

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÀI TẬP LỚN

KIẾN TRÚC MÁY TÍNH

ĐỀ 07. SẮP XẾP DÃY SỐ NGUYÊN

Viết chương trình sắp xếp dãy số nguyên có 10 phần tử nhập từ bàn phím dùng giải thuật Selection sort. Yêu cầu xuất kết quả ra màn hình khi có thay đổi thứ tự trong dãy.

GVHD : GV. Nguyễn Xuân Minh

NHÓM: 16

SVTH : Nguyễn Xuân Trục 1513804

Nguyễn Hữu Thuận 1920060

TP.HCM, 2020

MỤC LỤC

1. Thống kê các lệnh đã sử dụng trong chương trình	2
1.1. Nhóm lệnh số học và luận lý	2
1.2. Nhóm lệnh di chuyển dữ liệu	2
1.3. Nhóm lệnh hỗ trợ ra quyết định	2
1.4. Các lệnh khác	3
2. Thời gian thực thi	3
3. Kiểm tra chương trình	3
3.1. Input: -6, -10, -2, -4, -1, -12, -20, -7, -100, -9	3
3.2. Input: 6, 10, 2, 4, 1, 12, 20, 7, 100, 9	3
3.3. Input: -6, 10, 2, 4, 0, -12, 20, -7, 100, 9	3
3.4. Input: -6, -6, -6, 10, 10, 2, 3, -12, 100, 100	4

1. Thống kê các lệnh đã sử dụng trong chương trình

1.1. Nhóm lệnh số học và luận lý

Lệnh	Cú pháp đã sử dụng	Hoạt động
sll	sll \$t1, \$t2, 10	Set \$t1 to result of shifting \$t2 left by number of bits specified by immediate
sub	sub \$t1, \$t2, \$t3	Set \$t1 to (\$t2 minus \$t3)
subi	subi \$t1, \$t2, 100000	Set \$t1 to (\$t2 minus 32-bit immediate)
add	add \$t1, \$t2, \$t3	Set \$t1 to \$t2 (\$t2 plus \$t3)
addi	addi \$t1, \$t2, 100000	Set \$t1 to (\$t2 plus 32-bit immediate)

1.2. Nhóm lệnh di chuyển dữ liệu

Lệnh	Cú pháp đã sử dụng	Hoạt động
la	la \$t1, label	Set \$t1 to label's address
li	li \$t1, 4	Set \$t1 to unsigned 16-bit immediate(zero-extended)
move	move \$t1, \$t2	Set \$t1 to contents of \$t2
sw	sw \$t1, -100(\$t2)	Store contents of \$t1 into effective memory word address
lw	lw \$t1, -100(\$t2)	Set \$t1 to contents of effective memory word address

1.3. Nhóm lệnh hỗ trợ ra quyết định

Lệnh	Cú pháp đã sử dụng	Hoạt động
bge	bge \$t1, \$t2, label	Branch to statement at label if \$t1 is greater or equal to \$t2
bgt	bgt \$t1, \$t2, label	Branch to statement at label if \$t1 is greater than \$t2

1.4. Các lệnh khác

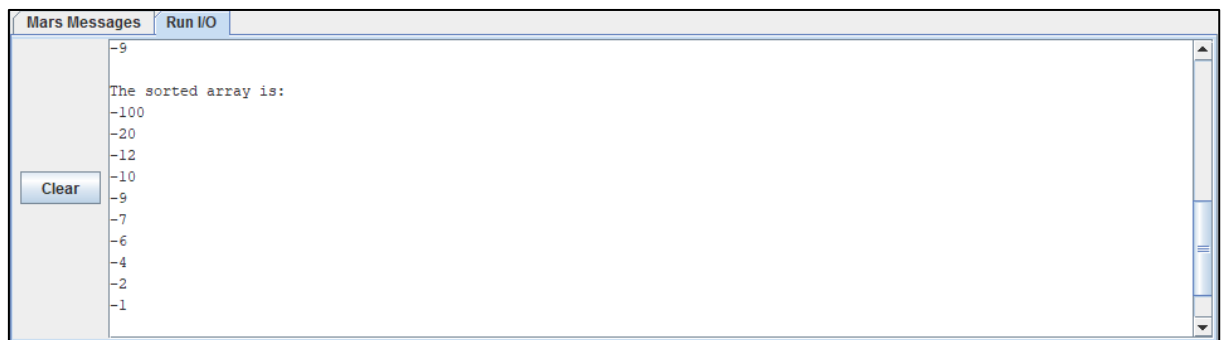
Lệnh	Cú pháp đã sử dụng	Hoạt động
j	j target	Jump to statement at target address
jal	jal target	Set \$ra to Program Counter (return address) then jump to statement at target address
jr	jr \$t1	Jump to statement whose address is in \$t1

2. Thời gian thực thi

3. Kiểm tra chương trình

3.1. Input: -6, -10, -2, -4, -1, -12, -20, -7, -100, -9

Kết quả của chương trình:



The screenshot shows the Mars Messages window with the following output:

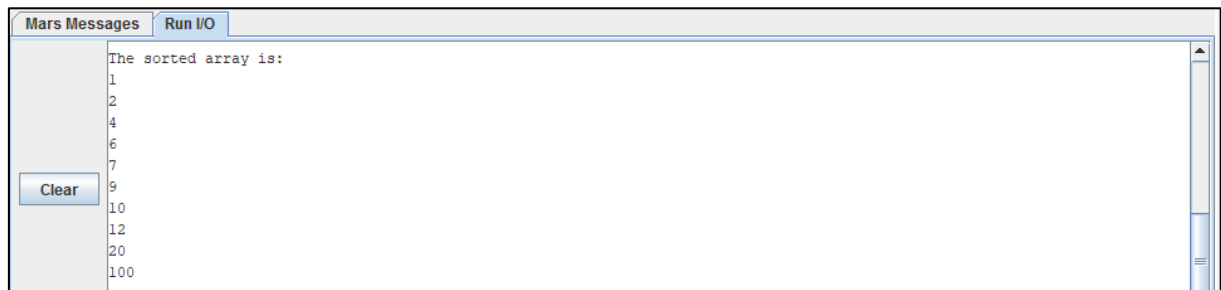
```
-9
The sorted array is:
-100
-20
-12
-10
-9
-7
-6
-4
-2
-1
```

A "Clear" button is visible on the left side of the window.

Nhận xét: Đúng

3.2. Input: 6, 10, 2, 4, 1, 12, 20, 7, 100, 9

Kết quả chương trình:



The screenshot shows the Mars Messages window with the following output:

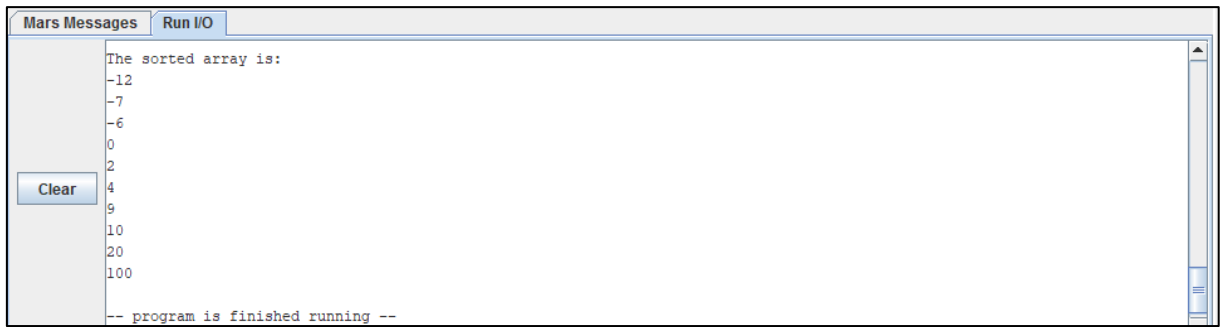
```
The sorted array is:
1
2
4
6
7
9
10
12
20
100
```

A "Clear" button is visible on the left side of the window.

Nhận xét: Đúng

3.3. Input: -6, 10, 2, 4, 0, -12, 20, -7, 100, 9

Kết quả của chương trình:



The screenshot shows a window titled "Mars Messages" with a "Run I/O" button. The output text is as follows:

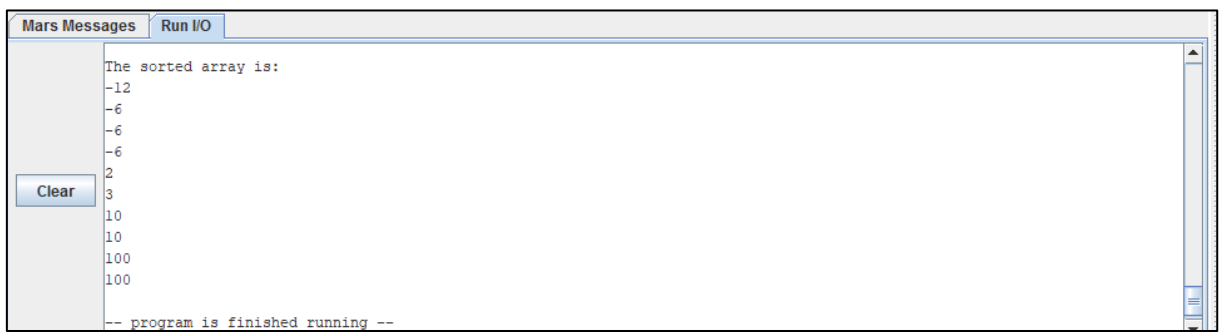
```
The sorted array is:  
-12  
-7  
-6  
0  
2  
4  
9  
10  
20  
100  
  
-- program is finished running --
```

A "Clear" button is visible on the left side of the window.

Nhận xét: Đúng

3.4. Input: -6, -6, -6, 10, 10, 2, 3, -12, 100, 100

Kết quả của chương trình:



The screenshot shows a window titled "Mars Messages" with a "Run I/O" button. The output text is as follows:

```
The sorted array is:  
-12  
-6  
-6  
-6  
2  
3  
10  
10  
100  
100  
  
-- program is finished running --
```

A "Clear" button is visible on the left side of the window.

Nhận xét: Đúng