| --- | --- | --- |
| Họ tên | Nguyễn Việt Nhật | Lớp: KHTN2021 |
| MSSV | 21520378 | STT: 15 |
| Bài Thực Hành | LAB04 | |
| CBHD | **Trương Văn Cương** | |

# Điểm buổi thực hành

|  |  |  |
| --- | --- | --- |
| **Chuyên cần (20%)** |  |  |
| **Trình bày (20%)** |  |  |
| **Nội dung thực hành (60%)** |  |  |
| **Tổng (100%)** |  |  |

**Bài tập thực hành:**

Viết chương trình Assembly thực hiện các yêu cầu sau:

1. Nhập một mảng số tự nhiên.
2. Tính tổng các phần tử trong mảng.
3. Tìm số lớn nhất, nhỏ nhất trong mảng.
4. Đếm số lượng phần tử chẵn, số lượng phần tử lẻ.

# Pseudocode

## Nhập mảng gồm N số tự nhiên (Input)

get number of elements (N) // $s0 <- N

while (N < 0 or N > 100) // break when 0 < $s0 < 101

get number of elements

end

// read elements

set index to 0 // $t1 = 0

while (index < N) // $t1 < $s0

get value of i-th element of array // first: $t8 <- value - then: arr[index] <- $t8

while (array[index] <= 0)

get value of i-th element of array

end

end

## Tính tổng các phần tử trong mảng (sum)

set index to 1 // $t1 = 0

set sum to array[0] // $s3 = array[0]

while (index < N)

sum <- sum + array[index] // $s3 <- $s3 + array[index]

end

## Tìm phần tử nhỏ nhất, lớn nhất (min – max)

set min to array[0] // $s4 = array[0]

set max to array[0] // $s5 = array[0]

set index to 1 // $t1 = 0

while (index < N)

if (min > array[index])

min <- array[index] // minimize($s4, array[index])

if (max < array[index])

max <- array[index] // maximize($s5, array[index])

end

## Đếm số lượng số chẵn, số lẻ trong mảng (num\_even, num\_odd)

set num\_even to 0 // $s6 = 0

set num\_odd to 0 // $s7 = 0

set index to 0 // $t1 = 0

while (index < N)

if (array[index] is even)

num\_even++ // $s6 <- $s6 + 1

else

num\_odd++ // $s7 <- $s7 + 1

end

## Xuất các giá trị (1.2 – 1.4)

print sum, min, max, num\_even, num\_odd

## Nhận xét

Dễ dàng quan sát, những đoạn chương trình con (mã giả) nhằm tính giá trị sum, min, max, số số chẵn, số số lẻ (1.2 – 1.4) đều sử dụng đến vòng lặp while. Do đó, ta có thể biến đổi một chút các đoạn code để chương trình Assembly được tinh gọn hơn rất nhiều.

# Thông tin chi tiết:

## [Repo Github](https://github.com/nv259/LAB04)

## [Source code](https://github.com/nv259/LAB04/blob/2ac691f9f36db3b0f330fdce3fadff06a00674da/21520378.asm)