Mini-projects Course 2019-02

Projects

Select two of the following problems, write assembly program.

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

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

(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:

Input: 1432

Output: one thousand four hundred and thirty two

(4). Create a program to:

- Input an array of integer from keyboard.
- Find the maximum element of array.
- Calculate the number of elements in the range of (m, M). Range m, M are integer numbers entered from keyboard.

(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.

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.

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].

(8). Write a program to:

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

(9). Write a program to:

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

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

(11). 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 don't 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.

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

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".

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

(15) Write a program that input some variable names. Check if variable names consist only of English letters, digits and underscores and they cannot start with a digit.

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.

(16) 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.

(17) You are taking part in an Escape Room challenge designed specifically for programmers. In your efforts to find a clue, you have found a binary code written on the wall behind a vase and realized that it must be an encrypted message. After some thought, your first guess is that each consecutive 8 bits of the code stand for the character with the corresponding ASCII code.

Assuming that your hunch is correct, decode the message.

Example

For code = "010010000110010101101100011011000110111100100001",

the output should be messageFromBinaryCode(code) = "Hello!".

(18) Define a *word* as a sequence of consecutive English letters. Find the longest *word* from the given string.

Example

For text = "Ready, steady, go!", the output should be
longestWord(text) = "steady".

(19) Given some integer, find the maximal number you can obtain by deleting exactly one digit of the given number.

Example

For n = 152, the output should be deleteDigit(n) = 52;

For n = 1001, the output should be deleteDigit(n) = 101.

Cách thực hiện:

- Mỗi nhóm gồm 2 sinh viên.
- Mỗi nhóm thực hiện 2 bài tập.
- Cách phân chia nhóm do SV tự sắp xếp.
- Các bài tập của mỗi nhóm sẽ được gán ngẫu nhiên.

Kết quả thực hiện:

- Viết báo cáo trình bày:
  - Phân tích cách thực hiện
  - Ý nghĩa của các thanh ghi được sử dụng
  - Ý nghĩa của các chương trình con nếu có
  - Ảnh chụp màn hình kết quả thực hiện
- Mã nguồn chương trình:
  - Có chú thích trong mã nguồn. Ví dụ

```
#-----------------------------------------------------------------------
# @brief      Kiem tra hieu ung cua mot scene bang cach polling
# @param[in]  Scene_Ptr bien toan cuc, co gia tri la dia chi cua
#             Scene vua duoc thiet lap, dia chi cua Scene1,2,3,4
# @param[in]  Scene_Len bien toan cuc, cho biet do dai
# @return     $v0       Thanh ghi chua ma loi
# @note       Phai goi ham SetScene truoc
#-----------------------------------------------------------------------
.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). Đồng thời sinh viên đó cũng phải nắm được về bài tập còn lại.
- 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 chuỗi ký tự).

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.