

**ĐẠI HỌC QUỐC GIA TP HCM**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**NHÓM NGÀNH MÁY TÍNH VÀ CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN 1**  
**BIỂU DIỄN VÀ TÍNH TOÁN SỐ NGUYÊN LỚN**

Bộ môn : Kiến trúc máy tính và hợp ngữ

Giảng viên : Chung Thùy Linh

Lớp : 18CTT5

Sinh viên :

Lê Ngọc Bảo Ngân 18120214

Ngô Trung Tín 18120596

Nguyễn Phan Công Vương 18120654

**Thành phố Hồ Chí Minh, tháng 06 năm 2020**

## MỤC LỤC

I.	Mô tả đồ án .....	3
II.	Bảng phân công công việc.....	3
III.	Môi trường lập trình.....	3
IV.	Ý tưởng thiết kế .....	4
V.	Kiểm tra các chức năng của chương trình.....	5
VI.	Tiến độ công việc.....	9
6.1.	Đã làm được .....	9
6.2.	Chưa làm được .....	9
VII.	Mức độ hoàn thành công việc .....	9
VIII.	Các nguồn tham khảo.....	9

## I. Mô tả đề án

Mô tả kiểu dữ liệu số nguyên lớn có dấu gọi là QInt có độ lớn 16 byte gồm một số chức năng:

- (a) Chuyển đổi số QInt từ hệ thập phân sang hệ nhị phân (dạng bù 2) và ngược lại
- (b) Chuyển đổi số QInt từ hệ nhị phân (dạng bù 2) sang hệ thập lục phân và ngược lại
- (c) Chuyển đổi số QInt từ hệ thập lục phân sang hệ thập phân và ngược lại
- (d) Các operator =, operator +, operator -, operator \*, operator / trên các hệ cơ số
- (e) Các toán tử AND “&”, OR “|”, XOR “^”, NOT “~”
- (f) Các toán tử dịch trái “<<”, dịch phải “>>” số học
- (g) Các phép xoay trái “rol”, xoay phải “ror” mỗi lần xoay chỉ xử lý cho đúng 1 bit, không cần xử lý cho trường hợp tổng quát xoay k bit

## II. Bảng phân công công việc

MSSV	Họ tên	Phân công
18120214	Lê Ngọc Bảo Ngân	- Chức năng (a), (b), (c) - Viết báo cáo
18120596	Ngô Trung Tín	- Chức năng (d), (e) - Viết báo cáo
18120654	Nguyễn Phan Công Vương	- Chức năng (f), (g) - Viết báo cáo

## III. Môi trường lập trình

Visual Studio 2019 – ngôn ngữ C++.

#### IV. Ý tưởng thiết kế

Dùng mảng tĩnh unsigned char 16 bytes để lưu trữ 128 bits của 1 số nguyên lớn 16 bytes theo yêu cầu của đồ án.

Phạm vi biểu diễn của kiểu dữ liệu đã thiết kế:  $-2^{64}$  đến  $2^{64} - 1$

Các chức năng:

- **Chuyển từ hệ 2 sang hệ 10:** Kiểm tra nếu số nhị phân là số âm thì thực hiện phép đảo bit; sau đó cho chạy vòng lặp cộng dồn chuỗi với chuỗi cộng  $2^i$  (với  $i$  là vị trí của bit đang xét); nếu số nhị phân là số âm, thêm dấu trừ vào trước kết quả).
- **Chuyển từ hệ 10 sang hệ 2:** Chia dần số thập phân cho 2 cho đến khi kết quả bằng 0, gán các số dư sau mỗi lần chia 2 và cộng dồn vào 1 chuỗi, sau đó đảo ngược chuỗi; nếu số thập phân là số âm, thực hiện phép đảo bit và + 1 (để trở thành bù 2).
- **Chuyển từ hệ 2 sang hệ 16:** Thực hiện vòng lặp xét mỗi 4 bit trong số nhị phân rồi cộng dồn ký tự hexa vào chuỗi kết quả.
- **Chuyển từ hệ 16 sang hệ 2:** Thực hiện vòng lặp tìm vị trí ký tự thứ  $i$  của số thập lục phân (dạng chuỗi) trong chuỗi HEX\_CHARS và thêm 4 bit tương ứng vào chuỗi kết quả.
- **Chuyển từ hệ 10 sang hệ 16:** Chuyển từ hệ 10 sang hệ 2 rồi chuyển từ hệ 2 sang hệ 16.
- **Chuyển từ hệ 16 sang hệ 10:** Chuyển từ hệ 16 sang hệ 2 rồi chuyển từ hệ 2 sang hệ 10.
- **Phép cộng “+”:** Thực hiện trên dãy nhị phân, cộng dãy bit từ trái sang phải, sử dụng 1 biến hàm bool để lưu số dư. Kiểm tra tràn số sau khi thực thi.
- **Phép trừ “-”:** Thực hiện trên dãy nhị phân, đổi dấu của số trừ rồi thực hiện phép cộng với số bị trừ. Kiểm tra tràn số sau khi thực thi.
- **Phép nhân “\*”:** Thực hiện trên dãy nhị phân, kiểm tra 2 giá trị đầu vào, thông báo tràn số nếu không thỏa mãn. Sử dụng thuật toán Booth để thực hiện phép nhân.
- **Phép chia “/”:** Thực hiện trên dãy nhị phân, kiểm tra giá trị đầu vào của số chia và số bị chia, kiểm tra trường hợp tràn số của phép chia. Sử dụng thuật toán Restoring Division Algorithm để thực hiện phép chia.
- **Phép gán “=”:** Thực hiện trên dãy nhị phân, thực hiện phép gán giá trị.
- **Phép AND “&”:** Thực hiện trên dãy nhị phân, với mỗi vị trí giống nhau của 2 dãy, thực hiện phép AND 2 bit.
- **Phép OR “|”:** Thực hiện trên dãy nhị phân, với mỗi vị trí giống nhau của 2 dãy, thực hiện phép OR 2 bit.

- ## V. Kiểm tra các chức năng của chương trình

- 
- The screenshot shows the Microsoft Visual Studio IDE with the following components:
- Menu Bar:** File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, Help.
  - Toolbar:** Standard IDE tools for opening files, saving, undo, redo, and running the program.
  - Source Explorer:** Located on the left, it shows the project structure for 'TEST\_PROJECT'.
  - Code Editor:** The main window displays 'Main.cpp' with line numbers 192 to 217. The code prompts the user for input numbers and displays binary representations of those numbers.
  - Output Window:** Located at the bottom, it shows the program's execution output.
  - Solution Explorer:** Located on the right, it shows the project structure for 'TEST\_PROJECT'.
- The code in 'Main.cpp' is as follows:
- ```

192 _ Input number (Enter: "0 0 0" to stop): 2 10 111111111111111101010101110110010100010101011111101100001111
193 1180585621117413619471
194
195 _ Input number (Enter: "0 0 0" to stop): 10 2 234528342373423623940
196
197 11001010101011100000011001100111000010001101101011100000100
198
199 _ Input number (Enter: "0 0 0" to stop): 2 16 1001010101010101010101000000111011000110001101
200
201 12AAAAA07B18D
202
203 _ Input number (Enter: "0 0 0" to stop): 16 2 3BAF3694EB448754DC475488E
204
205 111000101011100110110100110011101011000010000011010101001101110001000111010100100010001110
206
207 _ Input number (Enter: "0 0 0" to stop): 10 16 3284902735347623453482343
208
209 2B79AEF21661021375067
210
211 _ Input number (Enter: "0 0 0" to stop): 16 10 347AC53BA89DE4F57C3D43746348934
212
213 4359832272430921644623523111245220148
214
215 _ Input number (Enter: "0 0 0" to stop):
216
217

```
- The 'Output' window shows the following text:
- ```

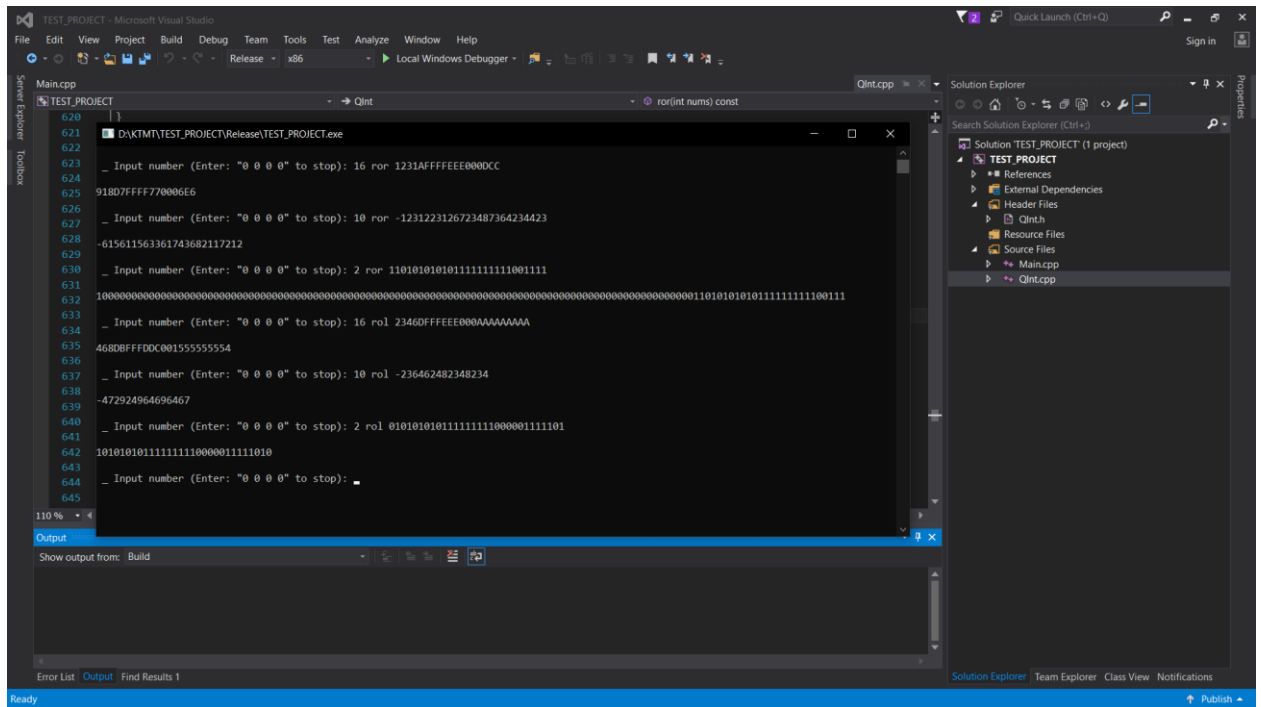
Show output from: Build

```
- The 'Solution Explorer' on the right shows the project structure for 'TEST\_PROJECT'.

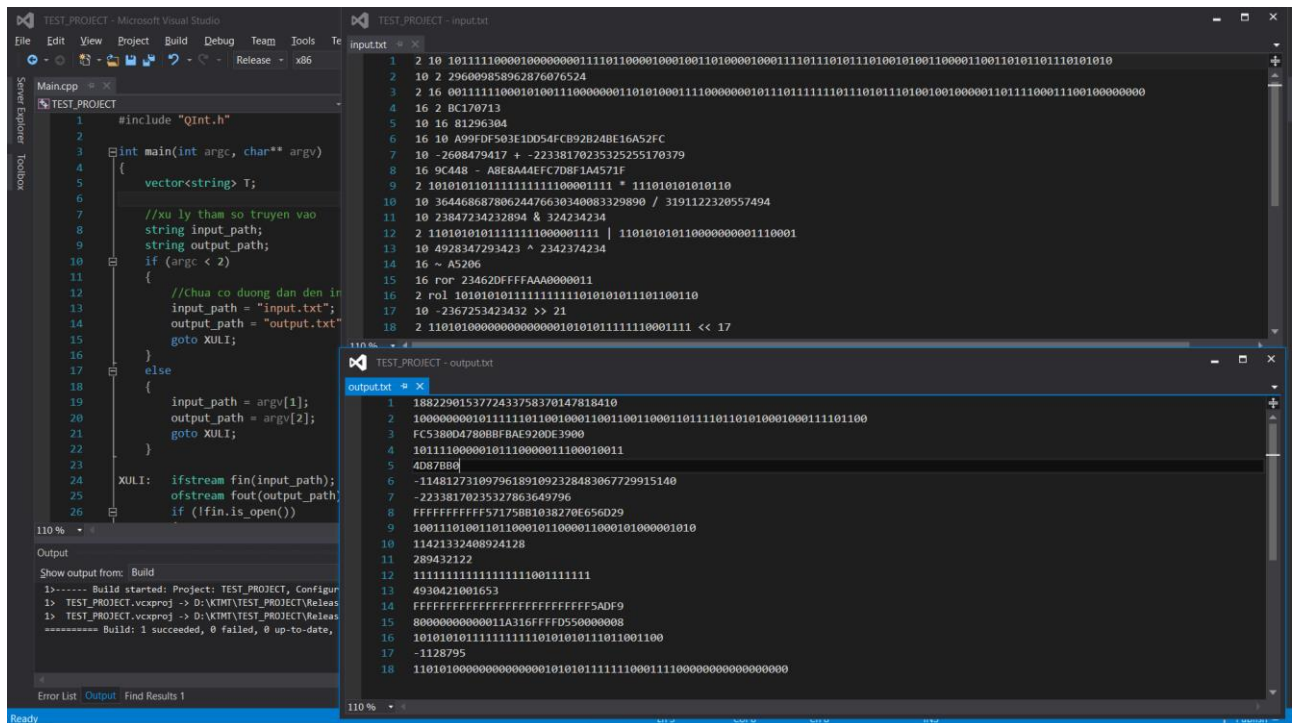




- **Các phép xoay trái, xoay phải:**



- **Xử lý nhập, xuất file:**





## VI. Tiến độ công việc

### 6.1. Đã làm được:

- Chuyển đổi hệ nhị phân (dạng bù 2) sang hệ thập phân và ngược lại.
- Chuyển đổi hệ nhị phân (dạng bù 2) sang hệ thập lục phân và ngược lại.
- Chuyển đổi hệ thập phân sang hệ thập lục phân và ngược lại.
- Các operator =, operator +, operator -, operator \* ,operator / trên các hệ cơ số.
- Các phép toán AND “&”, OR “|”, XOR “^”, NOT “~”.
- Các toán tử: Dịch trái “<<”, dịch phải số học “>>”.
- Các phép xoay trái “rol”, xoay phải “ror” ( đã xử lý cho trường hợp k bit).

### 6.2. Chưa làm được:

Không có.

## VII. Mức độ hoàn thành công việc

Đã hoàn thành 100%.

## VIII. Các nguồn tham khảo

- Slide Bài Giảng:

<https://drive.google.com/drive/folders/0B3Lo3u6fkAuoUS1JVHJHVG95d2M>

- Phép toán thao tác bit:

[https://vi.wikipedia.org/wiki/Ph%C3%A9p\\_to%C3%A1n\\_thao\\_t%C3%A1c\\_bit](https://vi.wikipedia.org/wiki/Ph%C3%A9p_to%C3%A1n_thao_t%C3%A1c_bit)