

IT3280

THỰC HÀNH KIẾN TRÚC MÁY TÍNH

BÁO CÁO MINI-PROJECT

Sinh viên 1:

Nguyễn Trung THÀNH
20176874

Sinh viên 2:

Hoàng Thị Thu TRANG
20176891

Giảng viên hướng dẫn:

TS. Lê Xuân THÀNH

Ngày 15 tháng 6 năm 2020

Mục lục

1	Tổng Quan	2
1.1	Đề tài được phân công	2
1.2	Đề tài làm thêm	2
1.3	Công cụ sử dụng	3
1.4	Source Code	3
2	Project 7 – Sort By Height	4
2.1	Phân tích cách thức thực hiện	4
2.2	Ý nghĩa của các thanh ghi được sử dụng	4
2.3	Source code	5
2.4	Kết quả chạy chương trình	11
3	Project 8 – Students	12
3.1	Phân tích cách thức thực hiện	12
3.2	Ý nghĩa của các thanh ghi được sử dụng	12
3.3	Ý nghĩa của chương trình con	13
3.4	Source code	13
3.5	Kết quả chạy chương trình	20
4	Project 21 – Digit Degree	21
4.1	Phân tích cách thức thực hiện	21
4.2	Ý nghĩa của các thanh ghi được sử dụng	21
4.3	Ý nghĩa của chương trình con	21
4.4	Source code	21
4.5	Kết quả chạy chương trình	24
5	Project 23 – Surpassing Words	25
5.1	Phân tích cách thức thực hiện	25
5.2	Ý nghĩa của các thanh ghi được sử dụng	25
5.3	Ý nghĩa của chương trình con	26
5.4	Source code	26
5.5	Kết quả chạy chương trình	29
6	Project 24 – Cyclone Word (Challenge)	30
6.1	Phân tích cách thức thực hiện	30
6.2	Ý nghĩa của các thanh ghi được sử dụng	30
6.3	Ý nghĩa của chương trình con	30
6.4	Source code	31
6.5	Kết quả chạy chương trình	33

1 Tổng Quan

1.1 Đề tài được phân công

1. Project 7 - Sort By Height

Some people are standing in a row in a park. There are trees between them which cannot be moved. Your task is to rearrange the people by their heights in a nondescending order without moving the trees. People can be very tall!

Example: For $a = [-1, 150, 190, 170, -1, -1, 160, 180]$, the output should be $\text{sortByHeight}(a) = [-1, 150, 160, 170, -1, -1, 180, 190]$.

2. Project 8 - Students

Write a program to:

- Input the number of students in class.
- Input the name of students in class, mark.
- Sort students due to their mark.

1.2 Đề tài làm thêm

1. Project 21 - Digit Degree

Let's define the digit degree of some positive integer as the number of times we need to replace this number with the sum of its digits until we get to a one digit number. Given an integer, find its digit degree.

Example:

- For $n = 5$, the output should be $\text{digitDegree}(n) = 0$
- For $n = 100$, the output should be $\text{digitDegree}(n) = 1$
- For $n = 91$, the output should be $\text{digitDegree}(n) = 2$

2. Project 23 - Surpassing Words

Surpassing words are English words for which the gap between each adjacent pair of letters strictly increases. These gaps are computed without "wrapping around" from Z to A.

For example:

Write a function to determine whether a word passed into a function is a surpassing word. You can assume the word is made of only alphabetic characters, and are separated by whitespace. We will consider the empty string and a 1-character string to be a valid surpassing word.

- $\text{is_surpassing_word}(\text{"superb"}) \Rightarrow \text{True}$
- $\text{is_surpassing_word}(\text{"excellent"}) \Rightarrow \text{False}$

3. Project 24 - Cyclone Word (Challenge)

Cyclone Word (challenge) Cyclone words are English words that have a sequence of characters in alphabetical order when following a cyclic pattern.

For example:

Write a function to determine whether a word passed into a function is a cyclone word. You can assume that the word is made of only alphabetic characters, and is separated by whitespace.

- `is_cyclone_phrase("adjourned") => True`

- `is_cyclone_phrase("settled") => False`

1.3 Công cụ sử dụng

Mars4_5

1.4 Source Code

<https://github.com/thanhff/Computer-Architecture-Lab/tree/master/Mini-project>

2 Project 7 – Sort By Height

Đề bài: Some people are standing in a row in a park. There are trees between them which cannot be moved. Your task is to rearrange the people by their heights in a nondescending order without moving the trees. People can be very tall!

Example:

For $a = [-1, 150, 190, 170, -1, -1, 160, 180]$, the output should be $\text{sortByHeight}(a) = [-1, 150, 160, 170, -1, -1, 180, 190]$.

2.1 Phân tích cách thức thực hiện

Phân tích đề bài:

- Đầu vào: mảng array chứa các giá trị.
- Chú ý: giá trị -1 biểu thị cho tree (cây).
- Đầu ra: mảng các giá trị được sắp xếp tăng dần (các giá trị -1 giữ nguyên vị trí của nó).

Ý tưởng:

- **Bước 1:** @find_height: Lấy các giá trị là chiều cao của con người cho vào mảng height.
- **Bước 2:** @sort_height: Sắp xếp mảng height theo thứ tự tăng dần (theo thuật toán Bubble Sort).
- **Bước 3:** @change: Thay thế những giá trị đã được sắp xếp trong mảng height vào mảng array.

2.2 Ý nghĩa của các thanh ghi được sử dụng

- \$s1: lưu địa chỉ của array
- \$s2: lưu địa chỉ height
- \$s3: lưu giá trị -1 (biểu thị cho tree)
- \$s4: n - số lượng phần tử của array
- \$s5: m - số lượng phần tử của height

1. Trong @find_height:

- \$t0: biến i
- \$t2: biến j

2. Trong @sort_height:

- \$t0: biến i
- \$t1: biến j
- \$t2: m - i - 1

3. Trong @change:

- \$t0: biến i
- \$t2: biến j

2.3 Source code

```
1 #-----
2 # Project 7: sortByHeight
3 # Task: Some people are standing in a row in a park. There are trees
   between them which cannot be moved.
4 # Your task is to rearrange the people by their heights in a non-
   descending order without moving the trees.
5 # People can be very tall!
6 #-----
7
8
9 #-----
10 # @input: array (mang chua cac gia tri dau vao)
11 # @note: cac gia tri -1 bieu thi tree
12 # Mang height: chi chua cac chieu cao lay tu mang array
13 # @idea: Input array
14 #   1. @find_height: Lay cac gia tri la chieu cao cua nguoi => height
15 #   2. @sort_height: Sap xep mang height theo thu tu tang dan
16 #   3. @change: Thay the nhung gia tri da duoc sap xep trong mang height
   vao mang array
17 #-----
18
19 .data
20
21 # Input array
22 array: .space 4000 # Gioi han do dai cua mang la 1000 phan tu
23
24 # Luu cac gia tri height cua nguoi ra mot mang moi
25 height: .space 4000
26 message_input: .asciiz "Nhap gia tri dau vao cho mang (-1: bieu thi
   cho tree) !\n"
27 message_input_length: .asciiz "Ban chua nhap bat ky gia tri nao, vui
   long nhap lai!"
28 message_input_1: .asciiz "Nhap gia tri (-1: cay, cancel: ket thuc
   nhap)"
29 message_input_2: .asciiz "Nhap sai gia tri dau vao. Vui long nhap
   lai!"
30 message_input_3: .asciiz "Ket qua sau khi nhap la:\n"
31 tab: .asciiz " "
32 message_result: .asciiz "\nKet qua sau khi sap xep la:\n"
33
34
35 .text
36 main:
37     la $s1, array # Lay dia chi cua array cho vao $s1
```

```

38  la  $s2, height      # Lay dia chi height cho vao $s2
39  addi $s3, $zero, -1   # -1: bieu thi Tree trong array input
40
41  addi $s4, $zero, 0    # n: So luong phan tu cua array
42  addi $s5, $zero, 0    # m: So luong phan tu cua height
43
44
45  ### Input array
46  li  $v0, 4
47  la  $a0, message_input
48  syscall
49
50  jal input
51  nop
52
53  ### Show input array
54  li  $v0, 4
55  la  $a0, message_input_3
56  syscall
57
58  jal show_array
59  nop
60
61  ### Find height
62  jal find_height
63  nop
64
65  ### Sort height
66  jal  sort_height
67  nop
68
69  ### Change
70  jal  change
71  nop
72
73  ### Show result
74  li  $v0, 4
75  la  $a0, message_result
76  syscall
77
78  jal show_array
79  nop
80
81  ### END
82  li $v0, 10
83  syscall
84 end_main:
85
86  #-----
87  # 0. @input: Nhap cac gia tri tu ban phim
88  # Chu y: ki hieu -1 la tree
89  # Nhap -2 de ket thuc

```

```

90 #-----
91
92
93 input:
94     li $v0, 51
95     la $a0, message_input_1
96     syscall
97
98     beq $a0, -1, input_ok    # Neu ma la -1: tree, oke
99     beq $a1, 0, input_ok    # Neu la int => oke
100    beq $a1, -2, end_input    # Neu an cacel => ket thuc nhap
101
102 input_fail:
103     li $v0, 55
104     la $a0, message_input_2
105     syscall
106     j input
107
108
109 input_ok:
110     beq $a0, 0, input_fail    # Yeu cau height phai > 0
111     slt $t0, $a0, $s3    # Neu height < -1 fail (Chi -1: bieu thi cho
112                          tree oke)
113     beq $t0, 1, input_fail
114
115     sll $t1, $s4, 2    # $t1 = 4*i
116     add $t1, $t1, $s1    # Vi tri cua input[i]
117     sw $a0, 0($t1)
118     addi $s4, $s4, 1    # i = i + 1
119     j input
120
121 end_input:
122     beqz $s4, input_length_fail
123     jr $ra
124
125 input_length_fail:
126     li $v0, 55
127     la $a0, message_input_length
128     syscall
129     j input
130
131 #-----
132 # 1. @find_height: Lay cac gia tri la chieu cao cua nguoi => height
133 # @input: array (mang chua gia tri dau vao)
134 # @output: mang height chua gia tri chieu cao cua nguoi
135 #-----
136
137 find_height:
138
139     # Khoi tao cac bien i, j bang 0
140     addi $t0, $zero, 0    # i = 0

```



```

141     addi    $t2, $zero, 0    # j = 0
142
143 fh_loop:
144     sll     $t1, $t0, 2      # $t1 = 4*i
145     add     $t1, $t1, $s1    # Vi tri cua input[i]
146     lw      $s6, 0($t1)     # Lay gia tri cua input[i]
147
148     beq     $s6, $s3, fh_continue # Neu gia tri input[i] == -1 => continue
149         (bo qua tree)
150
151     sll     $t3, $t2, 2      # $t3 = 4*j
152     add     $t3, $t3, $s2    # Vi tri cua height[j]
153     sw      $s6, 0($t3)     # Luu gia tri vao vi tri height[j]
154     addi    $t2, $t2, 1      # j = j + 1
155     addi    $s5, $s5, 1      # m = m + 1 (so luong phan tu trong height tang
156         len 1)
157
158 fh_continue:
159     addi    $t0, $t0, 1      # i = i + 1
160     slt     $t4, $t0, $s4    # if i < n => True: return 1; False: return 0
161     bne     $t4, $zero, fh_loop
162
163 fh_end_loop:
164     # j after_find_height
165     jr      $ra
166
167 ##### Sap xep cac phan tu trong mang height
168 ### BubbleSort
169
170 #-----
171 # 2. @sort_height: Sap xep mang height theo thu tu tang dan
172 # @case_study: Sap xep bang phuong phap 'BubbleSort'
173 # @input: height (chua gia tri chieu cao cua nguoi) - chua duoc sap
174 #         xep
175 # @output: height (chua gia tri chieu cao cua nguoi) - da sap xep theo
176 #         thu tu tang dan
177 #-----
178
179 sort_height:
180     # Khoi tao index i cua loop_1 bang 0
181     addi    $t0, $zero, 0    # i = 0
182
183 loop_1:
184     # Khoi tao index j cua loop_2 bang 0
185     addi    $t1, $zero, 0    # j = 0
186
187     addi    $t0, $t0, 1      # i = i + 1
188     sub     $t2, $s5, $t0    # m - i - 1
189
190     # Kiem tra dieu kien: i < m - 1
191     slt     $t6, $t0, $s5

```

```

189     beq     $t6, $zero, end_loop_1
190
191 loop_2:
192     # Kiem tra dieu kien j < m - i - 1
193     slt     $t5, $t1, $t2    # j < m - i - 1: True return 1; else return 0
194     beq     $t5, $zero, end_loop_2
195
196     sll     $t3, $t1, 2      # $t3 = 4*j
197     add     $t3, $t3, $s2    # Vi tri cua A[j]
198     lw      $s6, 0($t3)     # Lay gia tri cua A[j]
199     lw      $s7, 4($t3)     # Lay gia tri cua A[j+1]
200
201 if:
202     slt     $t4, $s6, $s7    # Kiem tra A[j] < A[j+1] => True: return 1;
203     # False: return 0
204     bne     $t4, $zero, end_if
205
206     # Neu A[j] > A[j + 1] => Thuc hien Swap 2 phan tu nay
207     sw      $s7, 0($t3)
208     sw      $s6, 4($t3)
209
210 end_if:
211     addi    $t1, $t1, 1     # j = j + 1
212     j       loop_2
213
214 end_loop_2:
215     j       loop_1
216
217 end_loop_1:
218     jr      $ra
219
220 #-----
221 # 3. @change: Thay the nhung gia tri da duoc sap xep trong mang height
222 #             vao mang array
223 # @input: array (mang chua gia tri dau vao)
224 # @output: array (mang chua gia tri duoc sap xep theo chieu cao cua
225 #             nguoi)
226 # @note: Reset cac gia tri cua height = 0
227 #-----
228
229 change:
230     # Khoi tao bien i, j bang 0
231     addi    $t0, $zero, 0   # i = 0
232     addi    $t2, $zero, 0   # j = 0
233
234 i_loop:
235     sll     $t1, $t0, 2     # $t1 = 4*i
236     add     $t1, $t1, $s1    # Vi tri cua array[i]
237     lw      $s6, 0($t1)     # Lay gia tri cua array[i]
238
239     beq     $s6, $s3, i_continue # Neu gia tri array[i] == -1 => continue

```

```

238
239 sll    $t3, $t2, 2    # $t3 = 4*j
240 add    $t3, $t3, $s2  # Vi tri cua height[j]
241 lw     $s6, 0($t3)    # Lay gia tri cua height[j]
242 sw     $s6, 0($t1)    # array[i] = height[j]
243
244 addi   $s7, $zero, 0
245 sw     $s7, 0($t3)    # Reset gia tri cua height[j] = 0
246 addi   $t2, $t2, 1    # j = j + 1
247
248 i_continue:
249 addi   $t0, $t0, 1    # i = i + 1
250 slt    $t4, $t0, $s4  # if i < n: True return: 1; False return: 0
251 bne    $t4, $zero, i_loop
252
253 i_end_loop:
254 jr     $ra
255
256
257
258 #-----
259 # @show_array: Hien thi array
260 #-----
261 show_array:
262 addi   $t0, $zero, 0  # i = 0
263
264 show_array_loop:
265 sll    $t1, $t0, 2    # $t1 = 4*i
266 add    $t1, $t1, $s1  # Vi tri cua input[i]
267 lw     $s6, 0($t1)    # Lay gia tri cua input[i]
268
269 li     $v0, 4          # in dau cach de de nhin
270 la     $a0, tab
271 syscall
272 li     $v0, 1          # in gia tri
273 addi   $a0, $s6, 0
274 syscall
275
276
277 addi   $t0, $t0, 1    # i = i + 1
278 slt    $t4, $t0, $s4  # if i < n => True: return 1; False: return 0
279 bne    $t4, $zero, show_array_loop
280
281 end_show_array:
282 jr     $ra
283
284 #-----
285 # END
286 #-----

```

Project 07 - Sort By Height

2.4 Kết quả chạy chương trình

- Đầu vào:

```
Nhap gia tri dau vao cho mang (-1: bieu thi cho tree) !
Ket qua sau khi nhap la:
 1 -1 5 10 -1 2 10
Ket qua sau khi sap xep la:
 1 -1 2 5 -1 10 10
-- program is finished running --
```

3 Project 8 – Students

Đề bài: Write a program to:

- Input the number of students in class.
- Input the name of students in class, mark .
- Sort students due to their mark.

3.1 Phân tích cách thức thực hiện

Phân tích đề bài:

- Input: Số sinh viên và danh sách sinh viên.
- Output: Danh sách sinh viên được sắp xếp theo điểm (tăng dần).

Ý tưởng:

- Nhập số lượng sinh viên (số nguyên dương)
- Nhập lần lượt mã số sinh viên, tên và điểm sinh viên.
- Nhập lần lượt tên và điểm sinh viên với điều kiện: $0 \leq \text{điểm} \leq 10$
- Sắp xếp danh sách sinh viên theo thứ tự tăng dần của điểm sử dụng thuật toán Bubble sort.

Cách thức thực hiện:

- **Bước 1:** Nhập n (nếu n không phải số nguyên dương thì nhập lại)
- **Bước 2:** Nhập tên và điểm sinh viên (nếu điểm nhập vào không thỏa mãn: $0 \leq \text{điểm} \leq 10$ thì nhập lại)
- **Bước 3:** Hiển thị thông tin sinh viên vừa nhập vào.
- **Bước 4:** Sắp xếp điểm sinh viên:
 - + $j = 0$ //xử lý lần đầu tiên
 - + $i = n$ //duyệt từ cuối dãy trở về vị trí
 - Trong khi $i > j$ thực hiện:
 - Nếu $a[i] < a[i-1]$ thì hoán đổi $a[i]$ và $a[i-1]$
 - $i = i - 1$
 - + Nếu $j = j + 1$
 - Nếu $j > n - 1$ thì dừng
 - Ngược lại lặp lại bước 2.
- **Bước 5:** Hiển thị danh sách sinh viên đã được sắp xếp.

3.2 Ý nghĩa của các thanh ghi được sử dụng

- \$s1: lưu địa chỉ của mảng student
- \$t1: gán địa chỉ của mảng vào \$t1
- \$a0: số sinh viên
- \$s0, \$s6, \$s7: số sinh viên

- \$a1: chỉ ra vị trí lưu tên
- \$f1: đọc giá trị của \$s4 (=10.0) vào \$f1
- \$t0: biến đếm của sinh viên được nhập thông tin
- \$t1: chứa giá trị của mảng được gán
- \$s3: lưu địa chỉ của students để thực hiện sắp xếp
- \$t2: biến i
- \$t3: biến j
- \$t4: j - 1

3.3 Ý nghĩa của chương trình con

- __read_info_student: nạp địa chỉ của mảng.
- count: nhập số lượng sinh viên.
- name: nhập tên sinh viên với giới hạn là 47 kí tự.
- mark: nhập điểm sinh viên với điều kiện $0 \leq \text{điểm} \leq 10$.
- __bubble_sort: thực hiện sắp xếp.
- __show_student: Hiển thị danh sách sau khi sắp xếp.

3.4 Source code

```

1 #-----
2 # Project 8: students
3 #- Input the number of students in class.
4 #- Input the name of students in class, mark.
5 #- Sort students due to their mark.
6 #-----
7
8
9 #-----
10 #@idea:
11 # - Nhập số lượng sinh viên.
12 # - Nhập mã số sinh viên.
13 # - Nhập tên sinh viên.
14 # - Nhập điểm sinh viên với điều kiện  $0 \leq \text{điểm} \leq 10$ .
15 # - Sắp xếp nhân viên theo điểm dùng thuật toán bubble sort.
16 #-----
17
18 .data
19 students: .space 5200          # Array to store blocks (52) of 100
    student
20 input_count: .ascii "Nhập số lượng sinh viên"
21 error_count: .ascii "Số sinh viên phải là số tự nhiên nhỏ hơn hoặc
    bằng 100"
22 input_mssv: .ascii "Nhập mã số sinh viên"
23 error_mssv: .ascii "20170000<= MSSV < 20180000\nMSSV không trùng
    nhau."
24 trung: .ascii "MSSV này đã tồn tại"

```

```

25 input_name: .asciiiz "Nhap ten sinh vien"
26 error_in: .asciiiz "Ten ban da nhap qua dai hoac khong nhap gi! Toi
    da 47 ki tu"
27 input_mark: .asciiiz "Nhap diem sinh vien"
28 note: .asciiiz "Diem duoc nhap phai thoa man: 0 <= diem <=10"
29 error: .asciiiz "Gia tri nhap vao khong dung kieu!"
30 st_list: .asciiiz "Danh sach sinh vien"
31 mark_and_name: .asciiiz "\nMSSV\t\tDiem\tHo va Ten\n"
32 sorted_list: .asciiiz "\nDanh sach da sap xep:\n"
33 so_sinh_vien: .word
34 ten: .float 10.0
35 zero: .float 0.0
36 number: .space 4
37
38 .text
39 #Doc thong tin sinh vien
40 main:
41     la $s1, students          # Nap dia chi cua mang $s1
42     move $t1, $s1             # Gan dia chi cua mang vao $t1
43     la $s4, ten               # $s4 = 10.0
44     la $s5, zero              # $s5 = 0.0
45     li $t2, 101               # $t2 = 101
46     li $t4, 20170001          # Cho t4 = 20170001
47     li $t5, 20180000          # Cho t5 = 20180000
48     jal __count               # Ham nhap so luong SV
49     nop
50     la $s2, number            # s2 la dia chi bien number
51     lw $s0, 0($s2)            # Doc gia tri number cho vao s0
52     move $s6, $s0             # s6, s7 = SSV
53     move $s7, $s0
54     jal __input_info          # Goi ham de nhap thong tin sinh vien
55     nop
56
57     li $v0, 4                 # In string
58     la $a0, st_list           # In "Danh sach sinh vien"
59     syscall
60     jal __show_student        # Goi ham show student
61     nop
62
63     jal __bubble_sort         # Sap xep student theo diem tang dan
64     nop
65
66     li $v0, 4                 # In string
67     la $a0, sorted_list       # In "Danh sach sinh vien da sap xep"
68     syscall
69
70     jal __show_student        # In lai danh sach sau khi sap xep
71     nop
72     li $v0, 10                # Exit chương trình
73     syscall
74 #nhap so sinh vien
75 __count:

```

```

76 do_count:
77     li    $v0, 51          # Goi hop thoai nhap so luong sinh vien
78     la    $a0, input_count # $a0 tieu de: "Nhap so luong sinh vien"
79     syscall                # $a0 so sinh vien
80
81     bnez  $a1, error_input
82
83     slt   $t0, $a0, $zero   # neu gia tri nhap vao < 0 => return 1
84     bne   $t0, $zero, error_input
85
86     slt   $v0, $a0, $t2     # Neu a0 (so sinh vien) < t2 ( = 100 ) -->
87     v0 = 1
88
89     beqz  $v0, nhap_lai     # Neu v0 == 0 --> Nhap lai
90
91     la    $s7, number       # doc dia chi cua number
92     sw    $a0, 0($s7)       # ghi SSV vao number
93     jr    $ra               # Thoat khoi ham nhap so SV
94
95 nhap_lai:
96     li    $v0, 55          # Goi hop thoai Thong bao
97     la    $a0, error_count
98     li    $a1, 0           # Hop thoai Error
99     syscall
100     j     do_count         # Quay lai ham Nhap so SV
101
102 error_input:
103     la    $v0, 55
104     la    $a0, error
105     syscall
106     j     do_count
107
108 __input_info:
109     li    $t0, 0           # khoi tao $t0 = 0, $t0[U+FFFD]bien dem (i) cua sinh
110     v0 = 0                  vien vua duoc nhap thong tin
111     addi   $sp, $sp, -4     # Tao vung nho
112     sw    $ra, 0($sp)      # luu dia chi thanh ghi $ra
113
114 loop:
115     slt   $v0, $t0, $s0     # So sanh $t0 (So sinh vien nhap thong tin
116     ) < $s0 (Tong so sinh vien)-> v0 = 1
117     beqz  $v0, end_loop     # Thoat vong lap khi nhap du thong tin cho
118     cac sinh vien
119
120 # Nhap mssv
121 mssv_loop:
122     li    $v0, 51          # Goi hop thoai nhap diem (Kieu Int)
123     la    $a0, input_mssv  # Tieu de "Nhap MSSV"
124     syscall                # Tra ve gia tri vua nhap vao a0
125
126     bnez  $a1, re_input_mssv
127
128     slt   $v0, $a0, $t4     # Neu a0 (so sinh vien) < t4 ( = 20170001 )
129     --> v0 = 1
130
131     bnez  $v0, re_input_mssv # v0 != 0 --> Nhap lai

```



```

123
124     slt    $v0, $a0, $t5      # Neu a0 (so sinh vien) < t5 ( = 20180000 )
125     --> v0 = 1
126     beqz   $v0, re_input_mssv  # Neu v0 == 0 --> Nhap lai
127
128     jal    check_mssv
129     nop
130     bnez   $t9, re_mssv       # Neu t9 != 0 --> Nhap lai
131
132     j      name
133
134 re_input_mssv:
135     li     $v0, 55            # Goi hop thoai Thong bao
136     la     $a0, error_mssv    # Goi xau bao loi
137     li     $a1, 0            # Hop thoai Error
138     syscall
139     j      mssv_loop         # Quay lai ham Nhap so SV
140
141 re_mssv:
142     li     $v0, 55            # Goi hop thoai Thong bao
143     la     $a0, trung        # Goi xau : "MSSV nay da ton tai"
144     li     $a1, 0            # Hop thoai Error
145     syscall
146     j      mssv_loop         # Quay lai ham Nhap so SV
147
148 # Kiem tra MSSV co bi trung khong
149 check_mssv:
150     addi   $sp, $sp, -12      # Tao vung nho cho $t1, $t2
151     sw     $ra, 8($sp)       # Luu gia tri thanh ghi $ra
152     sw     $t2, 4($sp)       # Luu gia tri $t2 vao $sp
153     sw     $t1, 0($sp)       # Luu gia tri $t1 vao sp
154     li     $t1, 0            # Cho $t1 = 0
155     la     $t2, students     # Gan dia chi mang
156     li     $t9, 0            # Thanh ghikiem tra
157
158 loop_mssv:
159     slt    $v0, $t1, $t0      # Neu t1 ( bien dem ) < t0 ( so SV da nhap
160     ) --> v0 = 1
161     beqz   $v0, end_check     # v0 == 0 --> Thoat vong lap
162     lw     $t8, 0($t2)        # Doc mssv SV thu t1
163     beq    $t8, $a0, mssv_trung # Neu t8 == a0 ( MSSV trung nhau ) thi
164     thoat
165
166     addi   $t1, $t1, 1        # Tang bien dem
167     addi   $t2, $t2, 52       # Tang vung nho
168     j      loop_mssv
169
170 mssv_trung:
171     li     $t9, 1            # t9 = 1 MSSV bi trung
172
173 end_check:
174     lw     $t1, 0($sp)
175     lw     $t2, 4($sp)
176     lw     $ra, 8($sp)        # Tra lai gia tri cho cac thanh ghi $t1, $t2,
177     $ra

```

```

171     addi    $sp, $sp, 12
172     jr     $ra
173 #Nhap ten sinh vien
174 name:
175     sw     $a0, 0($t1)        # Luu mssv
176 do_name:
177     li     $v0, 54            # Goi hop thoai nhap ten sinh vien
178     la     $a0, input_name    # Tieu de "Nhap ten sinh vien"
179     la     $a1, 8($t1)        # Chi ra vi[0+FFFD] luu ten
180     li     $a2, 44            # Gioi han do dai ten 43 ki tu
181     syscall
182
183     bnez   $a1, re_input      # Neu a1 != 0 --> Nhap ten qua dai ( a1 =
                                # 0 la trang thai dung ) --> Nhap lai name
184     j      mark              # Chay den ham nhap diem
185
186 # Nhap lai ten
187 re_input:
188     li     $v0, 55            # Goi hop thoai thong bao String
189     la     $a0, error_in      # Goi string loi
190     li     $a1, 0             # Goi hop thoai error
191     syscall
192     j      do_name           # Quay lai Nhap name
193 #Nhap diem
194 mark:
195 do:  li     $v0, 52            # Goi hop thoai nhap diem (Kieu float)
196     la     $a0, input_mark    # Tieu de "Nhap diem sinh vien"
197     syscall
198
199     bnez   $a1, err_input
200
201     l.s     $f1, ($s4)         # Doc gia tri cua s4 (= 10.0) vao thanh ghi f1
202     c.le.s  $f0, $f1          # f0 <= f1
203     li     $a0, 1
204     movt    $a0, $zero
205     bne     $a0, $zero, condi  # Neu a0 != 0 --> incre
206
207     l.s     $f1, ($s5)
208     c.le.s  $f1, $f0          # f1 <= f0
209     li     $a0, 1
210     movt    $a0, $zero
211     beq     $a0, $zero, exit_do # Neu a0 != 0 --> incre
212 err_input:
213     la     $v0, 55
214     la     $a0, error
215     syscall
216     j      do
217 condi:
218     la     $v0, 55
219     la     $a0, note
220     syscall
221     j      do

```

```

222 #Thoat nhap diem
223 exit_do:
224     s.s    $f0, 4($t1)          # Luu diem[U+FFFD]omang
225     addi    $t0, $t0, 1          #Tang bien dem len 1
226     addi    $t1, $t1, 52         # Chuyen sang vung nho tiep theo, moi vung
                                   nho 52 bit
227     j      loop                 # Lap lai vong lap
228 end_loop:
229     lw      $ra, 0($sp)          # Lay gia tri thanh ghi $ra
230     addi    $sp, $sp, 4
231     jr      $ra
232 #####
233 # Bat dau sap xep #
234 #####
235 #-----
236 #@case_study: Sap xep bang phuong phap bubble sort
237 #@input: danh sach sinh vien - chua duoc sap xep theo diem
238 #@output: danh sach sinh vien - da duoc sap xep theo thu tu tang dan
        cua diem so
239 #-----
240 __bubble_sort:
241     addi    $sp, $sp, -4         # Tao 1 o nho
242     sw      $ra, 0($sp)         # cat gia tri thanh ghi ra
243     la      $s3, students       # Doc dia chi cua mang student vao s3
244     addi    $t2, $t0, 0         # So sinh vien la t2 la bien dem
245 loop1: # for i = n-1 to 0 ( loop 1 )
246     addi    $t2, $t2, -1        # i = i - 1 ( t2 la bien diem tu phai sai i
                                   )
247     add     $s0, $s3, $zero      # Cho dia chi cua mang s0 = s3
248     li      $t3, 0              #[U+FFFD]nj = 0 ( t3 la bien dem 2 tu trai sang j )
249     beq     $t2, 0, break_1      # Neu t2 (i) == 0 --> Exit loop 1
250 loop2: #for j = 0 to i - 1 ( loop 2 )
251     beq     $t2, $t3, loop1      # Neu j == i --> loop1
252     l.s     $f1, 4($s0)          # Load diem students[j] = f1
253     addi    $s0, $s0, 52         # Tang dia chi len 52 --> Chuyen sang
                                   student khac --> s0 = a[j+1] = f2
254     l.s     $f2, 4($s0)          # Load diem students[j+1] = f2
255     c.lt.s  $f2, $f1            # Neu f1 < f2
256     li      $a0, 1
257     movt    $a0, $zero
258     bne     $a0, $zero, incre    # --> incre
259     jal     swap                # Neu f2 < f1 --> swap
260     nop
261 incre:
262     addi    $t3, $t3, 1          # j = j + 1
263     j      loop2                # Quay ve Loop 2
264
265 break_1:
266     lw      $ra, 0($sp)          # Lay gia tri thanh ghi ra
267     addi    $sp, $sp, 4          # Tra vung nho cho stack
268     jr      $ra                # Show Students
269 #a[j-1] vs a[j]

```

```

270 swap:
271     addi    $sp, $sp, -4          # Tao 1 vung nho trong tron stack
272     sw      $ra, 0($sp)          # Cat gia tri thanh ghi $ra
273     li      $s7, 0              # s7 = 0
274     addi    $t4, $s0, -52        # t4 = students[j] --> f1 --> Vung nho dau
                                   tien
275 loopx:
276     slti    $v0, $s7, 13         # s7 < 13 --> v0 = 1
277     beqz    $v0, end_loopx       # v0 == 0 --> s7 >= 13 --> Thoat vong lap
                                   . 1 phan tu cos 52 byte moi o 4 byte --> 13 byte
278     lw      $a1, 0($t4)          # Load gia tri cua f1 ( a[j] ) vao a1 4 byte sau
279     lw      $a2, 52($t4)         # Load gia tri cua f2 ( a[j+1] ) vao a2 4 byte
                                   lien ke
280     sw      $a1, 52($t4)         # Ghi gia tri cua a1 vao a[j+1] 4 byte
281     sw      $a2, 0($t4)         # Ghi gia tri cua a2 vao a[j] 4 byte
282     addi    $t4, $t4, 4         # Tang vung nho cua students[j] len 4
283     addi    $s7, $s7, 1         # Tang bien dem len 1
284     j       loopx               # Chay lai loopx
285 end_loopx:
286     lw      $ra, 0($sp)         # Lay gia tri thanh ghi $ra da cat
287     addi    $sp, $sp, 4         # Tra lai vung nho stack
288     jr      $ra                 # Quay ve incre
289 #in ra danh sach sinh vien da duoc sap xep
290 __show_student:
291     add     $s0, $s7, $zero      #so sinh vien = 0
292     la      $t1, students        # Nap dia chi dau mang cac block luu thong
                                   tin sinh vien
293
294     la      $a0, mark_and_name    # in ra chuoi "Diem Ho va Ten"
295     syscall
296     li      $t0, 0              # Khoi tao $t0 = 0, $t0[U+FFFD]bien diem so sinh
                                   vien da duoc duyet
297 loop3:
298     slt     $v0, $t0, $s6        # So sanh $t0 (So sv da duyet)[U+FFFD]$s0 (
                                   Tong so sinh vien)
299     beqz    $v0, exit_show       # Thoat vong lap khi duyet het sinh vien
300
301     li      $v0, 1              # Hien thi console
302     lw      $a0, 0($t1)          # In mssv ra[U+FFFD]nhinh console
303     syscall
304
305     li      $v0, 11              # In char
306     li      $a0, '\t'           # In dau tab
307     syscall
308
309     li      $v0, 2
310     l.s     $f12, 4($t1)         # In diem ra man hinh console
311     syscall
312
313     li      $v0, 11              #in char
314     li      $a0, '\t'           # In dau tab
315     syscall

```

```

316
317 li    $v0, 4          #in string
318 la    $a0, 8($t1)     # In ten sinh vien ra man hinh console
319 syscall
320
321 addi   $t0, $t0, 1     # Tang bien dem so luong sv da duyet
322 addi   $t1, $t1, 52    #tang dia chi t1 len 52
323 j      loop3          # Lap lai vong lap
324 exit_show:
325 jr     $ra
326 #-----
327 # END
328 #-----

```

Project 08 - Students

3.5 Kết quả chạy chương trình

- Ban đầu:

Danh sach sinh vien	
Diem	Ho va Ten
9.0	Nguyen Kim Anh
5.5	Nguyen Quang Huy

- Sau khi sắp xếp:

Danh sach da sap xep:	
Diem	Ho va Ten
5.5	Nguyen Quang Huy
9.0	Nguyen Kim Anh

4 Project 21 – Digit Degree

Đề bài: Project 21 - Digit Degree

Let's define the digit degree of some positive integer as the number of times we need to replace this number with the sum of its digits until we get to a one digit number. Given an integer, find its digit degree.

Example:

- For $n = 5$, the output should be $\text{digitDegree}(n) = 0$
- For $n = 100$, the output should be $\text{digitDegree}(n) = 1$
- For $n = 91$, the output should be $\text{digitDegree}(n) = 2$

4.1 Phân tích cách thức thực hiện

Phân tích đề bài:

- Đầu vào: một số nguyên dương.
- Ban đầu: $\text{digit_degree} = 0$.
- Đầu ra: digit degree của số nguyên nhập vào đó.

Cách thức thực hiện:

Bước 1: Check: nếu số $< 10 \Rightarrow$ return digit_degree . Nếu $> 10 \Rightarrow \text{digit_degree} += 1$ và tiếp tục bước 2.

Bước 2: Sử dụng vòng lặp để tách các chữ số bằng cách chia cho 10 và lưu vào mảng number.

Bước 3: Count: tính tổng các chữ số quay lại Bước 1 check để kiểm tra.

4.2 Ý nghĩa của các thanh ghi được sử dụng

- \$a1: giá trị của số được nhập vào
- \$a0: vị trí của mảng number
 - Trong digit_degree
 - \$v0: đầu ra của chương trình con
 - \$s0: lưu giá trị của 10 để so sánh
 - \$s1: giá trị đầu vào của chương trình
 - \$s2: tổng của các chữ số sau khi số đó được tách

4.3 Ý nghĩa của chương trình con

- digit_degree : thực hiện tính toán số lượng digit degree của số nhập vào.

4.4 Source code

```

1 #-----
2 # Project 21 - Digit Degree
3 # @input: mot so nguyen duong
4 # digit_degree ban dau bang 0
5 # @idea:
6 # 1. Check: Neu so < 10: return digit_degree
7 # 2. Su dung vong lap de tach cac chu so bang cach chia dan cho 10 va
   luu vao mang number.
8 # 3. Count: tinh tong cac chu so, neu tong > 10: digit_degree cong
   them 1
9 # quay lai 'check' de kiem tra
10 #-----
11
12
13 .data
14 notice: .ascii "Nhap gia tri so nguyen: "
15 result: .ascii "digitDegree = "
16 error: .ascii "Gia tri nhap vao khong dung kieu!"
17 number: .word
18
19 .text
20 main:
21 # Nhap gia tri so nguyen
22 li $v0, 51
23 la $a0, notice
24 syscall
25
26 # Kiem tra input nhap vao co phai la so nguyen duong hay khong
27 addi $t0, $zero, -1
28 beq $t0, $a1, error_input
29 slt $t0, $a0, $zero # neu gia tri nhap vao < 0 => return 1
30 bne $t0, $zero, error_input
31
32 j end_error_input
33 error_input:
34 la $v0, 55
35 la $a0, error
36 syscall
37 j end_main
38
39 end_error_input:
40
41 add $a1, $a0, $zero # So nhap vao duoc luu vao $a1
42 la $a0, number # Luu vi tri cua number
43 jal digit_degree
44 nop
45
46 # Hien thi ket qua
47 add $a1, $v0, $zero
48 li $v0, 56
49 la $a0, result
50 syscall

```

```

51
52 end_main:
53     li $v0, 10
54     syscall
55
56 #-----
57 # Thuc hien tinh digit_degree
58 #-----
59 digit_degree:
60     addi $v0, $zero, 0    # Dau ra cua chuong trinh con
61     addi $s0, $zero, 10   # $s0 luu gia tri 10 de so sanh
62     add $s1, $zero, $a1   # $s1 gia tri dau vao cua chuong trinh
63
64 #-----
65 # 1. Check: Neu so < 10: return digit_degree
66 #-----
67
68 check:
69     slt $t0, $s1, $s0     # if number < 10 => return 1
70     bne $t0, $zero, end_digit_degree
71     addi $v0, $v0, 1      # Neu so do > 10 => degit_degree += 1
72
73 #-----
74 # 2. Su dung vong lap de tach cac chu so bang cach chia dan cho 10 va
75 #    luu vao mang number.
76 #-----
77     addi $t7, $zero, 0    # So luong cac chu so n
78 loop:
79     div $t1, $s1, 10
80     mfhi $t1
81
82     add $t2, $t7, $a0     # vi tri cua phan tu
83     sb $t1, 0($t2)
84     addi $t7, $t7, 1
85
86     div $s1, $s1, 10
87     beq $s1, $zero, end_loop
88
89     j loop
90
91 end_loop:
92
93 #-----
94 # 3. Count: tinh tong cac chu so, neu tong > 10: digit_degree cong
95 #    them 1
96 #    quay lai 'check' de kiem tra
97 #-----
98     addi $s2, $zero, 0    # tong cua cac chu so
99     addi $t3, $zero, 0    # i = 0
100 for:

```



```

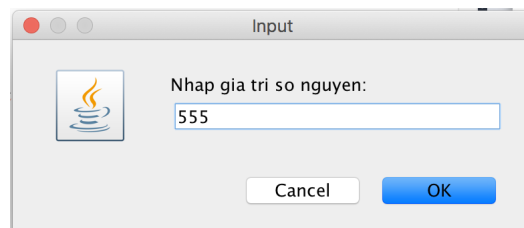
101 add $t5, $t3, $a0    # vi tri
102 lb $s3, 0($t5)
103
104 add $s2, $s2, $s3    # sum = sum + a[i]
105
106 addi $t6, $zero, 0    # xoa bo
107 sb $t6, 0($t5)
108
109 addi $t3, $t3, 1      # i = i + 1
110 slt $t4, $t3, $t7     # i >= n => False 0
111 beq $t4, $zero, end_for
112 j for
113
114 end_for:
115 sle $t0, $s0, $s2     # if 10 <= $s2 => return 1
116 beq $t0, $zero, end_digit_degree
117
118 add $s1, $zero, $s2    # tong moi sau khi cong cac chu so
119
120 j check
121
122 end_digit_degree:
123 jr $ra
124
125 #-----
126 # END
127 #-----

```

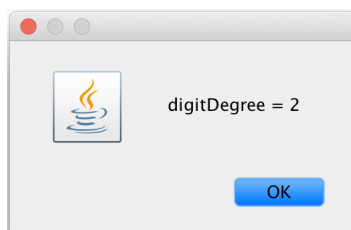
Project 21 - Digit Degree

4.5 Kết quả chạy chương trình

- Đầu vào:



- Kết quả:



5 Project 23 – Surpassing Words

Đề bài: Project 23 - Surpassing Words

Surpassing words are English words for which the gap between each adjacent pair of letters strictly increases. These gaps are computed without "wrapping around" from Z to A.

For example:

Write a function to determine whether a word passed into a function is a surpassing word. You can assume the word is made of only alphabetic characters, and are separated by whitespace. We will consider the empty string and a 1-character string to be a valid surpassing word.

- `is_surpassing_word("superb") => True`
- `is_surpassing_word("excellent") => False`

5.1 Phân tích cách thức thực hiện

Phân tích đề bài:

- Đầu vào: một từ tiếng Anh bất kỳ.
- Đầu ra: từ đó có phải surpassing words hay không.

Cách thức thực hiện:

- **Bước 1:** Sử dụng vòng lặp for từ $i = 0$ cho đến $i = n - 2$ để so sánh 2 ký tự đứng liền nhau. Mỗi lần so sánh 2 ký tự liền nhau này \Rightarrow lưu kết quả vào mảng `gap_length`. (Trong trường hợp số âm \Rightarrow Chuyển sang số dương).
- **Bước 2:** Duyệt mảng `gap_length` để kiểm tra. Nếu gặp bất kỳ trường hợp nào mà `gap_length[i] > gap_length[i+1]` \Rightarrow Trả về kết quả False. Duyệt hết mảng mà không gặp trường hợp nào như trên thì trả về True.

5.2 Ý nghĩa của các thanh ghi được sử dụng

- `$a0`: số ký tự của `input_word`
- `$a1`: lưu vị trí của `input_word`
- `$a2`: lưu vị trí của `gap_length`
- Trong vòng lặp loop thứ 1:
 - `$t0`: i
 - `$t1`: $n - 1$
- Trong vòng lặp loop thứ 2:
 - `$t0`: i
 - `$t1`: $n - 2$

5.3 Ý nghĩa của chương trình con

- check: kiểm tra từ đó có tuân thủ theo quy tắc của Surpassing Words hay không.

5.4 Source code

```
1 #-----
2 # Project 23: Surpassing Words
3 # @input: mot tu bat ky luu vao input_word
4 # @idea:
5 # Buoc 1: Su dung 1 vong lap for tu i = 0 cho den i = n - 2 de so sanh
   2 ky tu dung lien nhau
6 #   Moi lan so sanh 2 ki tu lien nhau => Luu vao mang gap_length
7 #   Trong truong hop so am => Dao dau sang so duong
8 # Buoc 2: Duyet mang gap_length kiem tra
9 #   Neu gap bat ki truong hop nao ma gap_length[i] > gap_length[i+1]
   => FALSE
10 #   Duyet het mang ma khong gap truong hop tren => TRUE
11 #-----
12
13 .data
14   # Chu y so luong phan tu cua tu
15   input_word: .asciiz "superb"
16   true: .asciiz "True"
17   false: .asciiz "False"
18   gap_length: .word
19
20 .text
21 main:
22   addi $a0, $zero, 6   # n: So cac ki tu cua input_word
23   la $a1, input_word
24   la $a2, gap_length
25
26
27   jal check
28   nop
29
30   beq $v0, $zero, false_answer
31
32 true_answer:
33   li $v0, 55
34   la $a0, true
35   syscall
36   j end_main
37
38 false_answer:
39   li $v0, 55
40   la $a0, false
41   syscall
42
```

```

43 end_main:
44     li $v0, 10
45     syscall
46
47
48 #-----
49 # @check: Kiem tra xem co phai Surpassing words hay khong
50 #-----
51
52
53 #-----
54 # Buoc 1: Su dung 1 vong lap for tu i = 0 cho den i = n - 2 de so sanh
55 #       2 ky tu dung lien nhau
56 #       Moi lan so sanh 2 ki tu lien nhau => Luu vao mang gap_length
57 #       Trong truong hop so am => Dao dau sang so duong
58 #-----
59 check:
60     addi $t0, $zero, 0 # i = 0
61     sub $t1, $a0, 1    # $t1 = n - 1
62 loop:
63
64     addi $t4, $t0, 1 # j = i + 1
65
66     add $s0, $a1, $t0
67     lb $s0, 0($s0)   # A[i]
68
69     add $s1, $a1, $t4
70     lb $s1, 0($s1)   # A[i+1]
71
72     sub $s2, $s0, $s1
73
74     slt $t5, $s2, $zero # if $s2 > 0 => 0
75     beq $t5, $zero, continue
76
77     nor $s2, $s2, $zero # Neu $s2 < 0 => Dao dau $s2 => Duong
78     add $s2, $s2, 1
79
80
81 continue:
82     add $t2, $t0, $a2
83     sb $s2, 0($t2)
84
85     addi $t0, $t0, 1 # i = i + 1
86     slt $t3, $t0, $t1 # i < n - 1 => 1; false => 0
87     beq $t3, $zero, end_loop
88     j loop
89
90 end_loop:
91
92 #-----
93 # Buoc 2: Duyet mang gap_length kiem tra

```

```

94 # Neu gap bat ki truong hop nao ma gap_length[i] > gap_length[i+1]
    => FALSE
95 # Duyet het mang ma khong gap truong hop tren => TRUE
96 #-----
97
98     addi $t0, $zero, 0    # i = 0
99     sub $t1, $a0, 2      # $t1 = n - 2
100
101 loop_2:
102     addi $t4, $t0, 1     # i + 1
103
104     add $s0, $a2, $t0
105     lb $s0, 0($s0)       # gap_length[i]
106
107     add $s1, $a2, $t4
108     lb $s1, 0($s1)       # gap_length[i+1]
109
110     slt $t2, $s0, $s1    # gap_length[i] < gap_length[i+1] => 1
111     beq $t2, $zero, return_false
112
113     addi $t0, $t0, 1     # i = i + 1
114     slt $t3, $t0, $t1    # i < n - 2 => 1; false => 0
115     beq $t3, $zero, end_loop_2
116
117     j     loop_2
118 end_loop_2:
119
120 return_true:
121     addi $v0, $zero, 1
122     jr $ra
123 return_false:
124     addi $v0, $zero, 0
125     jr $ra
126
127 #-----
128 # END
129 #-----

```

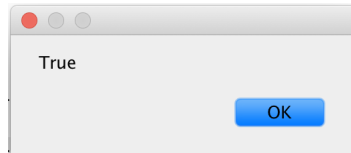
Project 23 - Surpassing Words

5.5 Kết quả chạy chương trình

- Đầu vào:

```
1 input_word: .asciiz "superb"
```

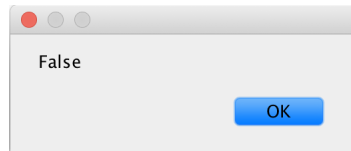
- Kết quả:



- Đầu vào:

```
1 input_word: .asciiz "excellent"
```

- Kết quả:



6 Project 24 – Cyclone Word (Challenge)

Đề bài: Project 24 - Cyclone Word (Challenge)

Cyclone Word (challenge) Cyclone words are English words that have a sequence of characters in alphabetical order when following a cyclic pattern.

For example:

Write a function to determine whether a word passed into a function is a cyclone word. You can assume that the word is made of only alphabetic characters, and is separated by whitespace.

- `is_cyclone_phrase("adjourned") => True`
- `is_cyclone_phrase("settled") => False`

6.1 Phân tích cách thức thực hiện

Phân tích đề bài:

- Đầu vào: một từ tiếng Anh bất kỳ.
- Đầu ra: từ đó có phải Cyclone Word hay không.

Cách thức thực hiện:

- **Trường hợp 1:** Nếu từ nhập vào có độ dài bằng 0 hoặc bằng 1 => TRUE.
- **Trường hợp 2:** Nếu từ nhập vào có độ dài > 1:
 - + Sử dụng vòng lặp có $i = 0$ (vị trí đầu tiên) và $j = n - 1$ (vị trí cuối cùng của từ đó).
 - + Thực hiện so sánh `c_word[i]` với `c_word[j]` và `c_word[i+1]` với `c_word[j]` và cứ tiếp tục cho đến khi $i \geq j$. Nếu `c_word[i] <= c_word[j]` và `c_word[j] <= c_word[i+1]` => TRUE.

6.2 Ý nghĩa của các thanh ghi được sử dụng

- `$a0`: lưu vị trí của `c_word`
- `$a1`: số lượng kí tự của từ đó
 - Trong `cyclone_word`:
 - `$v0`: kết quả trả về của chương trình
 - `$t0`: i (ban đầu bằng 0)
 - `$t1`: $j = n - 1$

6.3 Ý nghĩa của chương trình con

- `cyclone_word`: kiểm tra xem từ đó có phải Cyclone word hay không.

6.4 Source code

```
1 #-----
2 # Project 24: Cyclone Word (Challenge)
3 # @input: mot tu bat ky luu vao c_word
4 # @idea:
5 #   TH1: Neu tu nhap vao co do dai bang 0 hoac bang 1 => TRUE
6 # TH2: Neu tu nhap vao co do dai > 1:
7 #       + Su dung vong lap co i bat dau bang 0 (Vi tri dau tien)
8 #       + va j = n - 1 (vi tri cuoi cung cua tu do)
9 #       + Thuc hien so sanh c_word[i] voi c_word[j] va c_word[i+1] voi
10 #        c_word[j] va cu tiep tục cho den khi i >= j
11 #        Neu c_word[i] <= c_word[j] va c_word[j] <= c_word[i+1] =>
12 #        TRUE
13 #-----
14 .data
15 # Chu y so luong ki tu cua tu
16 c_word: .asciiz "adjourned" # input dau vao
17 true: .asciiz "True"
18 false: .asciiz "False"
19 .text
20
21 main:
22     la $a0, c_word
23
24     # So ki tu cua tu
25     addi $a1, $zero, 9
26
27     jal cyclone_word
28     nop
29
30     beq $v0, $zero, false_answer
31
32 true_answer:
33     li $v0, 55
34     la $a0, true
35     syscall
36     j end_main
37
38 false_answer:
39     li $v0, 55
40     la $a0, false
41     syscall
42
43 end_main:
44     li $v0, 10
45     syscall
46
47 #-----
48 # @cyclone_word: Kiem tra xem co phai la cyclone word hay khong
```



```

49 #-----
50
51 cyclone_word:
52 # If length of word = 0 or = 1 => TRUE
53 beq $a1, $zero, return_true
54 addi $t0, $zero, 1
55 beq $a1, $t0, return_true
56
57 addi $t0, $zero, 0      # i: 0
58 sub $t1, $a1, 1        # j: n - 1
59 loop:
60 slt $t2, $t0, $t1      # if i >= j => return 0
61 beq $t2, $zero, return_true
62
63 add $s1, $a0, $t0
64 lb $s1, 0($s1)          # c_word[i]
65 add $s2, $a0, $t1
66 lb $s2, 0($s2)          # c_word[j]
67 sle $t3, $s1, $s2      # if c_wrod[i] > c_word[j] false => 0
68 beq $t3, $zero, return_false # if return 0
69
70 addi $t4, $t0, 1
71 add $t4, $a0, $t4
72 lb $s3, 0($t4)          # c_word[i+1]
73 sle $t5, $s2, $s3      # if c_word[j] > c_word[i+1] false => 0
74 beq $t5, $zero, return_false
75
76 addi $t0, $t0, 1        # i = i + 1
77 addi $t1, $t1, -1       # j = j - 1
78 j loop
79 end_loop:
80
81 return_true:
82 addi $v0, $zero, 1
83 jr $ra
84 return_false:
85 addi $v0, $zero, 0
86 jr $ra
87
88 #-----
89 # END
90 #-----

```

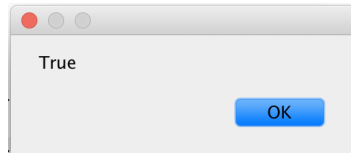
Project 24 - Cyclone Word (Challenge)

6.5 Kết quả chạy chương trình

- Đầu vào:

```
1 c_word: .asciiz "adjourned"
```

- Kết quả:



- Đầu vào:

```
1 c_word: .asciiz "settled"
```

- Kết quả:

