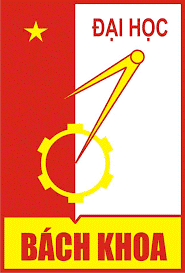
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**--------------------------------------------------------------------------**

***GVHD: TS. Lê Xuân Thành***

***Nhóm: 4***

***Thành Viên: Nguyễn Trung Thành – 20176874***

***Hoàng Thị Thu Trang – 20176891***

***Môn học: Thực hành Kiến Trúc Máy Tính***

***Mã Lớp: 113835***

Báo Cáo Mini-Projects

# **Thông tin về mini – projects**

1. Đề bài được phân công cho từng sinh viên như sau:

* Nguyễn Trung Thành

Project 7.

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 sortByHeight(a) = [-1, 150, 160, 170, -1, -1, 180, 190].

* Hoàng Thị Thu Trang

Project 8.

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. Đề bài làm thêm:

Project 21.

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 digitDegree(n) = 0;

● For n = 100, the output should be digitDegree(n) = 1. 1 + 0 + 0 = 1.

● For n = 91, the output should be digitDegree(n) = 2. 9 + 1 = 10 -> 1 + 0 = 1.

Project 23.

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

Project 24.

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

Mars4\_5

1. Đường Link code:

<https://github.com/thanhhff/Computer-Architecture-Lab/tree/master/Mini-project?fbclid=IwAR0TpNoyEM8uP5kTnsVSQHFruJYQ60EpeL-2UIbE0mlHOsXH2Mcki3I6GRo>

Mục Lục

[**I.** **Thông tin về mini – projects** 1](#_Toc39998344)

[**II.** **Project 7 – sortByHeight** 4](#_Toc39998345)

[**III.** **Project 8 – students** 5](#_Toc39998346)

[**IV.** **Project 21 – digit degree** 6](#_Toc39998347)

[**V.** **Project 23 – Surpassing words** 7](#_Toc39998348)

[**VI.** **Project 24 – Cyclone Word (challenge)** 8](#_Toc39998349)

# **Project 7 – sortByHeight**

**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 sortByHeight(a) = [-1, 150, 160, 170, -1, -1, 180, 190].

**Đường Link Code:** <https://github.com/thanhhff/Computer-Architecture-Lab/blob/master/Mini-project/project_7.asm>

**Phân tích cách thực hiện:**

**Ý nghĩa của các thanh ghi được sử dụng:**

# **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.

**Đường link code:** <https://github.com/thanhhff/Computer-Architecture-Lab/blob/master/Mini-project/project_8.asm>

**Phân tích cách thực hiện:**

* Ý tưởng thực hiện:
* Nhập số lượng sinh viên ( số nguyên dương)
* Nhập lần lượt tên và điểm sinh viên với điều kiện: 0 <= điểm <= 10
* Sắp xếp danh sách sinh viên theo thứ tự tăng dần của điểm sử dụng thuật toán Bubble sort.
* Các bước 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<= điểm <=10 thì nhập lại.

* Bước 3: Hiển thị thông tin sinh viên vừa nhập vào.
* Bước 4: Sắp xếp điểm sinh viên:

+ j = o //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.

**Ý nghĩa của các thanh ghi được sử dụng:**

* $s0: Tổng số sinh viên.
* $s1: Lưu địa chỉ của mảng students.
* $s3: Lưu địa chỉ của mảng students.
* $a0: Địa chỉ của mảng.
* $a1: vị trí lưu tên.
* $a2: giới hạn độ dài của tên.
* $t0: biến đếm của sinh viên được nhập thông tin.
* $t1: chứa giá trị của mảng được gán.
* $t2: Biến i.
* $t3: Biến j.
* $ra: return address – địa chỉ trở về

**Ý nghĩa của các chương trình con nếu có:**

* \_\_read\_info\_student: nạp địa chỉ của mảng.
* count: nhập số lượng sinh viên.
* name: nhập tên sinh viên với giới hạn là 46 kí tự.
* mark: nhập điểm sinh viên với điều kiện 0 <= điểm <=10.
* \_\_show\_student: Hiển thị danh sách sau khi sắp xếp.

# **Project 21 – digit degree**

**Đề bài:** 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.

**Đường link Code:** <https://github.com/thanhhff/Computer-Architecture-Lab/blob/master/Mini-project/project_21.asm>

**Ý nghĩa của các thanh ghi được sử dụng:**

**Ý nghĩa của các chương trình con nếu có:**

# **Project 23 – Surpassing words**

**Đề bài:** 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.

**Đường link Code:** <https://github.com/thanhhff/Computer-Architecture-Lab/blob/master/Mini-project/project_23.asm>

**Ý nghĩa của các thanh ghi được sử dụng:**

**Ý nghĩa của các chương trình con nếu có:**

# **Project 24 – Cyclone Word (challenge)**

**Đề bài:** Cyclone words are English words that have a sequence of characters in alphabetical order when following a cyclic pattern.

**Đường link Code:** <https://github.com/thanhhff/Computer-Architecture-Lab/blob/master/Mini-project/project_24.asm>

**Ý nghĩa của các thanh ghi được sử dụng:**

**Ý nghĩa của các chương trình con nếu có:**