**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

Logo

Description automatically generated**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP VỀ NHÀ**

**MÔN HỌC: HỆ THỐNG MÁY TÍNH**

**HW1: Data Representation**

Họ và tên sinh viên: Nguyễn Huy Hoàng

MSSV: 21120458

Lớp: 21\_4

Giảng viên: Chung Thùy Linh

Thành phố Hồ Chí Minh, tháng 3 năm 2021

**MỤC LỤC**

[**A. Programming:** 1](#_Toc129631013)

[**1.** **Input a singed integer X (16-bit)** 1](#_Toc129631014)

[**2.** **Output the binary bit pattern of X (2’s complement form)** 1](#_Toc129631015)

[ **Ảnh chụp màn hình kết quả:** 1](#_Toc129631016)

[**3.** **Input a single precision Y** 2](#_Toc129631017)

[**4.** **Output the binary bit pattern of Y** 2](#_Toc129631018)

[ **Ảnh chụp màn hình kết quả** 3](#_Toc129631019)

[ **Source code:** Integer 3](#_Toc129631020)

[**B. Complete these exercises:** 4](#_Toc129631021)

[**1.** **Sắp xếp các loại theo thứ tự tăng dần về kích thước:** 4](#_Toc129631022)

[**2.** **Để trả về giá trị interger 61 (0x3d hexadecimal) khi gọi hàm, tương đương với 0x11 XOR n 0x3d, ta có thể tìm giá trị của đối số n bằng cách thực hiện thao tác XOR trên 0x11 và 0x3d** 4](#_Toc129631023)

[**3.** **Kích thước của struct s:** 5](#_Toc129631024)

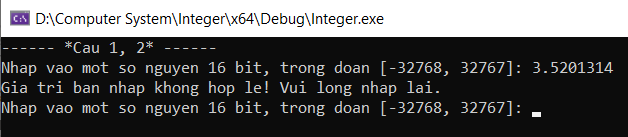
[**4. Chuyển các số dưới dạng hexadecimal sang dạng decimal sử dụng 32-bit IEE 754** 5](#_Toc129631025)

# **A. Programming:**

## **1.** **Input a singed integer X (16-bit)**

* Ý tưởng:
* Viết hàm inputSignedInteger, truyền vào tham số x
* Sử dụng vòng lặp vô hạn để yêu cầu người dùng nhập vào giá trị hợp lệ.
* Dùng cin.peek() để kiểm tra xem phần thập phân của giá trị nhập vào có bằng 0 hay không, nếu có phần thập phân thì giá trị nhập vào không phải là số nguyên 🡪 Yêu cầu người dùng nhập lại
* Tiếp theo kiểm tra điều kiện về giới hạn của số nguyên 16 bit, nếu nằm trong đoạn [-32768, 32767] thì hợp lệ, ngược lại thì yêu cầu người dùng nhập lại

## **2.** **Output the binary bit pattern of X (2’s complement form)**

* Ý tưởng:
* Viết hàm chuyển đổi số nguyên sang dạng binary, truyền vào tham số x
* Sử dụng toán tử ~ để đảo ngược tất cả các bit của số nguyên X, sau đó cộng thêm 1 để chuyển đổi sang dạng bù 2
* Chạy vòng lặp for từ 15 về 0 (16 bit của số nguyên X) và lấy ra bit tại vị trí thứ i bằng cách dùng toán tử >> để dịch bit của x sang phải i vị trí, rồi AND (&) với 1 để lấy bit tại vị trí thứ i và in kết quả bù 2 ra màn hình
* **Ảnh chụp màn hình kết quả:**
* Nhập một số không phải số nguyên, ví dụ 3.5201314. Chương trình sẽ yêu cầu người dùng nhập lại
* Nhập một số nguyên không phải 16 bit (không nằm trong [-32768, 32767], chương trình cũng sẽ yêu cầu nhập lại



* Nhập vào một số nguyên dương hợp lệ, ví dụ 520. Chương trình sẽ in ra kết quả dạng nhị phân và dạng bù 2 của 520.

Text

Description automatically generated

* Nhập vào một số nguyên âm hợp lệ, ví dụ -6. Chương trình sẽ in ra kết quả dạng nhị phân và dạng bù 2 của -6.

Text

Description automatically generated

## **3.** **Input a single precision Y**

* Ý tưởng:
* Viết hàm inputSinglePrecision, truyền vào tham số y để cho phép người dùng nhập vào giá trị của Y.
* Đưa vào vòng lặp do-while để kiểm tra điều kiện hợp lệ của Y (không phải là số vô cực, không phải NaN, đảm bảo nằm trong khoảng giá trị của số single precision), yêu cầu người dùng nhập lại cho tới khi giá trị nhập vào là hợp lệ.

## **4.** **Output the binary bit pattern of Y**

* Ý tưởng:
* Viết hàm printBinary, truyền vào tham số y để in ra binary bit pattern của Y
* Sử dụng con trỏ và ép kiểu để chuyển đổi Y sang dạng unsigned int
* Sử dụng vòng lặp for (chạy từ 31 về 0) để duyệt từng bit của số Y từ bit cao nhất đến bit thấp nhất.
* Sử dụng toán tử >> (dịch bit) để lấy giá trị của từng bit tại vị trí tương ứng, sau đó AND (&) với 1 để trích xuất giá trị của bit tại vị trí đó. Với mỗi bit, nếu nó bằng 1, in ra ký tự “1”, ngược lại in ra ký tự “0”
* **Ảnh chụp màn hình kết quả**
* Text

  Description automatically generatedNhập một giá trị không hợp lệ ngoài giới hạn của số single precision

**Text

Description automatically generated**

* Text

  Description automatically generatedNhập vào một giá trị âm hợp lệ, ví dụ -5.20
* Nhập vào một giá trị dương hợp lệ, ví dụ 23.25

**Text

Description automatically generated**

* **Source code:** [Integer](https://studenthcmusedu-my.sharepoint.com/:f:/g/personal/21120458_student_hcmus_edu_vn/Eji2cdAbCetMg7UcAFGSL88BkXkzjrCAr7h4oeePRV1GIw?e=p5kCe0)

# **B. Complete these exercises:**

## **1.** **Sắp xếp các loại theo thứ tự tăng dần về kích thước:**

**6 < 1 < 4 < 2 < 3 < (or =) 5**

*Giải thích:*

* Kích thước của kiểu dữ liệu char là 1 byte
* Kích thước của kiểu uint8\_t là 1 byte => Kích thước của struct minipont = 3\*1 = 3 bytes
* Kích thước của kiểu int là 4 bytes
* Kích thước của kiểu unsigned short là 2 byte => Kích thước của unsigned short[1] = 1\*2 = 2 bytes
* Kích thước của kiểu char\*\* là 4 bytes hoặc 8 bytes
* Kích thước của kiểu double là 8 bytes. Tuy nhiên kích thước của double[0] = 0 \* 8 = 0

## **2.** **Để trả về giá trị interger 61 (0x3d hexadecimal) khi gọi hàm, tương đương với 0x11 XOR n 0x3d, ta có thể tìm giá trị của đối số n bằng cách thực hiện thao tác XOR trên 0x11 và 0x3d**

* Chuyển đổi 0x11 và 0x3d (đang ở dạng hexadecimal) sang dạng nhị phân (binary)

0x11 = 11 (hexa) = 0001 0001 (binary)  
 0x3d = 3d (hexa) = 3 13 (hexa) = 0011 1101 (binary)

* Thực hiện phép XOR cho từng cặp bit tương ứng của 2 số ở dạng binary

0 XOR 0 = 0

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 1 = 0

0 XOR 1 = 1

0 XOR 1 = 1

0 XOR 0 = 0

1 XOR 1 = 0

* Gộp kết quả từng cặp bit lại với nhau 🡪 ta được số nhị phân mới: 0010 1100. Chuyển số nhị phân này sang dạng hexadecimal

0010 1100 (binary) = 2 12 = 2c (hexadecimal)

* Vậy đối số cần truyền vào hàm là 0x2c (= 44 (decimal))

## **3.** **Kích thước của struct s:**

**struct** **s** {

char f1[7];

char \*f2;

short f3;

int f4;

};

* Kích thước của f1 = 7\*1 = 7 bytes
* Kích thước của f2 = 16 bytes (với 16-byte alignment). Ta đệm thêm 9 bytes vào sau ô nhớ chứa biến f1 để đảm bảo tính căn chỉnh cấu trúc dữ liệu.
* Kích thước của f3 = 4 bytes (đã được căn chỉnh)
* Kích thước của f4 = 8 bytes. Do đó cần đệm thêm 4 bytes vào sau ô nhớ chứa biến f3

🡺 Tổng kích thước của struct s là: 7 + 9 + 16 + 4 + 4 + 8 = 48 bytes

## **4. Chuyển các số dưới dạng hexadecimal sang dạng decimal sử dụng 32-bit IEE 754**

a. 0x4000 0000 = 0100 0000 0000 0000 0000 0000 0000 0000

**sign:** 0

**exponent:** 1000 0000

n = exponent – 127 = (1000 0000)2 – 127 = 128 – 127 = 1

**fraction:** 000 0000 0000 0000 0000 0000

1. 000 0000 0000 0000 0000 0000)­2 x 21

(10. 00 0000 0000 0000 0000 0000)2 x 20

+(10. 00 0000 0000 0000 0000 0000)2 🡪 +(xxxxxx)­10

***Result:*** 1x20 + 0x21 + 0x2-1 + 0x2-2 +…. = **+2.0**

b. 0x3d80 0000 = 0011 1101 1000 0000 0000 0000 0000 0000

**sign:** 0

**exponent:** 0111 1011

n = exponent – 127 = (0111 1011)2 – 127 = 123 – 127 = -4

**fraction:** 000 0000 0000 0000 0000 0000

1. 000 0000 0000 0000 0000 0000)­2 x 2-4

(0.0001 000 0000 0000 0000 0000 0000)2 x 24

(0.0001 000 0000 0000 0000 0000 0000)2 🡪 +(xxxxxx)­10

***Result:*** 1x2-4 = **+ 0.0625**

c. 0xc259 48b4 = 1100 0010 0101 1001 0100 1000 1011 0100

**sign:** 1

**exponent:** 1000 0100

n = exponent – 127 = (1000 0100)2 – 127 = 132 – 127 = 5

**fraction:**101 1001 0100 1000 1011 0100

(1.101 1001 0100 1000 1011 0100)2  x 25

(1 101 10. 01 0100 1000 1011 0100)2 x 20

-(1 101 10. 01 0100 1000 1011 0100)2  🡪 - (xxxxxx)10

***Result:*** 1x25 + 1x24 + 1x22 + 1x21 + 1x2-2 + 1x2-4 1x2-7 + 1x2-11 + 1x2-13 + 1x2-14  + 1x2-16 = **-54.32099915**