

BÁO CÁO THỰC HÀNH

Họ tên	Nguyễn Việt Nhật	Lớp: KHTN2021
MSSV	21520378	STT: 15
Bài Thực Hành	LAB04	
CBHD	Trương Văn Cường	

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 gồm N số tự nhiên ($0 < N < 101$).
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.

1 Pseudocode

1.1 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
```

1.2 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
```

1.3 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
end
```

1.4 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) // (array[index] and 1)
    num_even++           // $s6 <- $s6 + 1
  else
    num_odd++            // $s7 <- $s7 + 1
end
```

1.5 Xuất các giá trị (1.2 – 1.4)

```
print sum, min, max, num_even, num_odd
```

1.6 Nhận xét

Để 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.

2 Khởi tạo biến cho chương trình Assembly:

- \$t6: giá trị hằng số 1
- \$t7: giá trị hằng số 101
- \$s0: kích thước mảng
- \$t1: chỉ số (index)
- \$t3: địa chỉ của mảng
- \$s3: tổng các phần tử
- \$s4: phần tử nhỏ nhất
- \$s5: phần tử lớn nhất
- \$s6: số lượng phần tử chẵn
- \$s7: số lượng phần tử lẻ

3 Thông tin chi tiết:

3.1 [Repo Github](#)

3.2 [Source code](#)

4 Kết quả thực nghiệm

```
NHAP CAC PHAN TU CUA MANG
Nhap do dai mang: 0
NHAP CAC PHAN TU CUA MANG
Nhap do dai mang: -1
NHAP CAC PHAN TU CUA MANG
Nhap do dai mang: 101
NHAP CAC PHAN TU CUA MANG
Nhap do dai mang: 6
Nhap a[0]: 1
Nhap a[1]: 0
Nhap a[1]: 2
Nhap a[2]: -1
Nhap a[2]: 0
Nhap a[2]: 3
Nhap a[3]: 4
Nhap a[4]: 5
Nhap a[5]: 6
CAC PHAN TU CUA MANG: 1 2 3 4 5 6
1. Tong cac phan tu: 21
2. Phan tu nho nhat: 1
3. Phan tu lon nhat: 6
4. So cac phan tu chan: 3
5. So cac phan tu le: 3
```

Hình 1: Kết quả thực nghiệm