**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Môn: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Đề tài: Bài toán chuyển đổi cơ số từ hệ thập lục phân sang hệ nhị phân, bát phân, thập lục phân sử dụng Stack dạng mảng.**

**Đỗ Thị Loan- 2010A05 - 20a10010035  
Phạm Đình Tân-2110A04 - 21A100100330  
Phạm Quang Vinh - 2210A02 - 22A1001D0370  
Trần Đại Phong - 2210A02 - 22A1001D0249  
Đặng Hồng Vân - 2210A02 - 22A1001D0249**

**Giảng Viên Hướng Dẫn:**

**Nguyễn Thanh Tùng**

**BÁO CÁO BÀI TẬP LỚN**

***Hà Nội, 2023***

Phân Công Kế Hoạch Và Thực Hiện

|  |  |  |
| --- | --- | --- |
| Công việc | Thành viên thực hiện | Thời gian thực hiện |
| 1. Tìm CTDL |  |  |
| 1. Tìm thuật toán |  |  |
| 1. Cài đặt lập trình |  |  |
| 1. Kiểm thử và Test |  |  |

Mục Lục

[1 Mô tả bài toán 4](#_Toc152177170)

[2 Hướng giải quyết 4](#_Toc152177171)

[2.1 Chuyển đổi từ hệ thập phân sang nhị phân: 4](#_Toc152177172)

[2.2 Chuyển đổi từ hệ thập phân sang bát phân 4](#_Toc152177173)

[2.3 Chuyển đổi từ hệ thập phân sang thập lục phân 5](#_Toc152177174)

[3 Xác định thuật toán 6](#_Toc152177175)

[3.1 Hàm khởi tạo danh sách rỗng 6](#_Toc152177176)

[3.2 Hàm kiểm tra xem một ngăn xếp (Stack) có rỗng hay không 7](#_Toc152177177)

[3.3 Hàm kiểm tra xem một ngăn xếp (Stack) có đầy hay không 8](#_Toc152177178)

[3.4 Hàm thêm một phần tử vào đỉnh của ngăn xếp (Stack) 10](#_Toc152177179)

[3.5 Hàm  để loại bỏ một phần tử từ đỉnh của ngăn xếp (Stack) 11](#_Toc152177180)

[3.6 Hàm truy cập vào phần tử trên cùng của ngăn xếp (Stack) mà không loại bỏ nó 13](#_Toc152177181)

[3.7 Hàm để chuyển đổi một số từ hệ cơ số 10 sang một hệ cơ số khác 15](#_Toc152177182)

[3.8 Hàm hiển thị tất cả các phần tử trong ngăn xếp (Stack) từ trên xuống dưới. 17](#_Toc152177183)

[3.9 Nhập số cần chuyển đổi. 19](#_Toc152177184)

[4 Lập trình – Cài đặt 20](#_Toc152177185)

[5 Kiểm thử - Lập bộ Test 22](#_Toc152177186)

# Mô tả bài toán

Đề tài : Bài toán chuyển đổi cơ số từ hệ thập lục phân sang hệ nhị phân, bát phân, thập lục phân sử dụng Stack dạng mảng.

# Hướng giải quyết

Bài toán chuyển đổi cơ số là bài toán chuyển đổi một số từ hệ cơ số này sang hệ cơ số khác. Bài toán này có thể được giải quyết bằng nhiều cách khác nhau, tùy thuộc vào hệ cơ số cần chuyển đổi.

## Chuyển đổi từ hệ thập phân sang nhị phân:

Để chuyển đổi một số thập phân sang nhị phân, ta thực hiện theo các bước sau:

Chia số thập phân cho 2.

Lấy phần dư của phép chia làm số hạng hàng đơn vị của số nhị phân.

Lấy số nguyên của phép chia làm số hạng hàng chục của số nhị phân.

Lặp lại bước 2 và 3 cho đến khi số nguyên của phép chia bằng 0.

Ví dụ:

Chuyển đổi số 123 sang nhị phân:

123 / 2 = 61 dư 1

61 / 2 = 30 dư 1

30 / 2 = 15 dư 0

15 / 2 = 7 dư 1

7 / 2 = 3 dư 1

3 / 2 = 1 dư 1

1 / 2 = 0 dư 1

Vậy, 123 = 11110111

## Chuyển đổi từ hệ thập phân sang bát phân

Để chuyển đổi một số thập phân sang bát phân, ta thực hiện theo các bước sau:

Chia số thập phân cho 8.

Lấy phần dư của phép chia làm số hạng hàng đơn vị của số bát phân.

Lấy số nguyên của phép chia làm số hạng hàng tám của số bát phân.

Lặp lại bước 2 và 3 cho đến khi số nguyên của phép chia bằng 0.

Ví dụ:

Chuyển đổi số 123 sang bát phân:

123 / 8 = 15 dư 3

15 / 8 = 1 dư 7

1 / 8 = 0 dư 1

Vậy, 123 = 173

## Chuyển đổi từ hệ thập phân sang thập lục phân

Để chuyển đổi một số thập phân sang thập lục phân, ta thực hiện theo các bước sau:

Chia số thập phân cho 16.

Lấy phần dư của phép chia làm số hạng hàng đơn vị của số thập lục phân.

Lấy số nguyên của phép chia làm số hạng hàng mười sáu của số thập lục phân.

Lặp lại bước 2 và 3 cho đến khi số nguyên của phép chia bằng 0.

Tuy nhiên, trong hệ thập lục phân, chỉ có 16 chữ số, từ 0 đến 9 và từ A đến F. Do đó, khi số thập phân lớn hơn 15, ta cần sử dụng các chữ số từ A đến F.

Ví dụ:

Chuyển đổi số 123 sang thập lục phân:

123 / 16 = 7 dư 11

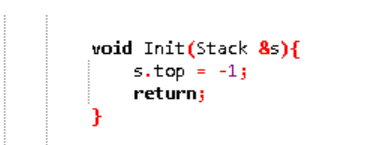
7 / 16 = 0 dư 7

Vậy, 123 = 7B

# Xác định thuật toán

## Hàm khởi tạo danh sách rỗng

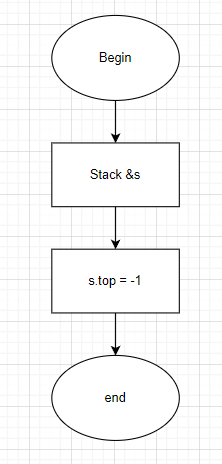
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

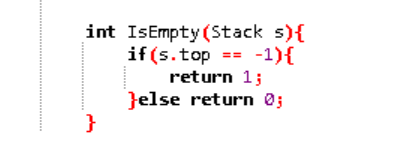
Hàm đặt giá trị của s.top là -1. -1 đại diện cho một ngăn xếp rỗng

### Sơ đồ thuật toán



## Hàm kiểm tra xem một ngăn xếp (Stack) có rỗng hay không

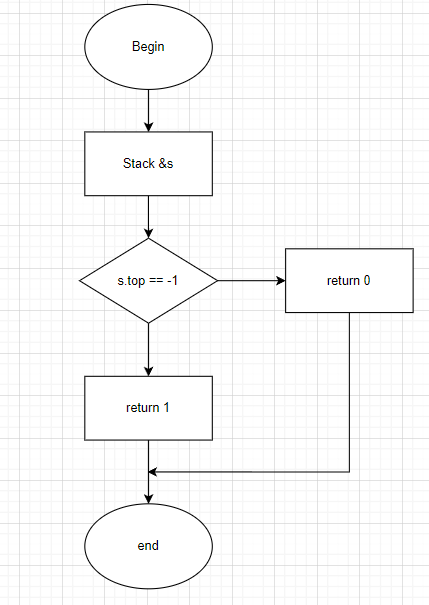
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

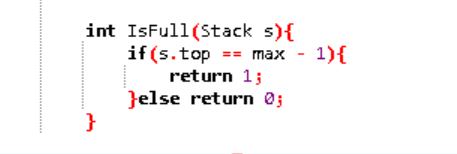
* Hàm kiểm tra xem giá trị của s.top có bằng -1 hay không. Trong trường hợp này, -1 đại diện cho một ngăn xếp rỗng (như đã được định nghĩa trong hàm Init mà bạn đã đề cập trước đó).
* Nếu s.top bằng -1, hàm trả về 1, đại diện cho việc ngăn xếp đang rỗng.
* Nếu s.top không bằng -1, hàm trả về 0, đại diện cho việc ngăn xếp không rỗng.

### Sơ đồ thuật toán



## Hàm kiểm tra xem một ngăn xếp (Stack) có đầy hay không

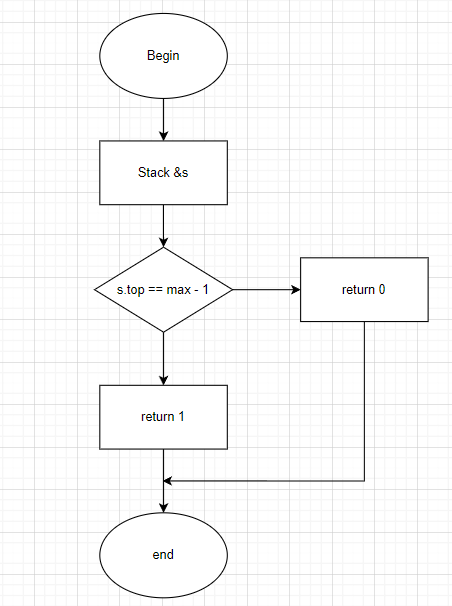
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

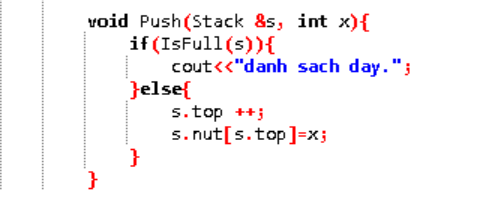
* Hàm kiểm tra xem giá trị của s.top có bằng max - 1 hay không. Trong trường hợp này, max là kích thước tối đa của ngăn xếp và max - 1 là chỉ số của phần tử cuối cùng trong ngăn xếp nếu nó đầy.
* Nếu s.top bằng max - 1, hàm trả về 1, đại diện cho việc ngăn xếp đang đầy.
* Nếu s.top không bằng max - 1, hàm trả về 0, đại diện cho việc ngăn xếp không đầy.

### Sơ đồ thuật toán



## Hàm thêm một phần tử vào đỉnh của ngăn xếp (Stack)

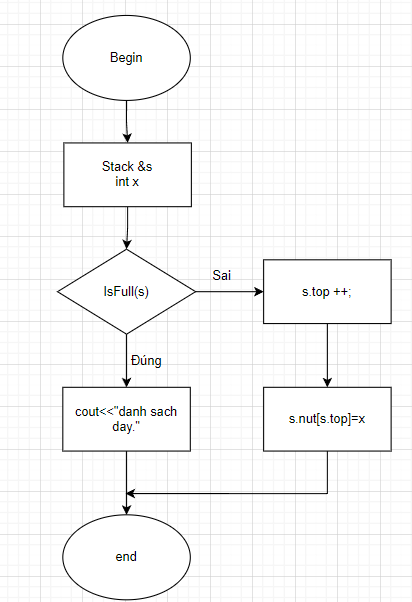
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

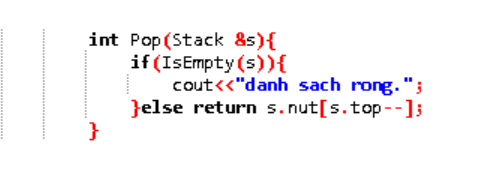
* Đầu tiên, hàm kiểm tra xem ngăn xếp có đầy hay không bằng cách gọi hàm IsFull(s). Nếu ngăn xếp đã đầy, nó sẽ in ra thông báo “danh sach day.” và không thực hiện thêm bất kỳ hành động nào.
* Nếu ngăn xếp chưa đầy, hàm sẽ tăng giá trị của s.top lên 1. Điều này đại diện cho việc tạo ra không gian cho một phần tử mới trên đỉnh của ngăn xếp.
* Cuối cùng, hàm gán giá trị x vào vị trí mới trên đỉnh ngăn xếp, được chỉ định bởi s.nut[s.top].

### Sơ đồ thuật toán



## Hàm  để loại bỏ một phần tử từ đỉnh của ngăn xếp (Stack)

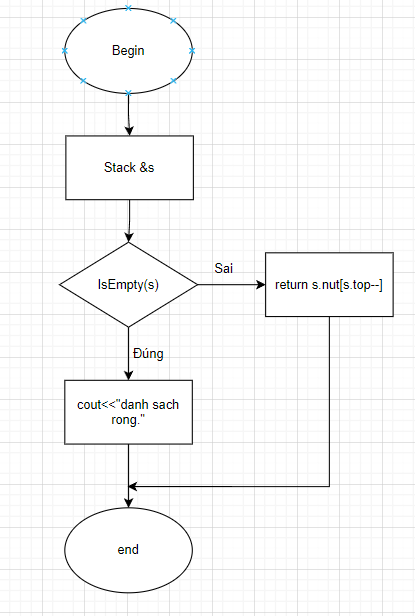
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

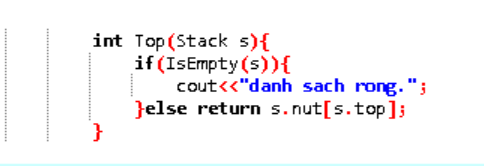
* Đầu tiên, hàm kiểm tra xem ngăn xếp có rỗng hay không bằng cách gọi hàm IsEmpty(s). Nếu ngăn xếp rỗng, nó sẽ in ra thông báo “danh sach rong.” và không thực hiện thêm bất kỳ hành động nào.
* Nếu ngăn xếp không rỗng, hàm sẽ trả về phần tử ở đỉnh ngăn xếp, được chỉ định bởi s.nut[s.top], và sau đó giảm giá trị của s.top đi 1 (được thể hiện bởi toán tử -- sau s.top). Điều này đại diện cho việc loại bỏ phần tử từ đỉnh ngăn xếp.

### Sơ đồ thuật toán



## Hàm truy cập vào phần tử trên cùng của ngăn xếp (Stack) mà không loại bỏ nó

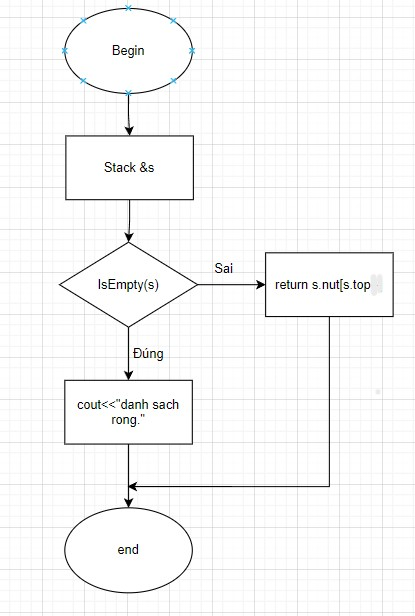
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

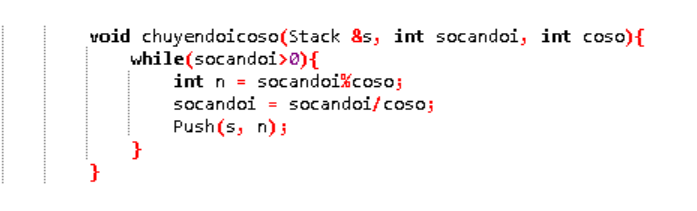
* Đầu tiên, hàm kiểm tra xem ngăn xếp có rỗng hay không bằng cách gọi hàm IsEmpty(s). Nếu ngăn xếp rỗng, nó sẽ in ra thông báo “danh sach rong.” và không thực hiện thêm bất kỳ hành động nào.
* Nếu ngăn xếp không rỗng, hàm sẽ trả về phần tử ở đỉnh ngăn xếp, được chỉ định bởi s.nut[s.top]. Điều này đại diện cho việc truy cập vào phần tử trên cùng của ngăn xếp mà không loại bỏ nó.

### Sơ đồ thuật toán



## Hàm để chuyển đổi một số từ hệ cơ số 10 sang một hệ cơ số khác

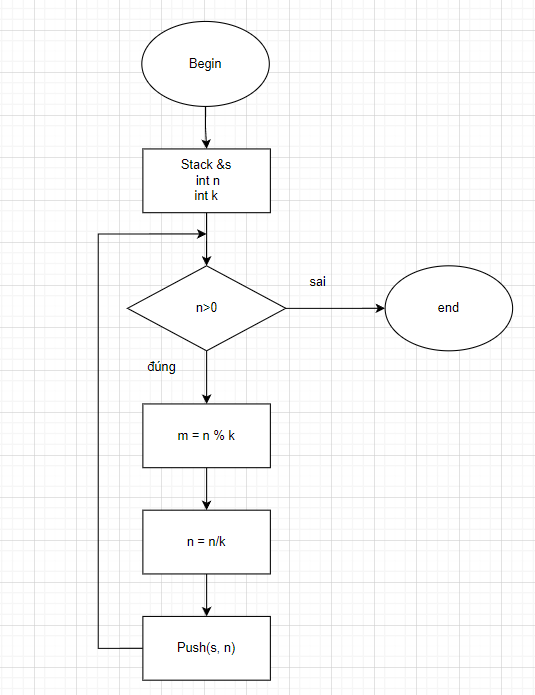
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

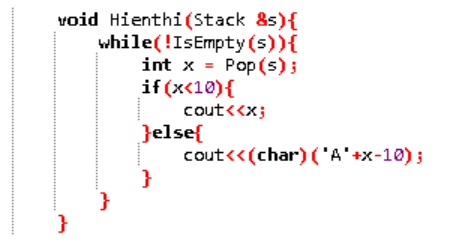
* Hàm bắt đầu với một vòng lặp while chạy miễn là socandoi (số cần chuyển đổi) lớn hơn 0.
* Trong mỗi lần lặp, hàm tính toán phần dư của socandoi khi chia cho coso (cơ số cần chuyển đổi sang), và gán kết quả vào biến n.
* Sau đó, socandoi được chia cho coso và kết quả được làm tròn xuống số nguyên gần nhất.
* Cuối cùng, n được đẩy vào ngăn xếp s bằng cách gọi hàm Push(s, n).

### Sơ đồ thuật toán



## Hàm hiển thị tất cả các phần tử trong ngăn xếp (Stack) từ trên xuống dưới.

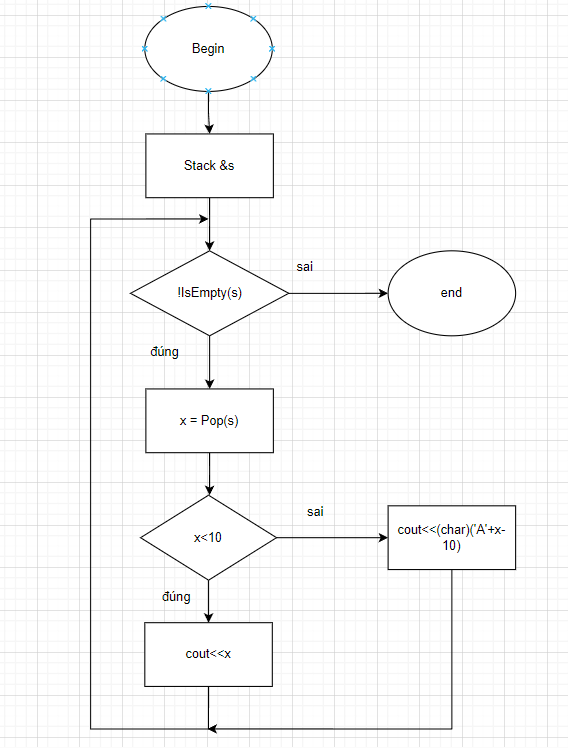
### Cấu trúc hàm



### Mô tả chi tiết về cách thức hoạt động của nó :

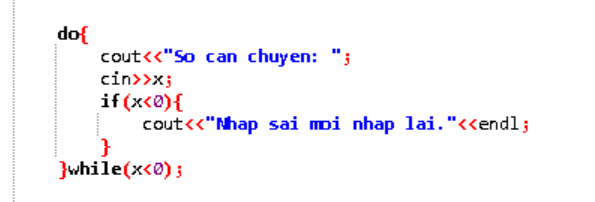
* Hàm bắt đầu với một vòng lặp while chạy miễn là ngăn xếp s không rỗng, được kiểm tra bằng hàm IsEmpty(s).
* Trong mỗi lần lặp, hàm gọi Pop(s) để loại bỏ và trả về phần tử trên cùng của ngăn xếp, và gán kết quả vào biến x.
* Nếu x nhỏ hơn 10, hàm in ra giá trị của x.
* Nếu x lớn hơn hoặc bằng 10, hàm chuyển đổi x thành một ký tự từ ‘A’ đến ‘F’ (đại diện cho các số từ 10 đến 15 trong hệ cơ số 16) và in ra ký tự đó. Điều này được thực hiện bằng cách cộng x với giá trị ASCII của ‘A’ và trừ đi 10, sau đó ép kiểu kết quả thành char.

### Sơ đồ thuật toán



## Nhập số cần chuyển đổi.

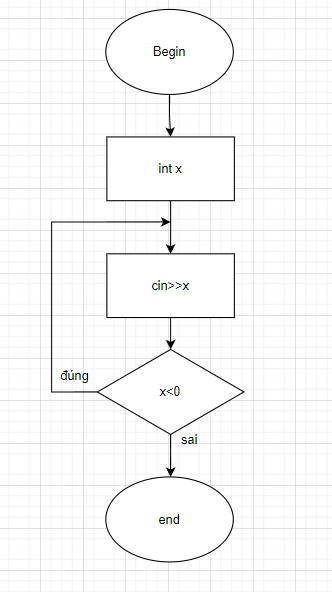
### Cấu trúc



### Mô tả chi tiết về cách thức hoạt động của nó :

* + Xuất thông báo "So can chuyen: " để yêu cầu người dùng nhập một số.
  + Nhận giá trị đầu vào từ người dùng và gán cho biến x.
  + Kiểm tra xem giá trị của x có nhỏ hơn 0 hay không.
  + Nếu x nhỏ hơn 0, xuất thông báo “Nhap sai moi nhap lai.” và yêu cầu người dùng nhập lại.
  + Nếu x không nhỏ hơn 0, kết thúc vòng lặp.
  + Quá trình này lặp lại cho đến khi người dùng nhập một số không nhỏ hơn 0.

### Sơ đồ thuật toán



# Lập trình – Cài đặt

Bài toán chuyển đổi cơ số từ hệ thập lục phân sang hệ nhị phân, bát phân, thạp lục phân sử dụng Stack dạng mảng.

#include<iostream>

using namespace std;

#define max 50

struct Stack{

int top;

int nut[max];

};

void Init(Stack &s){

s.top = -1;

return;

}

int IsEmpty(Stack s){

if(s.top == -1){

return 1;

}else return 0;

}

int IsFull(Stack s){

if(s.top == max - 1){

return 1;

}else return 0;

}

void Push(Stack &s, int x){

if(IsFull(s)){

cout<<"danh sach day.";

}else{

s.top ++;

s.nut[s.top]=x;

}

}

int Pop(Stack &s){

if(IsEmpty(s)){

cout<<"danh sach rong.";

}else return s.nut[s.top--];

}

int Top(Stack s){

if(IsEmpty(s)){

cout<<"danh sach rong.";

}else return s.nut[s.top];

}

void chuyendoicoso(Stack &s, int socandoi, int coso){

while(socandoi>0){

int n = socandoi%coso;

socandoi = socandoi/coso;

Push(s, n);

}

}

void Hienthi(Stack &s){

while(!IsEmpty(s)){

int x = Pop(s);

if(x<10){

cout<<x;

}else{

cout<<(char)('A'+x-10);

}

}

}

int main(){

Stack s;

Init(s);

int x;

int chon;

while(true){

cout<<endl;

cout<<"1. Nhap so can chuyen doi. \n";

cout<<"2. Chuyen thanh he nhi phan. \n";

cout<<"3. Chuyen thanh he bat phan. \n";

cout<<"4. Chuyen thanh he thap nhi phan. \n";

cout<<"0. Dung chuong trinh. \n";

cout<<"Moi ban chon: "; cin>>chon;

switch(chon){

case 1:{

do{

cout<<"So can chuyen: ";

cin>>x;

if(x<0){

cout<<"Nhap sai moi nhap lai."<<endl;

}

}while(x<0);

break;

}

case 2:{

cout<<"He nhi phan cua "<<x<<" la: "; chuyendoicoso(s, x, 2);

Hienthi(s);

break;

}

case 3:{

cout<<"He bat phan cua "<<x<<" la: "; chuyendoicoso(s, x, 8);

Hienthi(s);

break;

}

case 4:{

cout<<"He bat phan cua "<<x<<" la: "; chuyendoicoso(s, x, 16);

Hienthi(s);

break;

}

case 0: return 0;

}

}

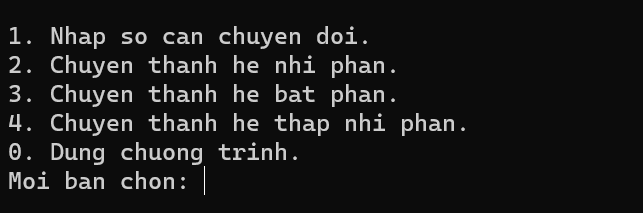
}

# Kiểm thử - Lập bộ Test

Bộ test :

|  |  |  |  |
| --- | --- | --- | --- |
| Hệ thập phân | Hệ nhị phân | Hệ Bát Phân | Hệ thập lục phân |
| 1844 | 111 0011 0100 | 3464 | 743 |
| 19421 | 100101111011101 | 45735 | 4BDD |
| 75 | 1001011 | 113 | 4B |
| 1943 | 11110010111 | 3627 | 797 |

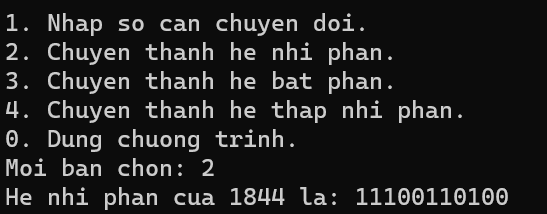
Màn hình sau khi chạy :



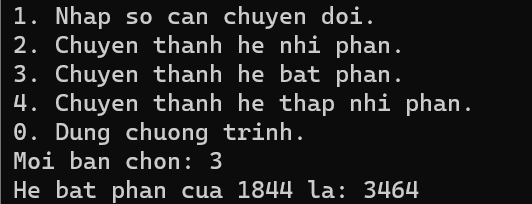
1. Nhập dữ liệu :

* Với hệ thập phân =1844

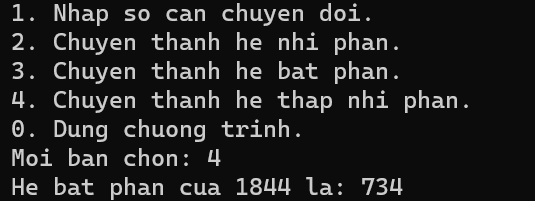
Kết quả khi chuyển sang nhị phân :



Kết quả khi chuyển sang hệ bát phân là :

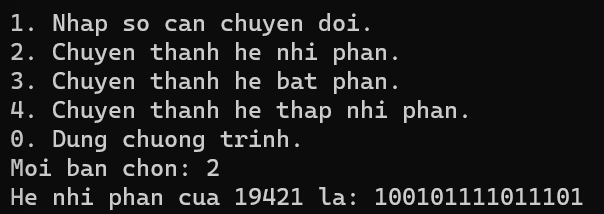


Kết quản khi chuyển sang hệ thập lục phân :

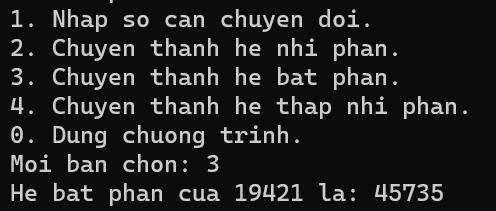


* Với hệ thập phân =19421

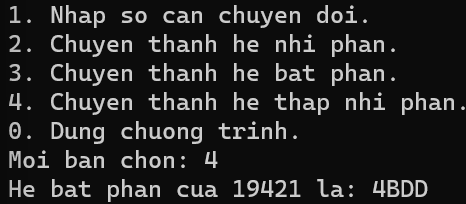
Kết quả khi chuyển sang nhị phân :



Kết quả khi chuyển sang hệ bát phân là :

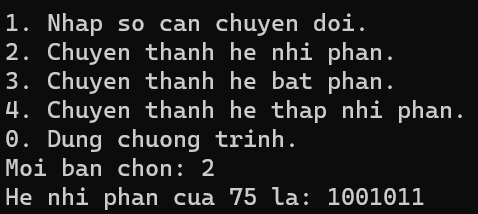


Kết quản khi chuyển sang hệ thập lục phân :

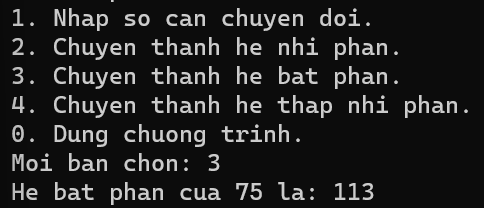


* Với hệ thập phân =75

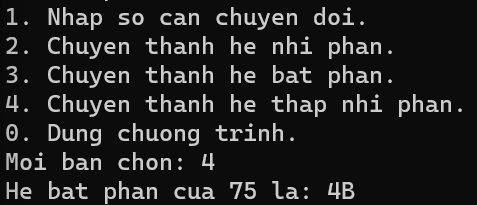
Kết quả khi chuyển sang nhị phân :



Kết quả khi chuyển sang hệ bát phân là :

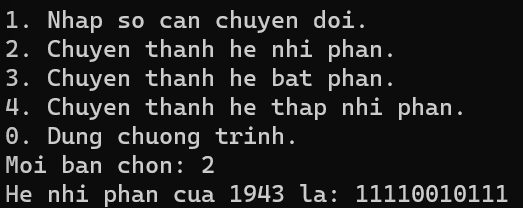


Kết quản khi chuyển sang hệ thập lục phân :

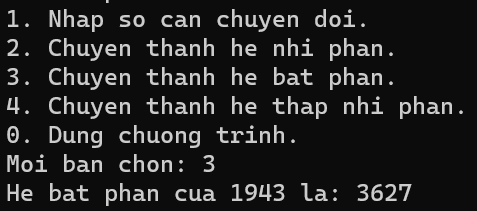


* Với hệ thập phân =1943

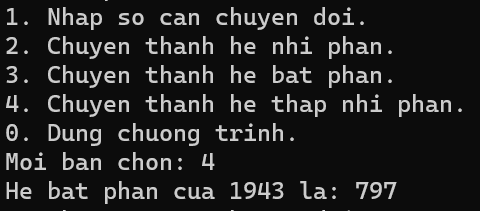
Kết quả khi chuyển sang nhị phân :



Kết quả khi chuyển sang hệ bát phân là :



Kết quản khi chuyển sang hệ thập lục phân :



Từ kiểm tra trên ta thấy chương trình đã chạy đũng bộ test