|  |  |
| --- | --- |
| **Họ và tên:** | **Lê Phạm Hoàng Trung** |
| **Mã số sinh viên:** | **21120157** |
| **Lớp:** | **Hệ thống máy tính - 21TNT** |

**BÁO CÁO BÀI TẬP 1**

**KHẢO SÁT SỐ NGUYÊN**

**I) Phần 1: Đánh giá**

**a. Bài 1) Viết chương trình cho phép chuyển đổi giữa các hệ cơ số sau (theo đề bài chỉ xử lý với số nguyên dương), source code bài 1 tên là exercise1.cpp:**

|  |  |  |
| --- | --- | --- |
| **Yêu cầu đề bài** | **Mức độ hoàn thành (%)** | **Chú thích** |
| Từ hệ 10 sang hệ 2 | 100% | - Viết hàm chuyển đổi từ hệ 10 sang hệ 2, với input là số nguyên hệ 10. Output sẽ là một chuỗi nhị phân trong hệ 2. |
| Từ hệ 10 sang hệ 16 | 100% | - Viết hàm chuyển đổi từ hệ 10 sang hệ 16, input là một số nguyên ở hệ 10. Output đầu ra là số ở hệ 16 dưới dạng chuỗi. |
| Từ hệ 2 sang hệ 10 | 100% | - Viết hàm chuyển đổi từ hệ 2 sang hệ 10, với input là số trong hệ 2 kiểu int, ví dụ input là: 1001101 thì sẽ để dưới dạng int và đưa vào hàm. Output sẽ là số ở hệ 10 dưới dạng chuỗi.  *- Khi nhập vào số dưới dạng nhị phân để chuyển đổi sang hệ 10, cần lược bỏ các số 0 ở đầu do input đang ở dạng integer. Ví dụ: 001010 thì sẽ nhập là 1010.* |
| Từ hệ 16 sang hệ 10 | 100% | - Viết hàm chuyển đổi từ hệ 16 sang hệ 10, với input là số ở hệ 16 dưới dạng chuỗi, ví dụ “A6”. Output sẽ là số ở dạng 10 dưới dạng chuỗi, vd “A6” -> “166”. |
| Từ hệ 2 sang hệ 16 | 100% | - Viết hàm chuyển từ hệ 2 sang hệ 16 bằng cách sử dụng lại các hàm ta đã định nghĩa ở trên. Input là một số ở hệ 2 viết dưới dạng số nguyên, ví dụ 10100110. Đầu tiên ta sẽ chuyển số ở hệ 2 sang hệ 10 bằng hàm Bin\_to\_Dec đã định nghĩa (tự viết trong yêu cầu chuyển từ hệ 2 sang hệ 10 ở trên), kết quả thu được là 1 số ở hệ 10 dưới dạng chuỗi. Sau đó ta sẽ sử dụng hàm Dec\_to\_Hex (cũng đã được viết trong yêu cầu chuyển từ hệ 10 sang hệ 16) để chuyển sang kết quả ở hệ 16 dưới dạng chuỗi.  *- Khi nhập vào số dưới dạng nhị phân để chuyển đổi sang hệ 16, cần lược bỏ các số 0 ở đầu do input đang ở dạng integer. Ví dụ: 001010 thì sẽ nhập là 1010.* |

**b. Bài 2) Nhập vào 2 số nhị phân 8 bit (số nguyên có dấu dạng bù 2) hãy thực hiện các phép tính theo menu sau, source code bài 2 tên là exercise2.cpp:**

1. Tính tổng hai dãy bit

2. Tính hiệu hai dãy bit

3. Tính tích hai dãy bit

4. TÍnh thương 2 dãy bit

|  |  |  |
| --- | --- | --- |
| **Yêu cầu đề bài** | **Mức độ hoàn thành (%)** | **Chú thích** |
| Tạo menu cho 4 yêu cầu | 100% | - Tạo menu cho 4 chức năng, người dùng chọn chức năng sau đó nhập số dưới dạng NHỊ PHÂN 8 BIT vào và chương trình sẽ thực hiện tính toán. |
| Tính tổng hai dãy bit | 100% | - Input là 2 số nhị phân dạng 8 bit, chương trình sẽ tính và xuất ra màn hình kết quả dưới dạng nhị phân 8 bit.  - Nếu tràn số sẽ báo lỗi OVERFLOW. |
| Tính hiệu hai dãy bit | 100% | - Input là 2 số nhị phân dạng 8 bit, chương trình sẽ tính và xuất ra màn hình kết quả dưới dạng nhị phân 8 bit.  - nếu tràn số sẽ báo lỗi OVERFLOW.  *LƯU Ý: Nếu tính toán 7 – 3, thì ta chỉ cần input số thứ nhất là 7, và số thứ hai là 3, chương trình sẽ tự thực hiện phép toán 7 – 3. Không input đầu vào là -3 cho số thứ hai, chương trình sẽ hiểu sai và thực hiện 7 – (-3*) |
| Tính tích hai dãy bit | 100% | - Input là 2 số nhị phân dạng 8 bit, chương trình sẽ sử dụng thuật toán nhân nhị phân để tính và xuất ra màn hình kết quả dưới dạng nhị phân 16 bit (theo thuật toán trong slide kết quả sẽ là [A, Q].  - Kết quả là 16 bit nhưng vẫn sẽ có thêm phần thông báo OVERFLOW cho kết quả nếu lưu dạng 8 bit. (Vd nếu 11 \* 12 thì kết quả vẫn xuất dưới dạng 16 bit, nhưng sẽ hiển thị thêm là nếu lưu dưới dạng 8 bit thì sẽ bị tràn số). |
| Tính thương hai dãy bit | 100% | - Input là 2 số nhị phân dạng 8 bit, chương trình sẽ sử dụng thuật toán chia nhị phân để tính và xuất ra màn hình kết quả bao gồm phần thương số và phần dư của phép chia.  - Có check dấu cho số dư và thương ở kết quả theo nguyên tắc (số dư trùng dấu với số bị chia) |

***- Đánh giá tổng thể bài nộp: Em đã hoàn thành tất cả các yêu cầu bao gồm: Bài 1 (5 yêu cầu) và Bài 2 (4 yêu cầu). Trong bài tập 2 có check OVERFLOW cho phép cộng, trừ và phép nhân. Có check dấu số dư và thương khi chia 2 số nhị phân. Về tính hiệu quả của code thì có một số hàm phát sinh do em chưa bao quát được vấn đề từ đầu dẫn đến code ở bài tập 2 bị dài và có một số phần bị lặp lại. Sau khi hoàn thiện bài làm thì em cũng đã tạo một bộ test case để test các kết quả có thể xảy ra và đánh giá bộ test là khá hoàn thiện.***

**I) Phần 2: Kết quả bài làm**

**a. Bài 1) Sử dụng file code: Homework1/exercise1.cpp**

**a.1. Từ hệ 10 sang hệ 2:**

- Hàm sử dụng trong câu này là hàm Dec\_to\_Bin:

Text

Description automatically generated

- Test case ở câu này sẽ là các số dương: 77, 12, 122, 0 (Số 0 không là số dương cũng không là số âm, nhưng trong code có xử lý để xuất ra kết quả).

- Ta tiến hành gọi hàm và đưa inputs vào:

Text

Description automatically generated

- Kết quả trả ra tương ứng sẽ là:

Graphical user interface, text, application

Description automatically generated with medium confidence

**a.2. Từ hệ 10 sang hệ 16:**

- Hàm được sử dụng trong câu này là hàm Dec\_to\_Hex:

Text

Description automatically generated

- Test case ở câu này sẽ là các số dương: 166, 122, 222, 57.

- Ta tiến hành gọi hàm và đưa inputs vào:

Text

Description automatically generated

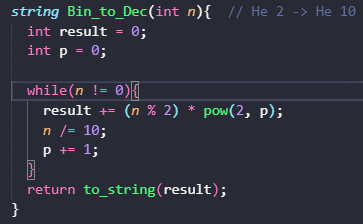
- Kết quả trả ra tương ứng sẽ là:

Text

Description automatically generated

**a.3. Từ hệ 2 sang hệ 10:**

- Hàm sử dụng ở câu này là hàm Bin\_to\_Dec:



- Test case ở câu này sẽ là các số dương nhị phân dưới dạng chuỗi: 1001101, 1100, 1111010, 0, 11111111 (Do chỉ xử lý các số nguyên dương, nên số 11111111 cũng sẽ đổi thành số hệ 10 dương)

Lưu ý: Khi đưa input vào, các số nhị phân cần bỏ hết những số 0 không cần thiết ở đầu, do đầu vào input là integer. Ví dụ: 0010 thì chỉ nhập 10.

- Ta tiến hành gọi hàm và đưa inputs vào:

Text, chat or text message

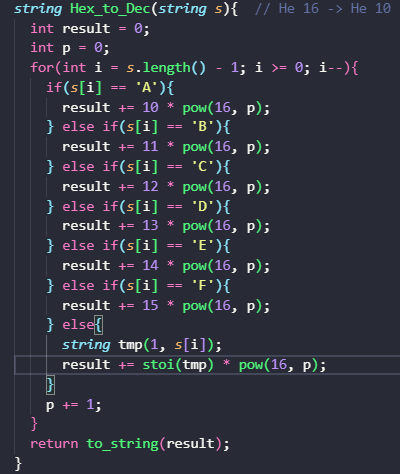
Description automatically generated

- Kết quả trả ra tương ứng sẽ là:

Text

Description automatically generated

**a.4. Từ hệ 16 sang hệ 10:**

- Hàm sử dụng ở câu này là hàm Hex\_to\_Dec:

- Test case ở câu này sẽ là các số dương ở hệ 16 dưới dạng chuỗi: A6, BB, 157, 153, 9F.

- Ta tiến hành gọi hàm và đưa inputs vào:

Text

Description automatically generated

- Kết quả trả ra sẽ là:

Text

Description automatically generated

**a.5. Từ hệ 2 sang hệ 16:**

- Hàm sử dụng ở câu này là hàm Bin\_to\_Hex:

Text

Description automatically generated

- Test case ở câu này là: 10100110, 1, 111111, 110001.

Lưu ý: Khi đưa input vào, các số nhị phân cần bỏ hết những số 0 không cần thiết ở đầu, do đầu vào input là integer. Ví dụ: 0010 thì chỉ nhập 10.

- Ta tiến hành gọi hàm và đưa inputs vào:

A screenshot of a computer

Description automatically generated with low confidence

- Kết quả trả ra sẽ là:

Text

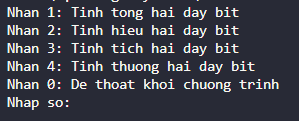
Description automatically generated

**b. Bài 2) Sử dụng file code: Homework1/exercise2.cpp**

**\*\* Cách sử dụng menu:**

Khi chạy chương trình sẽ có menu hiện ra cho người dùng chọn. Người dùng sẽ chọn các chức năng từ 1 đến 4 (ứng với phép cộng, trừ, nhân, chia) rồi mới tiến hành nhập đầu vào. Để thoát khỏi chương trình người dùng nhấn số 0.

Màn hình hiển thị của menu:



Nhưng trong phần bên dưới, để tiện lợi hơn thì ta sẽ không dùng menu mà GỌI THẲNG HÀM để thuận tiện trong việc so sánh đối chiếu kết quả.

**b.1. Tính tổng hai dãy bit:**

- Hàm sử dụng ở câu này là hàm add, với inputs đầu vào là 2 chuỗi số ở dạng nhị phân 8 bits (số có dấu ở dạng bù 2).

- Test case ở câu này là:

11111001 + 00000101 tức -7 + 5

11111100 + 00000100 tức -4 + 4

00000011 + 00000100 tức 3 + 4

11111100 + 11111111 tức -4 + (-1)

00000101 + 00000100 tức 5 + 4

11111001 + 11111010 tức -7 + (-6)

01111111 + 01111111 tức 127 + 127 (OVERFLOW)

- Ta sẽ tiến hành gọi hàm và nhập inputs:

Text

Description automatically generated

- Kết quả trả ra sẽ là:

Text

Description automatically generated

Như ta có thể thấy output của 2 dòng cuối cùng: 01111111 + 01111111 (127 + 127) là OVERFLOW.

**b.2. Tính hiệu hai dãy bit:**

- Hàm sử dụng ở câu này là hàm subtract, , với inputs đầu vào là 2 chuỗi số ở dạng nhị phân 8 bits (số có dấu ở dạng bù 2).

- Test case ở câu này là:

00000010 – 00000111 tức 2 - 7

00000101 – 00000010 tức 5 - 2

00000101 – 11111110 tức 5 - (-2)

11111010 – 11111100 tức -6 - (-4)

11111010 – 00000100 tức -6 - 4

01111111 – 10000001 tức 127 – (-127) OVERFLOW

- Ta tiến hành gọi hàm và nhập inputs:

Text, chat or text message

Description automatically generated

- Kết quả trả ra sẽ là:

Text

Description automatically generated

**b.3. Tính tích hai dãy bit:**

- Hàm sử dụng ở câu này là hàm multiply, , với inputs đầu vào là 2 chuỗi số ở dạng nhị phân 8 bits (số có dấu ở dạng bù 2).

- Test case ở câu này là:

00000101 \* 00000110 tức 5\*6

00000111 \* 11111101 tức 7\*(-3)

11111100 \* 11111010 tức -4 \* (-6)

11111011 \* 00000010 tức -5 \* 2

00001011 \* 00001100 tức 11\* 12

01111011 \* 01111110 tức 123 \* 126 (Kết quả vẫn sẽ ra dưới dạng 16 bits nhưng sẽ có đính kèm thông báo OVERFLOW)

- Ta tiến hành gọi hàm và nhập inputs:

Text

Description automatically generated

- Kết quả trả ra sẽ là:

Graphical user interface

Description automatically generated with medium confidence

**b.4. Tính thương hai dãy bit:**

- Hàm sử dụng ở câu này là hàm multiply, , với inputs đầu vào là 2 chuỗi số ở dạng nhị phân 8 bits (số có dấu ở dạng bù 2).

- Test case ở câu này là:

00000111 / 00000011 tức 7 / 3

00000111 / 11111101 tức 7 / (-3)

11111001 / 00000011 tức -7 / 3

11111001 / 11111101 tức -7 / (-3)

00001000 / 00000100 tức 8 / 4

11111101 / 00000101 tức -3 / 5

- Ta sẽ tiến hành gọi hàm và nhập inputs:

Text, chat or text message

Description automatically generated

- Kết quả trả về sẽ là:

A picture containing text, plaque

Description automatically generated