**Mini project**

Họ và tên :

Nguyễn Văn Thái – 20215135

Nguyễn Ngọc Thành -20215137

Bài 15:

**Code :**

|  |
| --- |
| .data  mess1: .asciiz "\nnhap vao so luong sinh vien: "  mess2: .asciiz "\nnhap vao ten sinh vien: "  mess3: .asciiz "\nnhap vao diem sinh vien: \0 "  numStudents: .word 0  students: .space 400  enter: .asciiz "\n"  .text  main:  # Nhập vào số lượng sinh viên  li $v0, 4  la $a0, mess1  syscall  li $v0, 5  syscall  move $t0, $v0  sw $t0, numStudents  # Nhập thông tin sinh viên và điểm  li $t1, 0 # i  la $s0, students # a[]  loop\_in:  slt $t2, $t1, $t0  beq $t2, $0, print  mul $t2, $t1, 16  add $t3, $s0, $t2  # Nhập tên sinh viên và lưu vào mảng  li $v0, 4  la $a0, mess2  syscall  li $v0, 8  move $a0, $t3  li $a1, 100  syscall  # Nhập điểm sinh viên  li $v0, 4  la $a0, mess3  syscall  li $v0, 5  syscall  sw $v0, -8($t3)  addi $t1, $t1, 1  j loop\_in  print:  # In ra danh sách sinh viên  li $t1, 0  la $s0, students  loop\_print:  slt $t2, $t1, $t0  beq $t2, $0, end\_print  li $v0, 4  la $a0, enter  syscall  mul $t2, $t1, 16  add $t3, $s0, $t2  li $v0, 4  la $a0, 0($t3)  syscall  li $v0, 1  lw $a0, -8($t3)  syscall  addi $t1, $t1, 1  j loop\_print  end\_print:  # Kết thúc chương trình  li $v0, 10  syscall |

Giải thích code :

Bài trên có thể chia làm 3 phần chính là :

1. Thực hiện nhập vào dữ liệu đầu vào là lưu lại vào mảng đã khai báo từ trước

2. Thực hiện sắp xếp sinh viên theo điểm (ở trong bài là thực hiện sắp xếp bằng thuật toán nổi bọt)

3. In ra màn hình dữ liệu sau khi sắp xếp xong

Giải thích từng thành phần :

1. Nhập dữ liệu đầu vào :

* Ở phần đầu thực hiện khai báo không gian để chứa thông tin như tên , điểm của sinh viên.(mỗi sinh viên có 8 byte để lưu giữ tên và 8 byte để lưu điểm )
* Nhập vào thông tin sinh viên bằng các hàm syscall

1. Sắp xếp sinh viên theo điểm (sử dụng thuật toán sắp xếp nổi bọt)

* Hàm for1 : kiểm tra i xem vòng lặp đã kết thúc chưa
* Hàm for2: lấy ra các giá trị như tên, điểm của sinh viên để thực hiện so sánh

🡪 Sau khi so sánh nếu không thỏa mãn điều kiện thì thực hiện đổi chỗ 2 sinh viên

🡪Nếu thỏa mãn thì thực hiện đi đến hàm skip\_swap

* Hàm skip\_swap : thực hiện quay lại vòng lặp for2 mà không đổi chỗ đồng thời tăng giá trị của j lên 1
* Hàm end\_for2 : thực hiện tăng giá trị của i sau 1 lượt so sánh theo j đồng thời quay trở lại for1

1. In ra dữ liệu sinh viên đã nhập sau khi sắp xếp xong

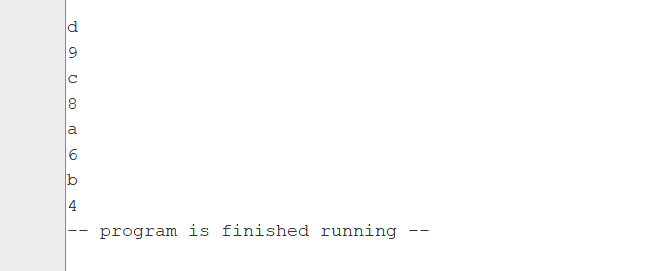
* Sau khi sắp xếp thì dữ liệu được lưu trong thanh ghi $s0 ta chỉ cần thực hiện in ra dữ liệu trong đó bằng các lệnh syscall

**Kết quả chạy :**

Nhập dữ liệu :



Kết quả sau khi sắp xếp :



**Bài 8:**

**Code:**

|  |
| --- |
| .data  data\_in:.word 1, 2, 2  step:.word 0  length:.word 3  .text  main:  la $t0, data\_in  li $t1, 0 #step  li $t2, 3 #length  li $t3, 0 #i  li $t4, 0 # moving step  li $t6, 0 #moving value  Start:  lw $t5, 0($t0)  loop:  addi $t0, $t0, 4  addi $t3, $t3, 1  beq $t3, $t2, end  lw $t6, 0($t0)  slt $t7, $t5, $t6  beq $t7, $0, next\_step  move $t5, $t6  j loop    next\_step:  sub $s0, $t5, $t6  add $t1, $s0, $t1  addi $t1, $t1, 1  addi $t5, $t5, 1  j loop    end:  li $v0, 1  move $a0, $t1  syscall  li $v0, 10  syscall |

**Giải thích :**

+ *Hàm main* : Đầu tiên thì ta cần thực hiện khai báo các giá trị cho các thanh ghi tương ứng như độ dài dãy số, địa chỉ mảng chứa dãy số đó, …

+ *Hàm start* : thực hiện lấy giá trị đầu tiên của mảng ra

+ *Hàm loop :*

* Thực hiện tăng địa chỉ đến vị trí của phần tử tiếp theo của mảng đồng thời tăng giá trị của i lên để kiểm tra kết thúc vòng lặp
* Lấy ra phần tử tiếp theo và kiểm tra so với phần tử trước đó
* Nếu như phần tử vừa lấy ra mà nhỏ hơn hoặc bằng phần tử trước nó trong mảng thì thực hiện đi đến hàm *next\_step*
* Nếu như phần tử vừa lấy ra mà lớn hơn phần tử trước đấy thì đã thỏa mãn dãy tăng. Ta thực hiện gán $t5 bằng phần tử đấy để so sánh với phần tử tiếp theo rối quay trở lại hàm loop để tiếp tục kiểm tra số tiếp theo.

+ *Hàm next\_step* : thực hiện tính toán bước cần thiết để số tiếp theo tối thiểu để số vừa lấy ra lớn hơn số trước nó 1 đơn vị bằng cách lấy số lớn hơn trừ đi số nhỏ hơn rồi cộng thêm 1 vào giá trị đó 🡪 sau đó thực hiện cộng vào bước cần nhảy để dãy tăng 🡪 sau khi thực hiện bước trên thì quay lại hàm loop để kiểm tra số tiếp theo.

+ *Hàm end* : in ra số bước nhảy tối thiểu để dãy là dãy tăng và kết thúc chương trình.

**Kết quả :**

**Ví dụ :**

Đầu vào :

****

Kết quả :

****

**Ví dụ 2 :**

Đầu vào :

****

Kết quả :

****

**Ví dụ 3 :**

Đầu vào :



Kết quả :

