



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN 1

MÔN KTMT&HN

PHAN NHẬT VINH – 1712914

LÊ VĂN VŨ – 1712919

PHẠM THỊ TUYẾT VY – 1712927

ĐỒ ÁN KIẾN TRÚC MÁY TÍNH & HỢP NGỮ 1

SỐ NGUYÊN LỚN QINT

I. Thông tin thành viên nhóm, bảng phân công công việc

STT	Họ Tên	MSSV	Phân công công việc
1	PHAN NHẬT VINH	1712914	<ul style="list-style-type: none">• Phương thức khởi tạo QInt cho tất cả các hệ cơ số.• Chuyển đổi cơ số hệ 2 sang 10, 10 sang 2.• Toán tử =, /• Toán tử AND, OR, XOR, NOT
2	LÊ VĂN VŨ	1712919	<ul style="list-style-type: none">• Chuyển đổi cơ số 16 sang nhị phân và ngược lại.• Toán tử +, -, *• Các hàm hỗ trợ cho tính toán.
3	PHẠM THỊ TUYẾT VY	1712927	<ul style="list-style-type: none">• Chuyển đổi hệ 16 sang hệ 10 và ngược lại.• Toán tử <<, >> và phép quay ror và rol.• Viết báo cáo.

II. Quá trình thực hiện đồ án:

- Môi trường lập trình: Visual Studio 2013 Express
- Ý tưởng thiết kế đồ án và phạm vi biểu diễn:
 - Để tạo nên số nguyên QInt 128bit, nhóm em chọn cách sử dụng mảng 4 phần tử int, mỗi phần tử 32bit. Class chứa phương thức getBitString() để lấy chuỗi nhị phân 128bit của số, làm nền tảng cho các phép toán ở dưới.

- Các thao tác chuyển đổi cơ số đều dựa trên chuỗi nhị phân của số QInt, với sự hỗ trợ của các hàm chuyển nhị phân sang thập phân và ngược lại, nhị phân sang thập lục phân và ngược lại.
- Các toán tử <<, >>, ror, rol đều được thực hiện dựa trên chuỗi nhị phân của số QInt. Riêng các phép toán +, -, *, / được thực hiện trung gian qua hệ thập phân rồi chuyển về hệ nhị phân.
- Các toán tử &, |, ^, ~ được thực hiện bằng cách tận dụng thao tác có sẵn trên bit của ngôn ngữ C++ (thực hiện thao tác đó trên từng phần tử int của mảng dữ liệu).
- Phạm vi biểu diễn: số nguyên có dấu 128bit, độ lớn từ -2^{127} đến $2^{127}-1$.
- Chạy kiểm thử:
 - Chuyển đổi cơ số và kết quả:

```

205         output << "ln";
206     }
207 }
208 Input::close();
209 output.close();
210 system("cls");
211 cout << "Operation Finished!\n";
212 //}
213
214
215 QInt nhậpIn, "100"; //Khởi tạo biến n1 kiểu QInt theo hệ cơ số 10, nhận giá trị 100
216 QInt n2("10"); //Khởi tạo biến n2 kiểu QInt theo hệ cơ số 2, nhận giá trị 10, tức là số 2 trong thập phân
217 QInt n3(16, "F"); //Khởi tạo biến n3 kiểu QInt theo hệ cơ số 16, nhận giá trị F, tức là số 15 trong thập phân
218
219 //Các bước chuyển đổi cơ số:
220
221 //Thập phân sang nhị phân và thập lục phân
222 cout << "Binary: " + n1.toBinary() << endl;
223 cout << "Hexa: " + n1.toHex() << endl;
224
225 //Nhị phân sang thập phân và thập lục phân
226 cout << "Decimal: " + n2.toDecimal() << endl;
227 cout << "Hexa: " + n2.toHex() << endl;
228
229 //Thập lục phân sang nhị phân và thập phân
230 cout << "Binary: " + n3.toBinary() << endl;
231 cout << "Decimal: " + n3.toDecimal() << endl;
232 }
  
```

```
C:\WINDOWS\system32\cmd.exe
Binary: 1100100
Hexa: 64
Decimal: 2
Hexa: 2
Binary: 1111
Decimal: 15
Press any key to continue . . .
```

- Các toán tử =, +, -, *, /:

```
265     output << "\n";
266 }
267 }
268 input.close();
269 output.close();
270 system("cls");
271 cout << "Operation finished\n";
272 }
273 }
274
275 int n1(10, "100"); // Khai báo biến n1 kiểu int theo hệ cơ số 10, nhận giá trị 100
276 int n2(10, "10"); // Khai báo biến n2 kiểu int theo hệ cơ số 10, nhận giá trị 10
277
278 //Các toán tử +, -, *, /
279
280 //Kết quả xuất ở thập phân
281 cout << "n1 + n2 = " << (n1 + n2).toDecimal() << endl;
282 cout << "n1 - n2 = " << (n1 - n2).toDecimal() << endl;
283 cout << "n1 * n2 = " << (n1*n2).toDecimal() << endl;
284 cout << "n1 / n2 = " << (n1/n2).toDecimal() << endl;
285 cout << endl;
286 //Kết quả xuất ở nhị phân
287 cout << "n1 + n2 = " << (n1 + n2).toBinary() << endl;
288 cout << "n1 - n2 = " << (n1 - n2).toBinary() << endl;
289 cout << "n1 * n2 = " << (n1*n2).toBinary() << endl;
290 cout << "n1 / n2 = " << (n1 / n2).toBinary() << endl;
291 cout << endl;
292 //Kết quả xuất ở thập lục phân
293 cout << "n1 + n2 = " << (n1 + n2).toHex() << endl;
294 cout << "n1 - n2 = " << (n1 - n2).toHex() << endl;
295 cout << "n1 * n2 = " << (n1*n2).toHex() << endl;
296 cout << "n1 / n2 = " << (n1 / n2).toHex() << endl;
297 cout << endl;
298 //Viết từ gần
299 n1 = n2;
300 cout << "n1 in binary: " << n1.toBinary() << endl;
301 cout << "n1 in decimal: " << n1.toDecimal() << endl;
302 cout << "n1 in hexadecimal: " << n1.toHex() << endl;
303 cout << endl;
304 }
```

```
C:\WINDOWS\system32\cmd.exe
n1 + n2 = 110
n1 - n2 = 90
n1 * n2 = 1000
n1 / n2 = 10

n1 + n2 = 1101110
n1 - n2 = 1011010
n1 * n2 = 1111101000
n1 / n2 = 1010

n1 + n2 = 6E
n1 - n2 = 5A
n1 * n2 = 3E8
n1 / n2 = A

n1 in binary: 1010
n1 in decimal: 10
n1 in hexadecimal: A

Press any key to continue . . .
```

- Các phép toán &, |, ^, ~, <<, >>, rol và ror:

```
17214 170916 170917 - Microsoft Visual Studio Express 2010 for Windows Desktop
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Local Windows Debugger - Release - Win32
Main.cpp - Qnt.cpp - OutOfFunctions.cpp
[Global Scope] - main.cpp arg, che" arg(0)
211 | cout << "Operation finished!\n";
212 |
213 |
214 |
215 | Qnt n1(10, "100"); //Khởi tạo biến n1 kiểu Qnt theo hệ cơ số 10, nhập giá trị 100
216 | Qnt n2(10, "10"); //Khởi tạo biến n2 kiểu Qnt theo hệ cơ số 10, nhập giá trị 10
217 |
218 | //Các toán tử &, |, ^, ~, <<, >>, rol, ror
219 |
220 | //Kết quả xuất ở thập phân
221 | cout << "n1 & n2 = " << (n1 & n2).toDecimal() << endl;
222 | cout << "n1 | n2 = " << (n1 | n2).toDecimal() << endl;
223 | cout << "n1 ^ n2 = " << (n1 ^ n2).toDecimal() << endl;
224 | cout << "~n1 = " << (~n1).toDecimal() << endl;
225 | cout << "n1 << 3 = " << (n1 << 3).toDecimal() << endl;
226 | cout << "n2 >> 2 = " << (n2 >> 2).toDecimal() << endl;
227 | cout << "rol n1 = " << (n1.rol()).toDecimal() << endl;
228 | cout << "ror n2 = " << (n2.ror()).toDecimal() << endl;
229 | cout << endl;
230 | //Kết quả xuất ở nhị phân
231 | cout << "n1 & n2 = " << (n1 & n2).toBinary() << endl;
232 | cout << "n1 | n2 = " << (n1 | n2).toBinary() << endl;
233 | cout << "n1 ^ n2 = " << (n1 ^ n2).toBinary() << endl;
234 | cout << "~n1 = " << (~n1).toBinary() << endl;
235 | cout << "n1 << 3 = " << (n1 << 3).toBinary() << endl;
236 | cout << "n2 >> 2 = " << (n2 >> 2).toBinary() << endl;
237 | cout << "rol n1 = " << (n1.rol()).toBinary() << endl;
238 | cout << "ror n2 = " << (n2.ror()).toBinary() << endl;
239 | cout << endl;
240 | //Kết quả xuất ở thập lục phân
241 | cout << "n1 & n2 = " << (n1 & n2).toHex() << endl;
242 | cout << "n1 | n2 = " << (n1 | n2).toHex() << endl;
243 | cout << "n1 ^ n2 = " << (n1 ^ n2).toHex() << endl;
244 | cout << "~n1 = " << (~n1).toHex() << endl;
245 | cout << "n1 << 3 = " << (n1 << 3).toHex() << endl;
246 | cout << "n2 >> 2 = " << (n2 >> 2).toHex() << endl;
247 | cout << "rol n1 = " << (n1.rol()).toHex() << endl;
248 | cout << "ror n2 = " << (n2.ror()).toHex() << endl;
249 | cout << endl;
250 |
251 | }
```

