

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**  
**MÔN CẤU TRÚC RỜI RẠC**

**TÌM HIỂU**  
**CÁC PHÉP TOÁN TRÊN SỐ NHỊ PHÂN**

*Người hướng dẫn:* **THẦY DUNG CẨM QUANG**

*Người thực hiện:* **NGUYỄN THẾ TRƯỜNG – 51900780**

**Lớp : 19050302**

**Khoá : 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

## **LỜI CẢM ƠN**

Em xin chân thành cảm ơn thầy Dung Cẩm Quang đã hướng dẫn em cách làm bài báo cáo này. Sự hướng dẫn nhiệt tình của thầy và những kiến thức học được từ thầy đã giúp em hoàn thành bài báo cáo này một cách hoàn thiện nhất. Chúc thầy luôn có sức khỏe và đạt nhiều thành công hơn trong cuộc sống.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Dung Cẩm Quang. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Thế Trường*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm  
(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm  
(kí và ghi họ tên)

## TÓM TẮT

Bài báo cáo này giới thiệu về hệ nhị phân, công dụng của hệ nhị phân trong đời sống thực tiễn. Mô tả các hàm tính toán có trong bài tập lớn như cộng, trừ, nhân, AND, OR, XOR hai số nhị phân, NOT, dịch trái, dịch phải một số nhị phân, cách chuyển một số từ hệ nhị phân sang hệ thập lục phân. Giải thích các thuật toán sử dụng để thực hiện các hàm trên, thực hiện ví dụ cụ thể. Dựa vào những kiến thức thao tác với hệ nhị phân đã học ở môn Tổ chức máy tính, Cấu trúc rời rạc và sử dụng ngôn ngữ lập trình Python để thực hiện bài tập lớn này.

## MỤC LỤC

LỜI CẢM ƠN .....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	iii
TÓM TẮT .....	iv
MỤC LỤC .....	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ .....	3
CHƯƠNG 1 – GIỚI THIỆU HỆ NHỊ PHÂN.....	4
1.1 Định nghĩa và chức năng .....	4
1.1.1 Định nghĩa.....	4
1.1.2 Chức năng .....	4
1.2 Các hàm tính toán trong bài tập lớn .....	4
1.2.1 Hàm sum(A,B) .....	4
1.2.2 Hàm dif(A,B) .....	4
1.2.3 Hàm prod(A,B) .....	5
1.2.4 Hàm bitwiseAnd(A,B) .....	5
1.2.5 Hàm bitwiseOr(A,B).....	5
1.2.6 Hàm bitwiseXor(A,B).....	6
1.2.7 Hàm bitwiseNot(A).....	6
1.2.8 Hàm bitwiseLeftShift(A) .....	6
1.2.9 Hàm bitwiseRightShift(A) .....	6
1.2.10 Hàm bin2Hex(A) .....	6
CHƯƠNG 2 – MÔ TẢ THUẬT TOÁN.....	8
2.1 Lấy ví dụ tính toán .....	8
2.2 Mô tả các hàm trong bài.....	8
2.2.1 Hàm sum(A,B) .....	8
2.2.2 Hàm dif(A,B) .....	8
2.2.3 Hàm prod(A,B) .....	9

2.2.4 Hàm bitwiseAnd(A, B) .....	9
2.2.5 Hàm bitwiseOr(A,B) .....	10
2.2.6 Hàm bitwiseXor(A,B) .....	10
2.2.7 Hàm bitwiseNot(A) .....	11
2.2.8 Hàm bitwiseLeftShift(A) .....	11
2.2.9 Hàm bitwiseRightShift(A) .....	11
2.2.10 Hàm bin2Hex(A) .....	12
2.2.11 Các hàm phụ được sử dụng trong bài .....	12
2.2.11.1 Hàm removeLeadZero(A) .....	12
2.2.11.2 Hàm zfill(a) .....	12
2.2.11.3 Hàm int(s) .....	12
2.2.11.4 Hàm str(n) .....	12
2.2.11.5 Hàm chr(n) .....	12
CHƯƠNG 3 – KẾT QUẢ .....	13

## **DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

### **DANH MỤC BẢNG BIỂU**

Bảng 1.1: Bảng chân trị phép AND .....	5
Bảng 1.2: Bảng chân trị phép OR .....	5
Bảng 1.3: Bảng chân trị phép XOR .....	6
Bảng 1.4: Bảng chân trị phép NOT .....	6
Bảng 1.5: Bảng quy tắc chuyển đổi hệ nhị phân sang hệ thập lục phân .....	7

### **DANH MỤC HÌNH VẼ**

Hình 3.1: Kết quả thực hiện các hàm .....	13
---	----

### **DANH MỤC ĐỒ THỊ**



## CHƯƠNG 1 – GIỚI THIỆU HỆ NHỊ PHÂN

### 1.1 Định nghĩa và chức năng

#### 1.1.1 Định nghĩa

Hệ nhị phân (hay hệ đếm cơ số hai hoặc mã nhị phân) là một hệ đếm dùng hai ký tự để biểu đạt một giá trị số, bằng tổng số các lũy thừa của 2. Hai ký tự đó thường là 0 và 1; chúng thường được dùng để biểu đạt hai giá trị hiệu điện thế tương ứng (có hiệu điện thế, hoặc hiệu điện thế cao là 1 và không có, hoặc thấp là 0).

#### 1.1.2 Chức năng

Do có ưu điểm tính toán đơn giản, dễ dàng thực hiện về mặt vật lý, chẳng hạn như trên các mạch điện tử, hệ nhị phân trở thành một phần kiến tạo căn bản trong các máy tính đương thời.

### 1.2 Các hàm tính toán trong bài tập lớn

#### 1.2.1 Hàm $sum(A,B)$

- Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả của phép cộng A và B.
- Quy tắc cộng 2 số nhị phân:
  - $0 + 0 = 0$
  - $0 + 1 = 1$
  - $1 + 0 = 1$
  - $1 + 1 = 0$  (Nhớ 1)

#### 1.2.2 Hàm $dif(A,B)$

- Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả của phép trừ A cho B.
- Quy tắc trừ 2 số nhị phân:
  - $0 - 0 = 0$
  - $0 - 1 = 1$  (Mượn 1)

- $1 - 0 = 1$
- $1 - 1 = 0$

### 1.2.3 Hàm *prod(A,B)*

Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả của phép nhân A và B.

### 1.2.4 Hàm *bitwiseAnd(A,B)*

Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép AND của A và B.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Bảng 1.1: Bảng chân trị phép AND

### 1.2.5 Hàm *bitwiseOr(A,B)*

Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép OR của A và B.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Bảng 1.2: Bảng chân trị phép OR

### 1.2.6 Hàm *bitwiseXor(A,B)*

Đầu vào là 2 chuỗi A, B được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép XOR của A và B.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 1.3: Bảng chân trị phép XOR

### 1.2.7 Hàm *bitwiseNot(A)*

Đầu vào là chuỗi A được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép NOT của A.

A	NOT A
0	1
1	0

Bảng 1.4: Bảng chân trị phép NOT

### 1.2.8 Hàm *bitwiseLeftShift(A)*

Đầu vào là chuỗi A được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép Left Shift của A.

### 1.2.9 Hàm *bitwiseRightShift(A)*

Đầu vào là chuỗi A được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả phép Right Shift của A.

### 1.2.10 Hàm *bin2Hex(A)*

Đầu vào là chuỗi A được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi biểu diễn kết quả của phép chuyển A từ hệ nhị phân sang hệ thập lục phân.

Hệ nhị phân	Hệ thập lục phân
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Bảng 1.5: Bảng quy tắc chuyển đổi hệ nhị phân sang hệ thập lục phân

## CHƯƠNG 2 – MÔ TẢ THUẬT TOÁN

### 2.1 Lấy ví dụ tính toán

MSSV: 51900780

Lấy chuỗi A bằng cách lấy từng ký tự trong MSSV chia lấy dư cho 2.

$\Rightarrow A = "11100100"$

Lấy chuỗi B bằng cách:  $B = A + A$ , sau đó bỏ kí tự đầu tiên bên trái ra khỏi chuỗi.

$\Rightarrow B = "11001000"$

### 2.2 Mô tả các hàm trong bài

#### 2.2.1 Hàm *sum(A,B)*

$A = "11100100", B = "11001000"$

Các bước thực hiện:

- $0 + 0 = 0$
- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $0 + 0 = 0$
- $1 + 0 = 1$
- $1 + 1 = 0$  (Nhớ 1)
- $1 + 1 = 0$  (Cộng thêm 1 nữa) = 1 (Nhớ 1)
- $0 + 0 = 0$  (Cộng thêm 1 nữa) = 1

Vậy kết quả là "110101100"

#### 2.2.2 Hàm *dif(A,B)*

$A = "11100100", B = "11001000"$

Các bước thực hiện:

- $0 - 0 = 0$

- $0 - 0 = 0$
- $1 - 0 = 1$
- $0 - 1 = 1$  (Mượn 1)
- $0 - 0 = 0$  (Trừ thêm 1 nữa) = 1 (Mượn 1)
- $1 - 0 = 1$  (Trừ thêm 1 nữa) = 0
- $1 - 1 = 0$
- $1 - 1 = 0$

Vậy kết quả là "11100"

### 2.2.3 Hàm *prod(A,B)*

A = "11100100", B = "11001000"

Các bước thực hiện:

							×	1	1	1	0	0	1	0	0
								1	1	0	0	1	0	0	0
							+		0	0	0	0	0	0	0
						+		0	0	0	0	0	0	0	
					+		0	0	0	0	0	0	0	0	
				+		1	1	1	0	0	1	0	0		
			+		0	0	0	0	0	0	0	0			
		+		0	0	0	0	0	0	0	0				
	+		1	1	1	0	0	1	0	0					
+		1	1	1	0	0	1	0	0						
1	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0

Vậy kết quả là "1011001000100000"

### 2.2.4 Hàm *bitwiseAnd(A, B)*

A = "11100100", B = "11001000"

Các bước thực hiện:

- $0 \text{ AND } 0 = 0$
- $0 \text{ AND } 0 = 0$
- $1 \text{ AND } 0 = 0$
- $0 \text{ AND } 1 = 0$
- $0 \text{ AND } 0 = 0$
- $1 \text{ AND } 0 = 0$
- $1 \text{ AND } 1 = 1$
- $1 \text{ AND } 1 = 1$

Vậy kết quả là "11000000"

### 2.2.5 Hàm *bitwiseOr*(A,B)

A = "11100100", B = "11001000"

Các bước thực hiện:

- $0 \text{ OR } 0 = 0$
- $0 \text{ OR } 0 = 0$
- $1 \text{ OR } 0 = 1$
- $0 \text{ OR } 1 = 1$
- $0 \text{ OR } 0 = 0$
- $1 \text{ OR } 0 = 1$
- $1 \text{ OR } 1 = 1$
- $1 \text{ OR } 1 = 1$

Vậy kết quả là "11101100"

### 2.2.6 Hàm *bitwiseXor*(A,B)

A = "11100100", B = "11001000"

Các bước thực hiện:

- $0 \text{ XOR } 0 = 0$
- $0 \text{ XOR } 0 = 0$
- $1 \text{ XOR } 0 = 1$
- $0 \text{ XOR } 1 = 1$
- $0 \text{ XOR } 0 = 0$
- $1 \text{ XOR } 0 = 1$
- $1 \text{ XOR } 1 = 0$
- $1 \text{ XOR } 1 = 0$

Vậy kết quả là "101100"

### 2.2.7 Hàm *bitwiseNot(A)*

A = "11100100"

Các bước thực hiện:

- $\text{NOT } 0 = 1$
- $\text{NOT } 0 = 1$
- $\text{NOT } 1 = 0$
- $\text{NOT } 0 = 1$
- $\text{NOT } 0 = 1$
- $\text{NOT } 1 = 0$
- $\text{NOT } 1 = 0$
- $\text{NOT } 1 = 0$

Vậy kết quả là "11011"

### 2.2.8 Hàm *bitwiseLeftShift(A)*

A = "11100100"

Dịch tất cả các bit (trừ bit đầu tiên) sang trái, sau đó đưa bit đầu tiên ra cuối.

Vậy kết quả là "11001001"

### 2.2.9 Hàm *bitwiseRightShift(A)*



A = "11100100"

Dịch tất cả các bit (trừ bit cuối cùng) sang phải, sau đó đưa bit cuối cùng về đầu.

Vậy kết quả là "1110010"

### **2.2.10 Hàm *bin2Hex(A)***

A = "11100100"

Các bước thực hiện:

- 0100 → 4
- 1110 → E

Vậy kết quả là "E4"

### **2.2.11 Các hàm phụ được sử dụng trong bài**

#### **2.2.11.1 Hàm *removeLeadZero(A)***

Hàm này em tự định nghĩa. Đầu vào là chuỗi A được biểu diễn dưới dạng nhị phân. Đầu ra là 1 chuỗi đã loại bỏ những phần tử "0" dư thừa ở đầu chuỗi A.

#### **2.2.11.2 Hàm *zfill(a)***

Hàm này dùng để thêm các phần tử "0" vào bên trái chuỗi sao cho chuỗi có số phần tử đúng bằng a.

#### **2.2.11.3 Hàm *int(s)***

Hàm này dùng để chuyển chuỗi s sang số nguyên.

#### **2.2.11.4 Hàm *str(n)***

Hàm này dùng để chuyển số nguyên n sang dạng chuỗi.

#### **2.2.11.5 Hàm *chr(n)***

Hàm này dùng để chuyển đổi số nguyên n sang chữ cái theo mã ASCII.

## CHƯƠNG 3 – KẾT QUẢ

```

51900780.py X
51900780.py > bin2Hex
128
129 # Exercise 10
130 def bin2Hex(A):
131     dic = {}
132     tmp = "0"
133     for i in range(16):
134         if i < 10:
135             dic[tmp.zfill(4)] = str(i)
136         else:
137             dic[tmp.zfill(4)] = chr(i%10 + 65)
138     tmp = sum(tmp, "1")
139     maxlen = len(A) + (4 - len(A)%4)
140     A = A.zfill(maxlen)
141     result = ""
142     for i in range(maxlen-1, -4, -4):
143         temp = ""
144         for j in range(i, i-4, -1):
145             temp = A[j] + temp
146         for item in dic:
147             if item == temp:
148                 result = dic[item] + result
149     return removeLeadZero(result)
150
151 A = "11100100"
152 B = "11001000"
153 print(" A = " + A)
154 print(" B = " + B)
155 print(" A + B = " + sum(A, B))
156 print(" A - B = " + dif(A, B))
157 print(" A * B = " + prod(A, B))
158 print(" A AND B = " + bitwiseAnd(A, B))
159 print(" A OR B = " + bitwiseOr(A, B))
160 print(" A XOR B = " + bitwiseXor(A, B))
161 print(" NOT A = " + bitwiseNot(A))
162 print(" LEFT SHIFT OF A: " + bitwiseLeftShift(A))
163 print(" RIGHT SHIFT OF A: " + bitwiseRightShift(A))
164 print(" CONVERT A TO HEX: " + bin2Hex(A))
  
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\Documents\TDU\Discrete Structures\Lab\Big Exercise> & "C:/U  
900780.py"  
A = 11100100  
B = 11001000  
A + B = 110101100  
A - B = 11100  
A \* B = 1011001000100000  
A AND B = 11000000  
A OR B = 11101100  
A XOR B = 101100  
NOT A = 11011  
LEFT SHIFT OF A: 11001001  
RIGHT SHIFT OF A: 1110010  
CONVERT A TO HEX: E4  
PS D:\Documents\TDU\Discrete Structures\Lab\Big Exercise>

Hình 3.1: Kết quả thực hiện các hàm