

# Mini-projects Course 20192

Select one of the following problems, write an assembly program.

**Projects** (1). Tạo một chương trình để nhập một dòng văn bản từ bàn phím và kiểm tra xem nó có phải là palindrome hay không. Ví dụ: "abc121cba" là một palindrome. Lưu trữ tất cả các palindromes mà người dùng đã nhập vào bộ nhớ, để đảm bảo rằng người dùng không trùng lặp các palindromes.

(1). Create a program to input a text line from the keyboard and test if it is a palindrome. For example: "abc121cba" is a palindrome. Store all palindromes which the user typed into the **memory**, to make sure that the user does not duplicate palindromes.

(2). Tìm tất cả các số nguyên tố (chẳng hạn như 2, 3, 5, 7, v.v.) trong một phạm vi từ số nguyên N đến số nguyên M. N, M được nhập từ bàn phím.

(2). Find all prime numbers (such as 2, 3, 5, 7, etc) in a range from the integer N to the integer M. N, M is entered from the keyboard.

(3\*). Create a program to convert from number to text, in English or Vietnamese (choice 1 of 2). The number in range from 0 to 999 999 999

For example: (3 \*). Tạo chương trình chuyển đổi từ số sang văn bản, bằng tiếng Anh hoặc tiếng Việt (lựa chọn 1 trong 2). Dãy số từ 0 đến 999 999 999

Input: 1432

Ví dụ:

Đầu vào: 1432

Đầu ra: một nghìn bốn trăm ba mươi hai

Output: one thousand four hundred and thirty-two

(4). Create a program to:

- Input an array of integers from the keyboard.

- Find the maximum element of the array.

- Calculate the number of elements in the range of (m, M). Range m, M are inputted from the keyboard.

(4). Tạo một chương trình để:

- Nhập một mảng số nguyên từ bàn phím.

- Tìm phần tử lớn nhất của mảng.

- Tính số phần tử trong khoảng (m, M). Phạm vi m, M được nhập từ bàn phím.

(5). Write a program to get decimal numbers, display those numbers in binary and hexadecimal.

(6) Given an array of word elements and the number of elements, write a procedure to find the pair of adjacent elements that has the largest product and return that product.

(6) Cho một mảng gồm các phần tử và số phần tử, hãy viết thủ tục tìm cặp phần tử liên nhau có tích lớn nhất và trả về tích đó.

Ví dụ: Đối với inputArray = [3, 6, -2, -5, 7, 3], đầu ra phải là tích của 7 và 3 (21)

Example: For inputArray = [3, 6, -2, -5, 7, 3], the output should be the product of 7 and 3 (21)

(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 non-descending order without moving the trees. People can be very tall!

(7) Một số người đang đứng thành hàng trong công viên. Giữa chúng có những cây không thể di chuyển được. Nhiệm vụ của bạn là sắp xếp lại những người theo chiều cao của họ theo thứ tự không giảm dần mà không di chuyển cây. Con người có thể rất cao!

Ví dụ: Đối với  $a = [-1, 150, 190, 170, -1, -1, 160, 180]$ , đầu ra phải là  $\text{sortByHeight}(a) = [-1, 150, 160, 170, -1, -1, 180, 190]$ .

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

**(8).** Write a program to:

You must create 1 variable (number students) and 2 arrays (student name, mark) to store the input data

- Input the number of students in class.
- Input the name of students in class, Math mark (0 → 10)
- Sort students due to their mark.
- Print list of students to screen

(số 8). Viết chương trình để:

Bạn phải tạo 1 biến (số sinh viên) và 2 mảng (tên sinh viên, dấu) để lưu dữ liệu đầu vào

- Nhập số lượng học sinh trong lớp.
- Nhập tên học sinh trong lớp, điểm Toán (0 → 10)
- Xếp loại học sinh do có dấu.
- In danh sách sinh viên ra màn hình

**(9).** Write a program to:

Assume that you already have 1 variable (number students) and 2 arrays (student name, mark) in memory

- Read in the number of students in the class.
- Read information about each student, including: Name, Math mark.
- List the names of all students who have not passed the Math exam.

(9). Viết chương trình để:

Giả sử rằng bạn đã có 1 biến (số học sinh) và 2 mảng (tên học sinh, dấu) trong bộ nhớ

- Đọc trong số học sinh trong lớp.
- Đọc thông tin về từng học sinh, bao gồm: Tên, điểm Toán.

Toán.

- Liệt kê tên tất cả học sinh không đạt môn Toán.

**(10\*)** Write a program that gets an integer  $i$  from the user and creates the table shown below on the screen (example inputs provided). Subroutines are required for power, square, and hexadecimal (in 32 bit arithmetic, attend to overflowed results). Hint: Hexadecimal can be done with shifts and masks because the size is 32 bits.

$i$	$\text{power}(2,i)$	$\text{square}(i)$	Hexadecimal( $i$ )
10	1024	100	0xA
7	128	49	0x7
16	65536	256	0x10

(10 \*) Viết chương trình lấy số nguyên  $i$  từ người dùng và tạo bảng hiển thị bên dưới trên màn hình (cung cấp đầu vào ví dụ). Các chương trình con được yêu cầu cho lũy thừa, bình phương và thập lục phân (trong số học 32 bit, chú ý đến kết quả bị tràn). Gợi ý: Hệ thập lục phân có thể được thực hiện với ca và mặt nạ vì kích thước là 32 bit.

**(11)** Programming an application to convert names from LastName-FirstName to FirstName-LastName.

1. Input name of 2 students with space character, for example "Vu Thi XYZ". Store them into the memory.
2. Change them from LastName-FirstName to FirstName-LastName, for example "XYZ Vu Thi"

(11) Lập trình ứng dụng chuyển đổi tên từ LastName-FirstName thành FirstName-LastName.

1. Nhập tên của 2 sinh viên bằng ký tự khoảng trắng, ví dụ "Vu Thi XYZ". Lưu trữ chúng vào bộ nhớ.
2. Thay đổi chúng từ LastName-FirstName thành FirstName-LastName, ví dụ "XYZ Vu Thi"
3. In tên ra màn hình.

(12) Phân tích cú pháp một chuỗi ASCII thành số nhị phân  
Viết một hàm chuyển đổi một chuỗi các chữ số ASCII thành một số nguyên 32 bit. Hàm sẽ nhận dưới dạng một đối số là địa chỉ bắt đầu của chuỗi và phải trả về một số nguyên 32 bit chứa giá trị nguyên của chuỗi. Giả sử rằng chuỗi là một chuỗi ASCIIZ, tức là, kết thúc bằng ký tự rỗng (mã ASCII 0). Bạn không cần phải kiểm tra lỗi trong chuỗi, tức là bạn có thể giả sử chuỗi chỉ chứa các ký tự từ '0' đến '9' (tức là mã ASCII tương ứng của chúng) và sẽ không đại diện cho số âm hoặc không phải số thập phân giá trị hoặc một số quá lớn. Ví dụ: a\_to\_i được gọi với đối số "12345" sẽ trả về số nguyên 12345. atoi()

3. Print names to the screen.

### (12) Parsing an ASCII string to binary number

Write a function that converts a string of ASCII digits into a 32-bit integer. The function will receive as an argument the starting address of the string and must return a 32-bit integer containing the integer value of the string. Assume that the string is an ASCIIZ string, i.e., ends with the null character (ASCII code 0). You do not need to check for errors in the string, i.e., you may assume the string contains only characters '0' through '9' (i.e., their corresponding ASCII codes), and will not represent a negative number or a non-decimal value or too large a number. For example, a\_to\_i called with the argument "12345" will return the integer 12345. **atoi()**

(13) Ticket numbers usually consist of an even number of digits. A ticket number is considered lucky if the sum of the first half of the digits is equal to the sum of the second half. Given a ticket number n, determine if it's lucky or not.

Example

For n = 1230, the output should be isLucky(n) = true;

For n = 239017, the output should be isLucky(n) = false.

(13) Số vé thường bao gồm một số chữ số chẵn. Một số vé được coi là may mắn nếu tổng của nửa số đầu tiên bằng tổng của nửa số thứ hai. Cho một vé số n, hãy xác định xem nó có may mắn hay không.

Ví dụ  
Với n = 1230, kết quả đầu ra phải là isLucky (n) = true;  
Đối với n = 239017, kết quả đầu ra phải isLucky (n) = false.

(14). Cho hai chuỗi, hãy tìm số ký tự chung giữa chúng.

(14). Given two strings, find the number of common characters between them.

Ví dụ: Đối với s1 = "aabcc" và s2 = "adcaa", đầu ra phải là commonCharacterCount (s1, s2) = 3.

Các chuỗi có 3 ký tự chung - 2 "a" s và 1 "c".

Example: For s1 = "aabcc" and s2 = "adcaa", the output should be

commonCharacterCount(s1, s2) = 3. Strings have 3 common characters - 2 "a"s and 1 "c".

(15) Bạn được cung cấp một mảng các số nguyên. Trên mỗi lần di chuyển, bạn được phép tăng chính xác một trong các phần tử của nó lên một. Tìm số lần di chuyển tối thiểu cần thiết để có được một chuỗi tăng dần từ đầu vào.

(15) You are given an array of integers. On each move you are allowed to increase exactly one of its elements by one. Find the minimal number of moves required to obtain a strictly increasing sequence from the input.

Ví dụ: Đối với inputArray = [1, 1, 1], đầu ra phải là arrayChange (inputArray) = 3. [1,2,3]

Example: For inputArray = [1, 1, 1], the output should be arrayChange(inputArray) =

3. [1,2,3] Số lần di chuyển tối thiểu cần thiết để có được một chuỗi tăng dần từ inputArray. Nó được đảm bảo rằng đối với các trường hợp thử nghiệm nhất định, câu trả lời luôn phù hợp với kiểu số nguyên 32 bit có dấu.

The minimal number of moves needed to obtain a strictly increasing sequence from inputArray. It's guaranteed that for the given test cases the answer always fits signed 32-bit integer type.

(16) Given a sequence of integers as an array, determine whether it is possible to obtain a strictly increasing sequence by **removing no more than one element** from the array.

Note: sequence a0, a1, ..., an is considered to be strictly increasing if a0 < a1 < ... < an. Sequences containing only one element are also considered to be strictly increasing.

(16) Cho một dãy số nguyên dưới dạng một mảng, hãy xác định xem liệu có thể có được một dãy số tăng dần hay không bằng cách loại bỏ không quá một phần tử khỏi mảng.

Lưu ý: dãy a0, a1, ..., an được coi là tăng nghiêm ngặt nếu a0 < a1 < ... < an. Các dãy chỉ chứa một phần tử cũng được coi là tăng nghiêm ngặt.

Ví dụ:

- Đối với dãy = [1, 3, 2, 1], đầu ra phải là

nearIncreasingSequence (chuỗi) = false. Không có một phần tử nào trong mảng này có thể bị loại bỏ để có được một chuỗi tăng dần.

- Đối với dãy = [1, 3, 2], đầu ra phải là

**Example:** nearIncreasingSequence (chuỗi) = true. Bạn có thể xóa 3 khỏi mảng để nhận được chuỗi tăng dần [1, 2]. Ngoài ra, bạn có thể loại bỏ 2 để có được trình tự tăng dần [1, 3].

- For sequence = [1, 3, 2, 1], the output should be

almostIncreasingSequence(sequence) = false. There is no one element in this array that can be removed in order to get a strictly increasing sequence.

- For sequence = [1, 3, 2], the output should be

almostIncreasingSequence(sequence) = true. You can remove 3 from the array to get the strictly increasing sequence [1, 2]. Alternately, you can remove 2 to get the strictly increasing sequence [1, 3].

(17) Viết chương trình nhập vào một chuỗi. Trích xuất các ký tự số và hiển thị ra màn hình theo thứ tự nghịch đảo bằng cách sử dụng ngăn xếp.

**(17)** Write a program that inputs a string. Extract number characters and show to screen in inverse order using stack.

(18) Hai mảng được gọi là tương tự nếu một mảng có thể nhận được từ mảng khác bằng cách hoán đổi nhiều nhất một cặp phần tử trong một mảng.

**(18)** Two arrays are called similar if one can be obtained from another by swapping

at most one pair of elements in one of the arrays.

- Đối với a = [1, 2, 3] và b = [1, 2, 3], đầu ra phải

areSimilar (a, b) = true. Các mảng bằng nhau, không cần

Cho hai mảng a và b, hãy kiểm tra xem chúng có giống nhau không.

hoán đổi bất kỳ phần tử nào.

Given two arrays a and b, check whether they are similar. Example:

- For a = [1, 2, 3] and b = [1, 2, 3], the output should be areSimilar(a, b) = true. The arrays are equal, no need to swap any elements.

- Đối với a = [1, 2, 3] và b = [2, 1, 3], kết quả đầu ra phải areSimilar (a, b) = true. Ta có thể thu được b từ a bằng cách hoán đổi 2 và 1 trong b.

- For a = [1, 2, 3] and b = [2, 1, 3], the output should be areSimilar(a, b) = true. We can obtain b from a by swapping 2 and 1 in b.

- Đối với a = [1, 2, 2] và b = [2, 1, 1], đầu ra phải areSimilar (a, b) = false. Bất kỳ hoán đổi nào của hai phần tử bất kỳ trong a hoặc trong b sẽ không làm cho a và b bằng nhau.

- For a = [1, 2, 2] and b = [2, 1, 1], the output should be areSimilar(a, b) = false. Any swap of any two elements either in a or in b won't make a and b equal.

**(19)** Write a program that input some variable names. Check if variable names consist only of English letters, digits and underscores and they can't start with a digit.

(19) Viết chương trình nhập một số tên biến. Kiểm tra xem tên biến chỉ bao gồm các chữ cái tiếng Anh, chữ số và dấu gạch dưới và chúng không được bắt đầu bằng chữ số.

**Example**

- For name = "var\_1\_\_Int", the output should be variableName (name)

= true;

- For name = "qq-q", the output should be variableName (name) =

false;

- For name = "2w2", the output should be variableName (name) =

false.

- Đối với name = "var\_1\_\_Int", đầu ra phải là variableName (name) = true;
- Đối với name = "qq-q", đầu ra phải là variableName (name) = false;
- Đối với name = "2w2", đầu ra phải là variableName (name) = false.

**(20)** Given a string which consists of lower alphabetic characters (a-z), count the number of different characters in it.

**Example:** For s = "cabca", the output should be differentSymbolsNaive(s) = 3.

There are 3 different characters a, b and c.

(20) Cho một chuỗi bao gồm các ký tự chữ cái thường (a-z), đếm số ký tự khác nhau trong đó.

Ví dụ: Đối với s = "cabca", đầu ra phải khácSymbolsNaive (s) = 3. Có 3 ký tự a, b và c khác nhau.

(21) Hãy xác định bậc chữ số của một số nguyên dương là số lần chúng ta cần thay số này bằng tổng các chữ số của nó cho đến khi chúng ta nhận được một số có một chữ số. Cho một số nguyên, tìm độ của chữ số của nó.  
Ví dụ

- Với  $n = 5$ , đầu ra phải là  $\text{digitDegree}(n) = 0$ ;
- Với  $n = 100$ , kết quả đầu ra phải là  $\text{digitDegree}(n) = 1$ .  $1 + 0 + 0 = 1$ .
- Với  $n = 91$ , đầu ra phải là chữ số  $\text{digitDegree}(n) = 2$ .  $9 + 1 = 10 \rightarrow 1 + 0 = 1$ .

(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  $\text{digitDegree}(n) = 0$ ;
- For  $n = 100$ , the output should be  $\text{digitDegree}(n) = 1$ .  $1 + 0 + 0 = 1$ .
- For  $n = 91$ , the output should be  $\text{digitDegree}(n) = 2$ .  $9 + 1 = 10 \rightarrow 1 + 0 = 1$ .

(22) Cyclone Word (challenge)

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

(22) Cyclone Word (thử thách)

Cyclone words là từ tiếng Anh có một chuỗi các ký tự theo thứ tự bằng chữ cái khi tuân theo một mô hình tuần hoàn.

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.

Viết một hàm để xác định xem một từ được truyền vào một hàm có phải là một từ lốc xoáy hay không. Bạn có thể giả định rằng từ chỉ được tạo bởi các ký tự chữ cái và được phân tách bằng khoảng trắng.

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

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

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

### Cách thực hiện:

- Mỗi sinh viên chọn một bài
- Cách phân chia nhóm do SV tự sắp xếp và gửi giáo viên
- Các bài tập của mỗi nhóm sẽ được gán ngẫu nhiên cho các nhóm
- Thời gian làm bài: tuần 9, 10. Sinh viên làm việc ở nhà
- Tuần kiểm tra: tuần 11

### Kết quả thực hiện:

- Viết báo cáo trình bày
  - o phân tích cách thực hiện
  - o ý nghĩa của các thanh ghi được sử dụng
  - o ý nghĩa của các chương trình con nếu có
- Mã nguồn chương trình
  - o Có chú thích trong mã nguồn. Ví dụ

#-----

# @brief    Kiểm tra hiệu ứng của một scene bằng cách polling

# @param[in] Scene\_Ptr biến toàn cục, có giá trị là địa chỉ của

#            Scene vừa được thiết lập, địa chỉ của Scene 1,2,3,4

# @param[in] Scene\_Len biến toàn cục, cho biết độ dài

# @return    \$v0    Thanh ghi chứa mã lỗi

# @note      Phải gọi hàm SetScene trước

#-----

.ent Test\_Scene

Test\_Scene:

    jal NextFrame

    nop

```
jal ShowScene  
  
nop  
  
j Test_Scene  
  
.end Test_Scene
```

### **Cách kiểm tra**

- Từng nhóm 2 sinh viên gặp giáo viên để kiểm tra.
- Khi gặp giáo viên kiểm tra, các SV có thể dùng máy tính laptop, nếu không có thể mời giáo viên tới bàn máy tính tại Lab.
- Mỗi SV sẽ trình bày trả lời các câu hỏi của giáo viên về 1 (trong 2 bài tập của nhóm).
- Nội dung chính để kiểm tra:
  1. Chạy chương trình và cho kết quả đúng
  2. Hạn chế được các lỗi thao tác nhập liệu của người dùng (ví dụ nếu người dùng cố tính nhập giá trị số nguyên là hello)
  3. Hạn chế được các số quá lớn, ngoài phạm vi chương trình (ví dụ, tính giai thừa của 1 tỷ)
  - 4. Hiểu được ý nghĩa của các lệnh sử dụng trong bài**
  - 5. Trả lời được các câu hỏi lý thuyết ứng với các lệnh trong chương trình (khuôn dạng của lệnh này là gì? Lệnh này mất bao nhiêu chu kì để thực hiện...)**
  6. Chỉnh sửa trực tiếp được chương trình.
  7. Mã nguồn có chú thích đầy đủ, rõ ràng.