BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**VIẾT CHƯƠNG TRÌNH LIỆT KÊ CÁC DÃY NHỊ PHÂN**

**BIỂU DIỄN CÁC SỐ NGUYÊN THEO THỨ TỰ**

**Giảng viên hướng dẫn: ThS. Hà Thị Thanh Ngà**

**Sinh viên thực hiện: Võ Tấn Khuê**

**Mã số sinh viên: 61133814**

Khánh Hòa – 2021

# LỜI CẢM ƠN

Để hoàn thành bài báo cáo thực tập cơ sở này, trước hết em xin gửi đến quý thầy, cô Khoa Công nghệ Thông tin - Trường Đại học Nha Trang lời cảm ơn chân thành.

Em xin gửi đến cô Hà Thị Thanh Ngà, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành bài báo cáo này lời cảm ơn sâu sắc nhất.

Trong quá trình nghiên cứu thực hiện đề tài, cũng như là trong quá trình làm bài báo cáo thực tập cơ sở, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do kiến thức cũng như kinh nghiệm thực tiễn của bản thân còn hạn chế nên đề tài, bài báo cáo cơ sở này khó thể không tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp từ quý thầy, cô để em học hỏi thêm được nhiều kinh nghiệm, cũng như kỹ năng cần thiết.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc92661764)

[MỤC LỤC 2](#_Toc92661765)

[1.1 CƠ SỞ LÝ THUYẾT 3](#_Toc92661766)

[1.1.1 Hệ Nhị Phân 3](#_Toc92661767)

[1.1.2 Hệ Thập Phân 3](#_Toc92661768)

[1.1.3 Phương pháp chuyển Hệ thập phân sang Hệ nhị phân 3](#_Toc92661769)

[1.1.4 Thuật toán Đệ Quy 4](#_Toc92661770)

[1.1.5 Thuật toán Quay Lui 4](#_Toc92661771)

[1.1.6 Phương pháp sinh 4](#_Toc92661772)

[1.2 CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH ĐƯỢC SỬ DỤNG 5](#_Toc92661773)

[1.2.1 Tổng quan về Dev-C++ 5](#_Toc92661774)

[1.2.2 Ngôn ngữ C/C++ 5](#_Toc92661775)

[1.2.3 Ngôn ngữ C# 6](#_Toc92661776)

[1.2.4 Windows Form 6](#_Toc92661777)

[Phần 2. ĐẶC TẢ BÀI TOÁN 7](#_Toc92661778)

[2.1 BÀI TOÁN 7](#_Toc92661779)

[2.2 HƯỚNG XỬ LÝ VÀ CÁC THUẬT TOÁN ĐỂ THỰC HIỆN YÊU CẦU 7](#_Toc92661780)

[2.2.1 Mức 1 7](#_Toc92661781)

[2.2.2 Mức 2.1 8](#_Toc92661782)

[2.2.3 Mức 2.2 8](#_Toc92661783)

[Phần 3. CÀI ĐẶT CHƯƠNG TRÌNH 11](#_Toc92661784)

[3.1 CODE CHƯƠNG TRÌNH 11](#_Toc92661785)

[3.1.1 Mức 1 11](#_Toc92661786)

[3.1.2 Mức 2.1 16](#_Toc92661787)

[3.1.3 Mức 2.2 19](#_Toc92661788)

[Phần 4. ĐỒ HỌA CHƯƠNG TRÌNH 24](#_Toc92661789)

[4.1 GIAO DIỆN ĐỒ HỌA 24](#_Toc92661790)

[Phần 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc92661791)

[5.1 KẾT QUẢ ĐẠT ĐƯỢC 34](#_Toc92661792)

[5.2 ƯU ĐIỂM 34](#_Toc92661793)

[5.3 HẠN CHẾ 34](#_Toc92661794)

[5.4 HƯỚNG PHÁT TRIỂN 34](#_Toc92661795)

[TÀI LIỆU THAM KHẢO 35](#_Toc92661796)

Phần 1. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

## CƠ SỞ LÝ THUYẾT

### Hệ Nhị Phân

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). 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]

### Hệ Thập Phân

Hệ thập phân (hệ đếm cơ số 10) là hệ đếm dùng số 10 làm cơ số. Đây là hệ đếm được sử dụng rộng rãi nhất trong các nền văn minh thời hiện đại. Hệ thống ký tự các con số dùng để biểu đạt các giá trị trong một hệ đếm. Trong hệ thập phân, 10 ký tự (còn gọi là con số) khác nhau được dùng để biểu đạt 10 giá trị riêng biệt (0, 1, 2, 3, 4, 5, 6, 7, 8 và 9), tức là 10 con số. Những con số này còn được dùng cùng với dấu thập phân - ví dụ dấu "phẩy" - để định vị phần thập phân sau hàng đơn vị. Con số còn có thể được dẫn đầu bằng các ký hiệu "+" hay "-" để biểu đạt số dương và số âm nữa. [2]

### Phương pháp chuyển Hệ thập phân sang Hệ nhị phân

Nguyên tắc của phương pháp là lấy số cần chuyển đổi chia cho 2 (kết quả chỉ lấy phần nguyên), sau đó tiếp tục lấy kết quả chia 2 cho đến khi kết quả chia 2 bằng 0 thì dừng, kết quả số nhị phân thu được là tập hợp các số dư của các phép chia theo thứ tự từ dưới lên. [3]

Ví dụ: Tính số thập phân 10 sang nhị phân, chúng ta lấy 10 chia 2, kết quả được 5 và số dư là 0 chúng ta lấy số 5 chia 2, kết quả được 2 và số dư là 1, ta lấy số 2 chia 2, kết quả được 1 và dư 0, Ta tiếp tục lặp lại quá trình này cho đến khi kết quả chia 2 chúng ta được 0. Số nhị phân chúng ta thu được chính là tập hợp các số dư của các phép chia (lấy từ dưới lên) Vậy số 10 trong hệ nhị phân sẽ là 1010.

### Thuật toán Đệ Quy

Ta nói một đối tượng là đệ quy nếu nó được định nghĩa qua chính nó hoặc một đối tượng khác cùng dạng với chính nó bằng quy nạp.

Ví dụ: Đặt hai chiếc gương cầu đối diện nhau. Trong chiếc gương thứ nhất chứa hình chiếc gương thứ hai. Chiếc gương thứ hai lại chứa hình chiếc gương thứ nhất nên tất nhiên nó chứa lại hình ảnh của chính nó trong chiếc gương thứ nhất… Ở một góc nhìn hợp lý, ta có thể thấy một dãy ảnh vô hạn của cả hai chiếc gương. [4]

### Thuật toán Quay Lui

Thuật toán quay lui dùng để giải bài toán liệt kê các cấu hình. Mỗi cấu hình được xây dựng bằng cách xây dựng từng phần tử, mỗi phần tử được chọn bằng cách thử tất cả các khả năng.

Giả thiết cấu hình cần liệt kê có dạng (x1, x2,…, xn). Khi đó thuật toán quay lui thực hiện qua các bước sau:

1) Xét tất cả các giá trị x1 có thể nhận, thử cho x1 nhận lần lượt các giá trị đó. Với mỗi giá trị thử gán cho x1 ta sẽ:

2) Xét tất cả các giá trị x2 có thể nhận, lại thử cho x2 nhận lần lượt các giá trị đó. Với mỗi giá trị thử gán cho x2 lại xét tiếp các khả năng chọn x3 … tiếp tục đến:

…

n) Xét tất cả các giá trị xn có thể nhận, thử cho xn nhận lần lượt các giá trị đó, thông báo cấu hình tìm được (x1, x2,…, xn).

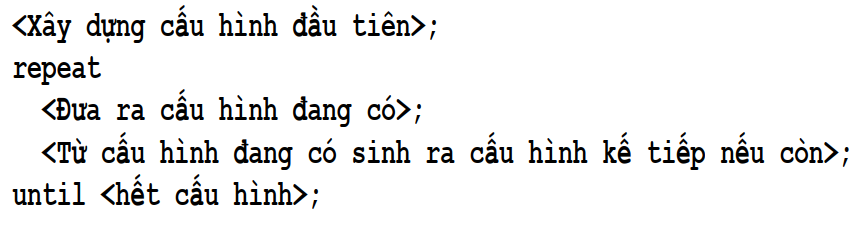
Trên phương diện quy nạp, có thể nói rằng thuật toán quay lui liệt kê các cấu hình n phần tử dạng (x1, x2, .., xn) bằng cách thử cho x1 nhận lần lượt các giá trị có thể. Với mỗi giá trị thử gán cho x1 lại liệt kê tiếp cấu hình n - 1 phần tử (x2, x3, …, xn). [5]

### Phương pháp sinh

Phương pháp sinh có thể áp dụng để giải bài toán liệt kê tổ hợp đặt ra nếu như hai điều kiện sau thoả mãn:

* Có thể xác định được một thứ tự trên tập các cấu hình tổ hợp cần liệt kê. Từ đó có thể biết được cấu hình đầu tiên và cấu hình cuối cùng trong thứ tự đó.
* Xây dựng được thuật toán từ một cấu hình chưa phải cấu hình cuối, sinh ra được cấu hình kế tiếp nó. [5]

Phương pháp sinh có thể mô tả như sau:



Hình 1.1.6. Mô hình của phương pháp sinh

## CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH ĐƯỢC SỬ DỤNG

### Tổng quan về Dev-C++

Dev-C++ là một môi trường phát triển tích hợp tự do (IDE) được phân phối dưới hình thức giấy phép Công cộng GNU hỗ trợ việc lập trình bằng C/C++. Nó cũng nằm trong bộ trình dịch mã nguồn mở MinGW. Chương trình IDE này được viết bằng ngôn ngữ Delphi. Dự án phát triển Dev-C++ được lưu trữ trên SourceForge. Dev-C++ nguyên được phát triển bởi một lập trình viên có tên là Colin Laplace và chỉ chạy trên hệ điều hành Microsoft Windows. [6]

### Ngôn ngữ C/C++

C++ là một loại ngôn ngữ lập trình bậc trung (middle-level). Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. [7]

### Ngôn ngữ C#

C# (C Sharp, đọc là "xi-sáp") là một ngôn ngữ lập trình hướng đối tượng đa năng, mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java. [8]

### Windows Form

Windows Forms (WinForms) là thư viện lớp đồ họa (GUI) mã nguồn mở và miễn phí được bao gồm như một phần của Microsoft.NET Framework hoặc Mono Framework, cung cấp nền tảng để viết các ứng dụng khách phong phú cho máy tính để bàn, máy tính xách tay và máy tính bảng. Mặc dù nó được coi là sự thay thế cho Thư viện lớp nền tảng Microsoft Foundation của C ++ trước đây và phức tạp hơn, nhưng nó không cung cấp mô hình tương đương và chỉ hoạt động như một nền tảng cho tầng giao diện người dùng trong một giải pháp nhiều tầng. [9]

# Phần 2. ĐẶC TẢ BÀI TOÁN

## 2.1 BÀI TOÁN

Viết chương trình liệt kê lần lượt các dãy nhị phân biễn diễn các số nguyên theo thứ tự 0, 1,... , 2n-1

Mục đích: thể hiện giải thuật dưới dạng chương trình hoàn chỉnh

Mức 1:

1. Trình bày giải thuật thực hiện chương trình
2. Sử dụng ngôn ngữ: Pascal/C/C++ để thể hiện

* Nhập số nguyên n
* Hiển thị lần lượt các dãy nhị phân biểu diễn số nguyên theo thứ tự 0,1,… 2n-1

Mức 2:

1. Nhập file văn bản chứa số nguyên n<=100 và ghi ra file chứa các dãy nhị phân độ dài n.
2. Liệt kê các dãy nhị phần độ dài n mà trong đó cụm chữ số “01” xuất hiện đúng 2 lần.

## HƯỚNG XỬ LÝ VÀ CÁC THUẬT TOÁN ĐỂ THỰC HIỆN YÊU CẦU

### Mức 1

Sử dụng thuật toán chuyển Hệ thập phân sang Hệ nhị phân đã được miêu tả ở mục 1.1.3

Ý tưởng của mình là:

Bước 1: Với mỗi phần tử số nguyên từ 0 đến 2n-1 đem chia cho 2 (kết quả chỉ lấy phần nguyên), sau đó tiếp tục lấy kết quả đó chia 2 cho đến khi kết quả chia 2 bằng 0 thì dừng, kết quả tập hợp các số dư của các phép chia được lưu vào chuỗi s.

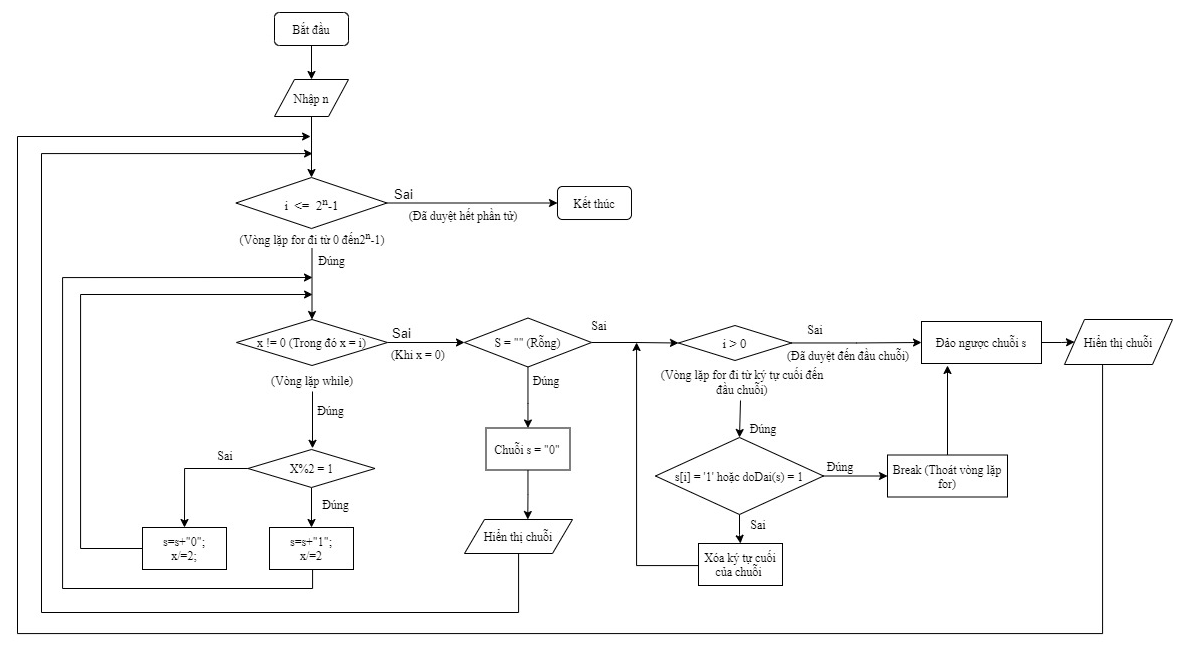
Bước 2: Kiểm tra chuỗi s có giá trị rỗng thì gán chuỗi s = 0 (Tránh bỏ trường hợp số nguyên 0)

Bước 3: Sau đó kiểm tra tiếp chuỗi s, xuất phát ký tự cuối đến ký tự đầu của chuỗi

* Nếu ký tự đó = 1 hoặc độ dài của chuỗi = 1 thì đến bước 4
* Ngược lại xóa đi một ký tự cuối của chuỗi và lặp lại bước 3

Bước 4: Đảo ngược và in chuỗi s ra màn hình

Ý tưởng được thể hiện qua sơ đồ khối như sau:

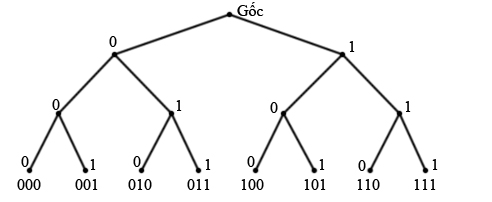


Hình 2.2.1. Sơ đồ khối thuật toán mức 1

### Mức 2.1

Sử dụng thuật toán đệ quy quay lui đã được miêu tả ở mục 1.1.5

Ý tưởng được thể hiện qua hình (ví dụ n = 3) như sau:



Hình 2.2.2. Ví dụ với n=3

### Mức 2.2

Sử dụng phương pháp sinh được miêu tả ở mục 1.1.6

Ý tưởng của mình là tạo 2 chuỗi: (Nếu n > 4 thì mới dùng chuỗi 2)

* Chuỗi 1

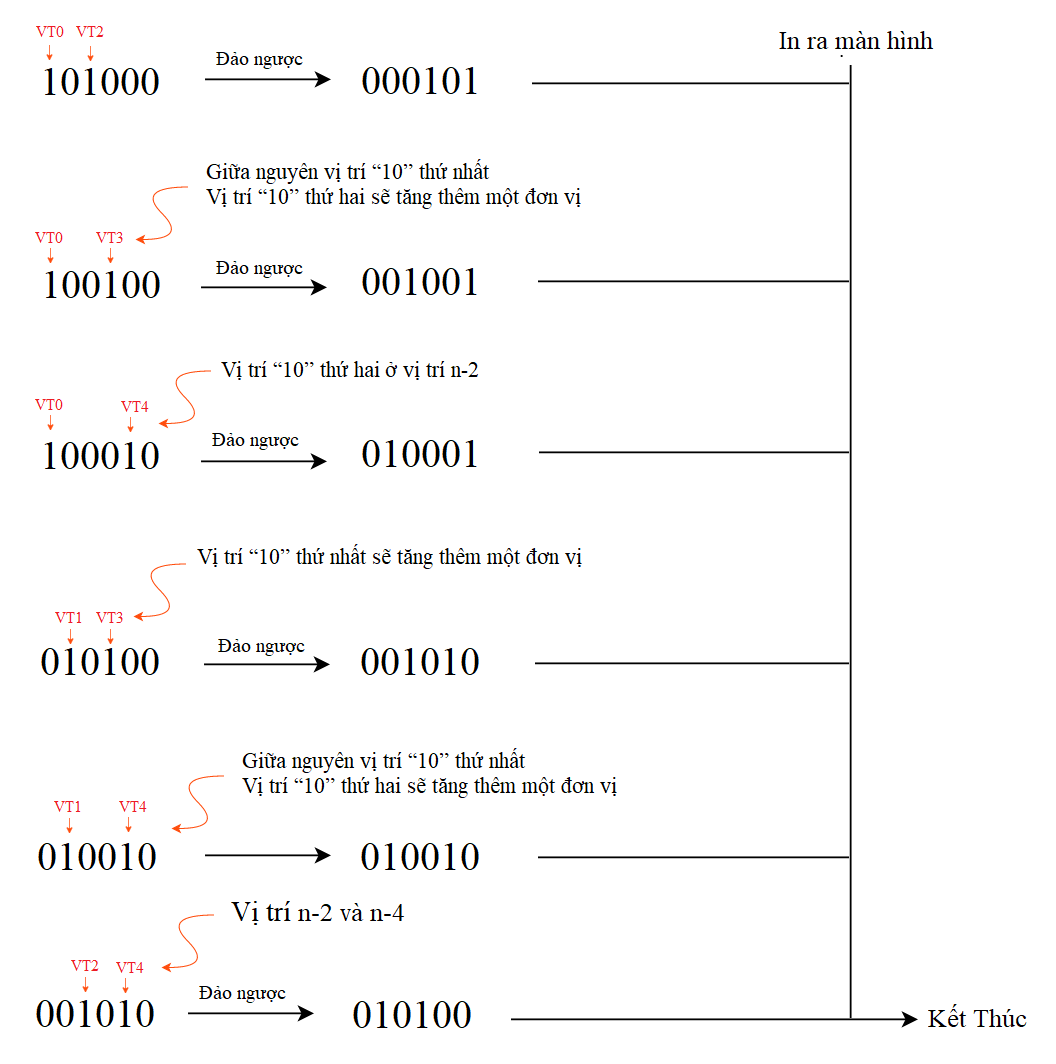
Bước 1: đầu tiên sẽ đánh dấu 2 vị trí được chọn để chứa 2 cặp “10” (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 0 và 2: 101000), sau đó điền các giá trị “10” vào các vị trí đó và **điền giá trị 0** vào các vị trí còn lại.

Bước 2: Sau đó duyệt từ trái sang phải, vị trí “10” thứ 2 sẽ tăng thêm một đơn vị (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 0 và 3: 100100), đến khi vị trí “10” thứ 2 đến vị trí n-2 của chuỗi (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 0 và 4: 100010) thì vị trí “10” thứ nhất sẽ tăng thêm một đơn vị (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 1 và 3: 010100). Và lặp lại bước 2.

Bước 3: Khi 2 vị trí “10” ở vị trí n-2 và n-4 (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 2 và 4: 000101) thì dừng.

Bước 4: Sau mỗi lần tìm được một chuỗi thì **đảo ngược** và in chuỗi tìm được.

Ý tưởng được thể hiện qua hình (ví dụ cụ thể n = 6) như sau:



Hình 2.2.3. Ví dụ với n=6

* Chuỗi 2 (gần như tương tự với chuỗi 1)

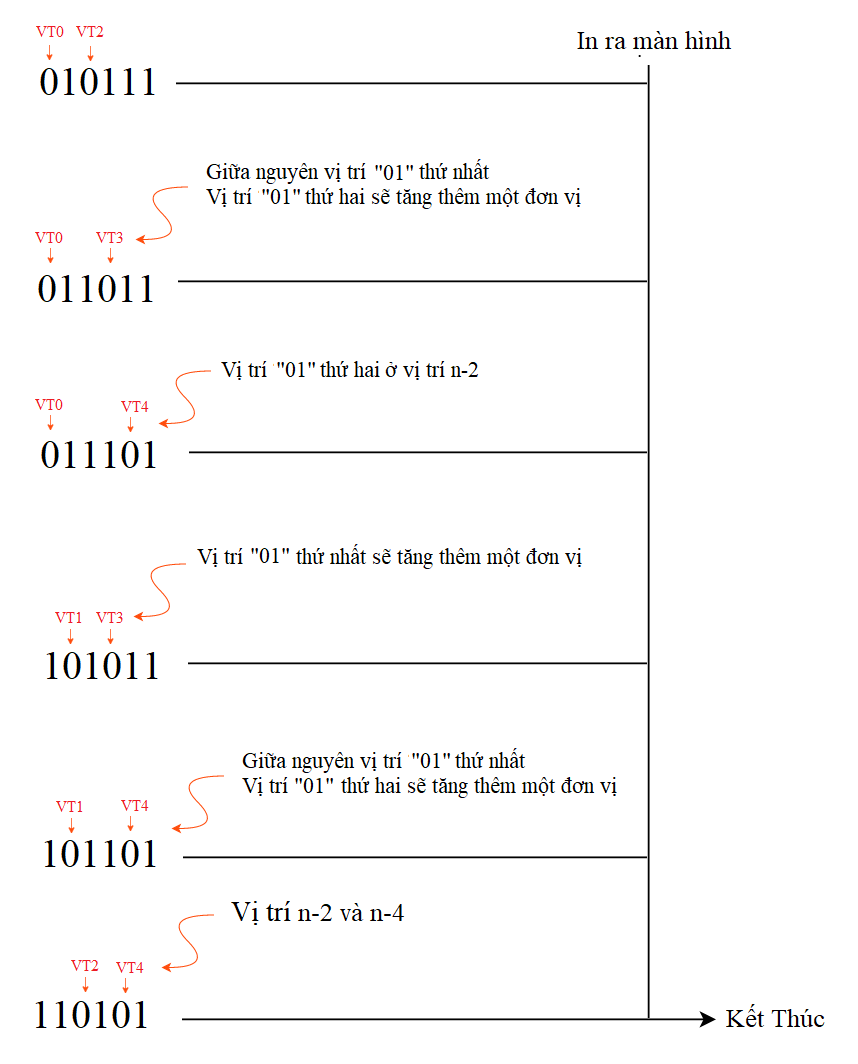
Bước 1: đầu tiên sẽ đánh dấu 2 vị trí được chọn để chứa 2 cặp “01” (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 0 và 2: 010111), sau đó điền các giá trị “01” vào các vị trí đó và **điền giá trị 1** vào các vị trí còn lại.

Bước 2: Sau đó duyệt từ trái sang phải, vị trí “01” thứ 2 sẽ tăng thêm một đơn vị (ví dụ nếu n = 6 thì 2 vị trí “01” ở chuỗi là 0 và 3: 011011), đến khi vị trí “10” thứ 2 đến vị trí n-2 của chuỗi (ví dụ nếu n = 6 thì 2 vị trí “01” ở chuỗi là 0 và 4: 011101) thì vị trí “01” thứ nhất sẽ tăng thêm một đơn vị (ví dụ nếu n = 6 thì 2 vị trí “01” ở chuỗi là 1 và 3: 101011). Và lặp lại bước 2.

Bước 3: Khi 2 vị trí “01” ở vị trí n-2 và n-4 (ví dụ nếu n = 6 thì 2 vị trí “10” ở chuỗi là 2 và 4: 110101) thì dừng.

Bước 4: Sau mỗi lần tìm được một chuỗi thì in chuỗi tìm được.

Ý tưởng được thể hiện qua hình (ví dụ cụ thể n = 6) như sau:



Hình 2.2.4. Ví dụ với n=6

# Phần 3. CÀI ĐẶT CHƯƠNG TRÌNH

## 3.1 CODE CHƯƠNG TRÌNH

### 3.1.1 Mức 1

Khai báo thư viện C++ và khai báo hằng số ll có kiểu dữ liệu là long long:

|  |
| --- |
| *#include <bits/stdc++.h>*  *#define ll long long*  *using namespace std;* |

Xây dựng hàm **doDai**()có chức năngtrả về độ dài chuỗi:

Đầu tiên khai báo biến cnt kiểu số nguyên có giá trị bằng 0, chạy một vòng lặp for từ đầu chuỗi đến cuối chuỗi với mỗi lần lặp tăng biến cnt lên một giá trị. Kết thúc vòng lặp trả về giá trị của hàm là độ dài của một chuỗi.

|  |
| --- |
| *// Ham tra ve do dai chuoi*  *int* ***doDai****(string s)*  *{*  *int cnt = 0;*  *for (int i = 0; s[i] != NULL; i++)*  *cnt++;*  *return cnt;*  *}* |

Xây dựng hàm **daoNguoc**() có chức năng đảo ngược chuỗi:

Đầu tiên khai báo một chuỗi tên là dao có giá trị “”, khai báo biến l có giá trị là độ dài của chuỗi cần đảo. Tiếp theo chạy vòng lặp for bắt đầu từ ký tự cuối cùng của chuỗi chạy ngược về đầu chuỗi, qua mỗi lần lặp ta gán ký tự đó vào chuỗi dao. Kết thúc vòng lặp trả về giá trị của hàm là một chuỗi đã được đảo.

|  |
| --- |
| *// Ham dao nguoc chuoi*  *string* ***daoNguoc****(string s)*  *{*  *string dao("");*  *int l = doDai(s);*  *for (int i = l - 1; i >= 0; --i)*  *dao += s[i];*  *return dao;*  *}* |

Xây dựng hàm **xoaKyTuCuoi**() có chức năng xóa ký tự cuối của một chuỗi:

Đầu tiên khai báo một chuỗi tên là r có giá trị “”, dùng một vòng lặp for chạy từ ký tự đầu của chuỗi đến ký tự bên cạnh ký tự cuối của chuỗi, qua mỗi lần lặp ta gán ký tự vào chuỗi r. Kết thúc vòng lặp trả về giá trị của hàm là một chuỗi đã được xóa ký tự cuối.

|  |
| --- |
| *//Ham xoa ky tu cuoi cua chuoi*  *string* ***xoaKyTuCuoi****(string s)*  *{*  *string r("");*  *for (ll i = 0; i < doDai(s) - 1; ++i)*  *r += s[i];*  *return r;*  *}* |

Xây dựng hàm **muc\_01**() có chức năng Hiển thị lần lượt các dãy nhị phân biểu diễn số nguyên theo thứ tự 0,1,… 2n-1.

Đầu tiên nhập số nguyên n, chạy một vòng lặp for i từ 0 đến 2n-1 với mỗi lần lặp:

Khai báo biến long long x có giá trị = i

Khai báo chuỗi s có giá trị là “”

Chạy một vòng lặp while khi x khác 0 với mỗi lần lặp:

* + Nếu biến x % 2 phần nguyên là 1 → gán thêm cho chuỗi s là 1
  + Nếu biến x % 2 phần nguyên là 0 → gán thêm cho chuỗi s là 0

Giá trị x = x / 2

Kiểm tra nếu chuỗi s có giá trị là “” thì gán cho chuỗi s là 0 (Trường hợp số nguyên là 0) và in chuỗi, ngược lại chạy một vòng lặp i từ ký tự cuối đến ký tự đầu của chuỗi với mỗi lần lặp:

* Nếu ký tự đó = 1 hoặc **doDai**(s) = 1 thì thoát khỏi vòng lặp
* Ngược lại **xoaKyTuCuoi**(s)

Cuối cùng **daoNguoc**(s) và in chuỗi ra màn hình.

|  |
| --- |
| *void muc\_01()*  *{*  *int n;*  *cout << "Hay nhap n: ";*  *cin >> n;*  *for (ll i = 0; i < (1LL << n); i++) //(1LL << n) chính là 2n*  *{*  *ll x = i;*  *string s = "";*  *while (x != 0)*  *{*  *if (x % 2 == 1)*  *s = s + "1";*  *else*  *s = s + "0";*  *x /= 2;*  *}*  *if (s == "")*  *s = "0";*  *else*  *for (ll i =* ***doDai****(s) - 1; i > 0; --i)*  *{*  *if (s[i] == '1' || doDai(s) == 1)*  *break;*  *else*  *s =* ***xoaKyTuCuoi****(s);*  *}*    *s =* ***daoNguoc****(s);*  *cout << s << "\n";*  *}*  *}* |

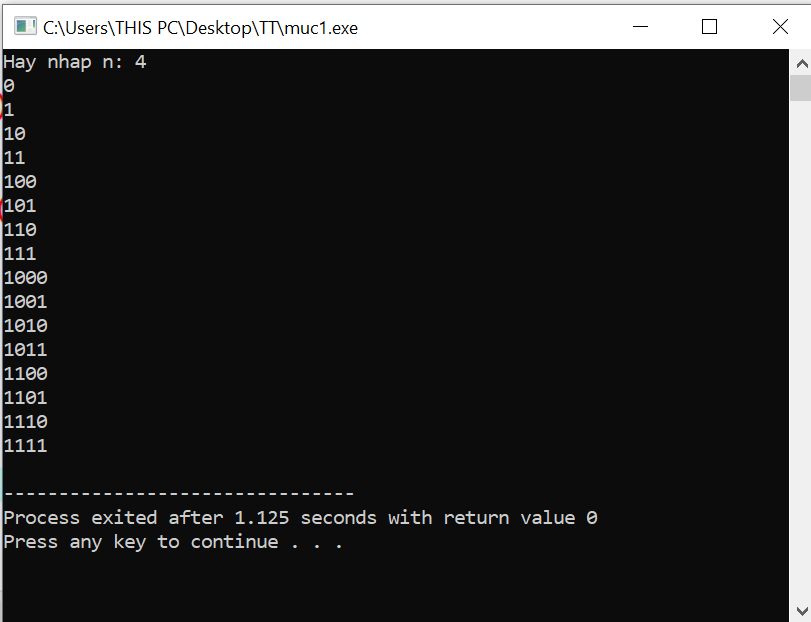
Hàm chương trình chính:

|  |
| --- |
| *int* ***main****()*  *{*  *muc\_01();*  *return 0;*  *}* |

Tổng hợp code chương trình hoàn chỉnh:

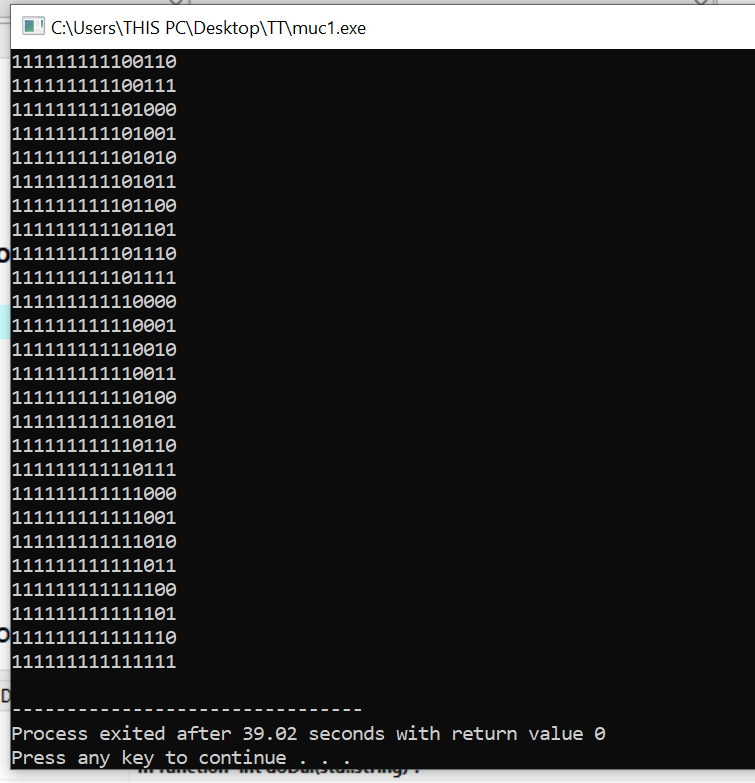
|  |
| --- |
| *#include <bits/stdc++.h>*  *#define ll long long*  *using namespace std;*  *int doDai(string s)*  *{*  *int cnt = 0;*  *for (int i = 0; s[i] != NULL; i++)*  *cnt++;*  *return cnt;*  *}*  *string daoNguoc(string s)*  *{*  *string dao("");*  *int l = doDai(s);*  *for (int i = l - 1; i >= 0; --i)*  *dao += s[i];*  *return dao;*  *}*  *string xoaKyTuCuoi(string s)*  *{*  *string r("");*  *for (ll i = 0; i < doDai(s) - 1; ++i)*  *r += s[i];*  *return r;*  *}*  *void muc\_01()*  *{*  *int n;*  *cout << "Hay nhap n: ";*  *cin >> n;*  *for (ll i = 0; i < (1LL << n); i++)*  *{*  *ll x = i;*  *string s = "";*  *while (x != 0)*  *{*  *if (x % 2 == 1)*  *s = s + "1";*  *else*  *s = s + "0";*  *x /= 2;*  *}*  *if (s == "")*  *s = "0";*  *else*  *for (ll i = doDai(s) - 1; i > 0; --i)*  *{*  *if (s[i] == '1' || doDai(s) == 1)*  *break;*  *else*  *s = xoaKyTuCuoi(s);*  *}*  *s = daoNguoc(s);*  *cout << s << "\n";*  *}*  *}*  *int main()*  *{*  *muc\_01();*  *return 0;*  *}* |

Kết quả chạy thử chương trình khi cho n=4 như sau:



Hình 3.1.1. Kết quả khi nhập n=4

Kết qua chạy thử chương trình khi cho n = 15 như sau:



Hình 3.1.2. Ví dụ chuyển số 30 sang hệ nhị phân

### 3.1.2 Mức 2.1

Khai báo thư viện C++ :

|  |
| --- |
| *#include <bits/stdc++.h>*  *#include <iostream>*  *using namespace std;* |

Xây dựng hàm **writeToFile**() có chức năngghi vào file:

Tạo biến f thuộc đối tượng fstream (tên file, ghi dữ liệu vào flle)

Ghi dữ liệu chuỗi vào file f và đóng file

|  |
| --- |
| *void* ***writeToFile****(const char\* fileName, string binStr) {*  *fstream f(fileName, ios::app);*  *f << binStr << endl;*  *f.close();*  *}* |

Xây dựng hàm **dequy()** có chức năng in ra các dãy nhị phân có độ dài n:

Dùng câu lệnh if

* + Nếu n = 0 thì ghi dữ liệt vào file output.txt
  + Ngược lại chạy một vòng lặp for i từ 0 đến 1, mỗi lần lặp gọi lại hàm **dequy()** với n-1 và s + char(i + ‘0’)

|  |
| --- |
| *void* ***dequy****(int n, string s) {*  *if (n == 0)*  *writeToFile("output.txt", s);*  *else {*  *for (int i = 0; i <= 1; i++)*  *{*  *dequy(n - 1, s + char(i + '0'));*  *}*  *}*  *}* |

Hàm chương trình chính:

Đọc giá tri n trong file input.txt, Nếu:

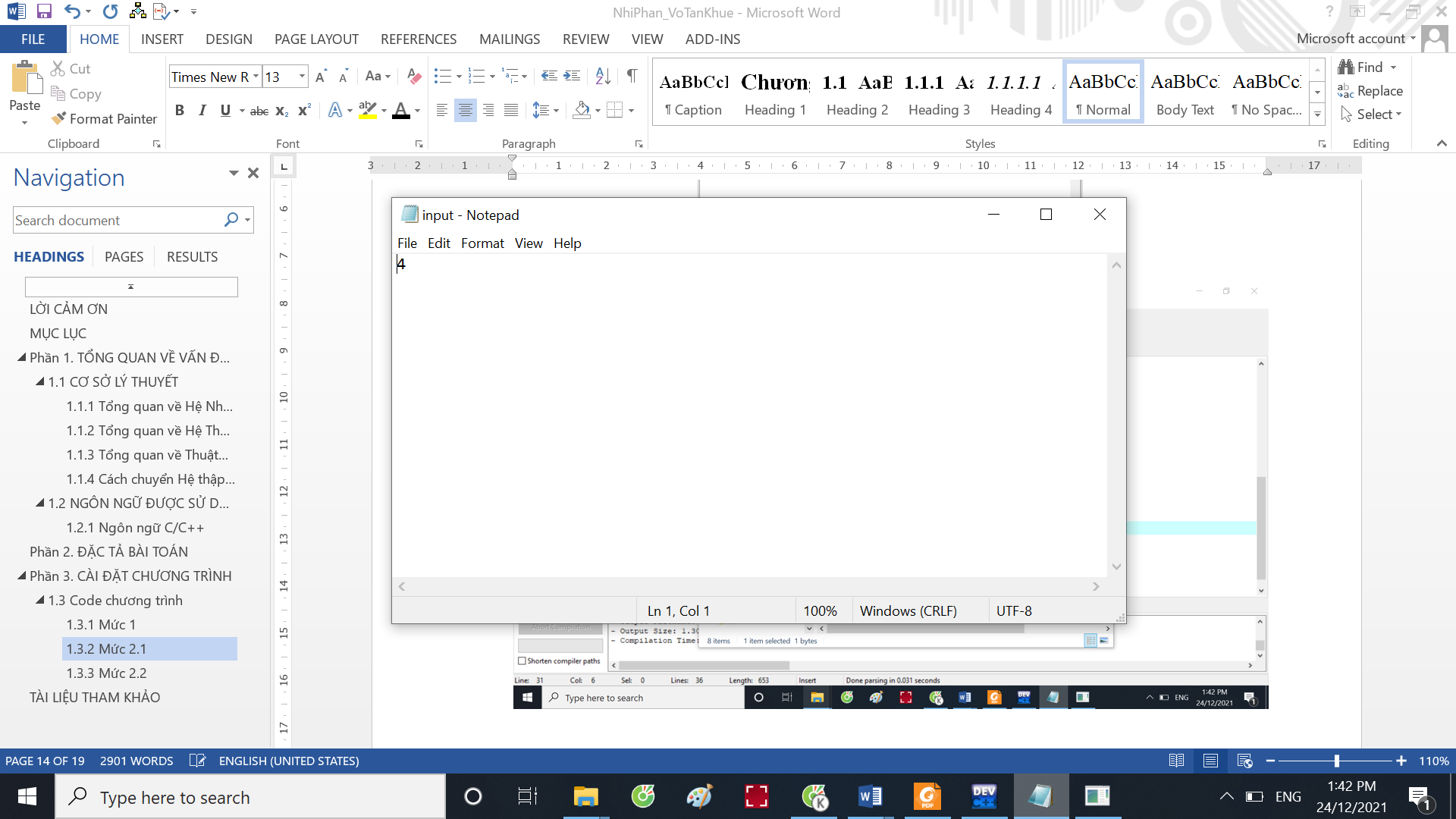
* Giá trị n > 100 thì thông báo nhập lại n
* N < 100 thì thực hiện chương trình

|  |
| --- |
| *int* ***main****() {*  *int n;*  *freopen("input.txt", "r", stdin);*  *cin >> n;*  *if (n > 100)*  *cout << "Nhap lai tham so n<=100 trong file";*  *else*  *{*  *writeToFile("output.txt", "==== Day nhi phan co do dai la n ==");*  *dequy(n, "");*  *}*  *return 0;*  *}* |

Tổng hợp code chương trình hoàn chỉnh:

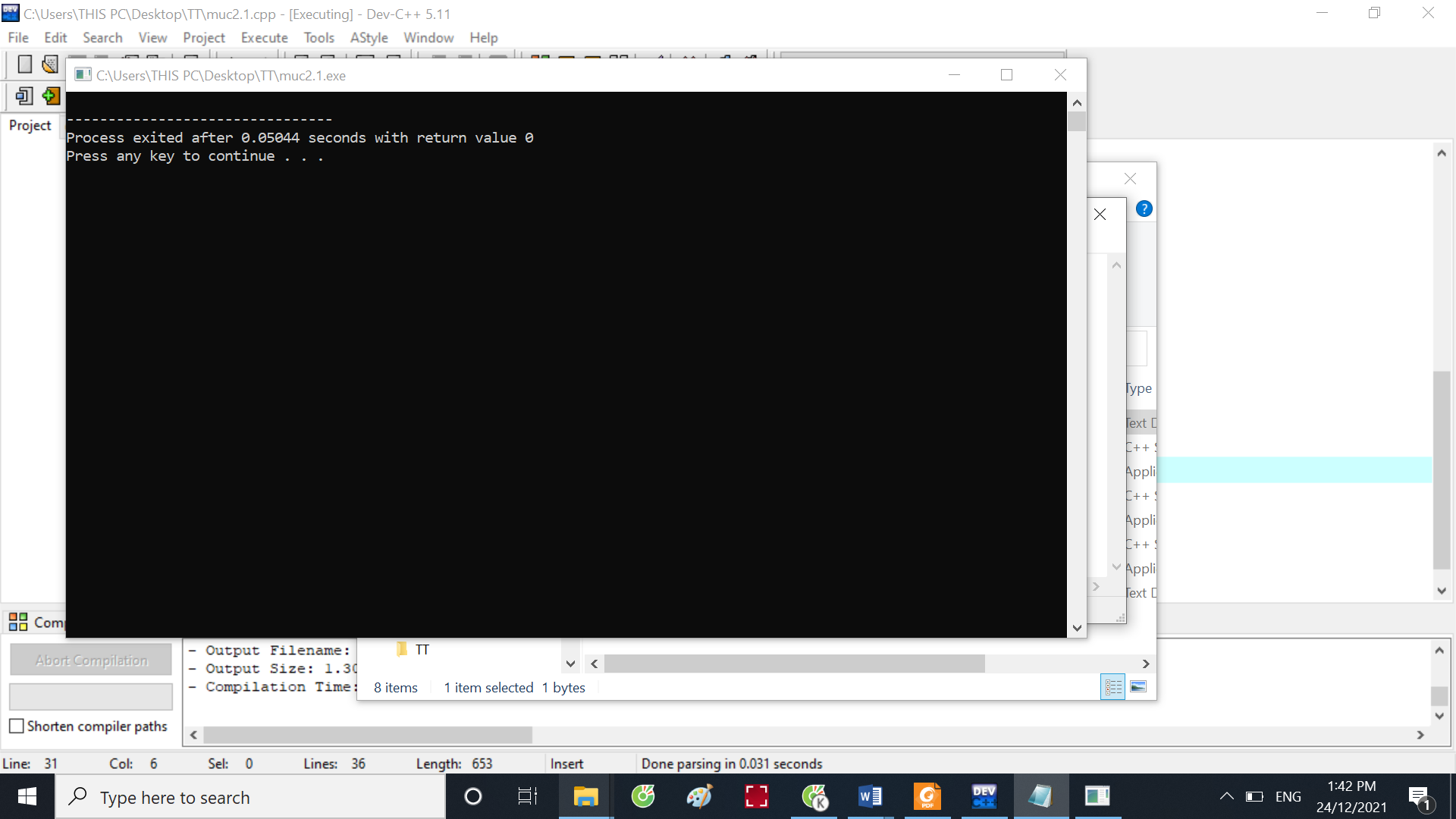
|  |
| --- |
| *#include <bits/stdc++.h>*  *#include <iostream>*  *using namespace std;*  *void writeToFile(const char\* fileName, string binStr) {*  *fstream f(fileName, ios::app);*  *f << binStr << endl;*  *f.close();*  *}*  *void dequy(int n, string s) {*  *if (n == 0)*  *writeToFile("output.txt", s);*  *else {*  *for (int i = 0; i <= 1; i++)*  *{*  *dequy(n - 1, s + char(i + '0'));*  *}*  *}*  *}*  *int main() {*  *int n;*  *freopen("input.txt", "r", stdin);*  *cin >> n;*  *if (n > 100)*  *cout << "Nhap lai tham so n<=100 trong file";*  *else*  *{*  *writeToFile("output.txt", "=== Day nhi phan co do dai la n ===");*  *dequy(n, "");*  *}*  *return 0;*  *}* |

Kết quả chạy thử chương trình khi cho file input là n=4 như sau:



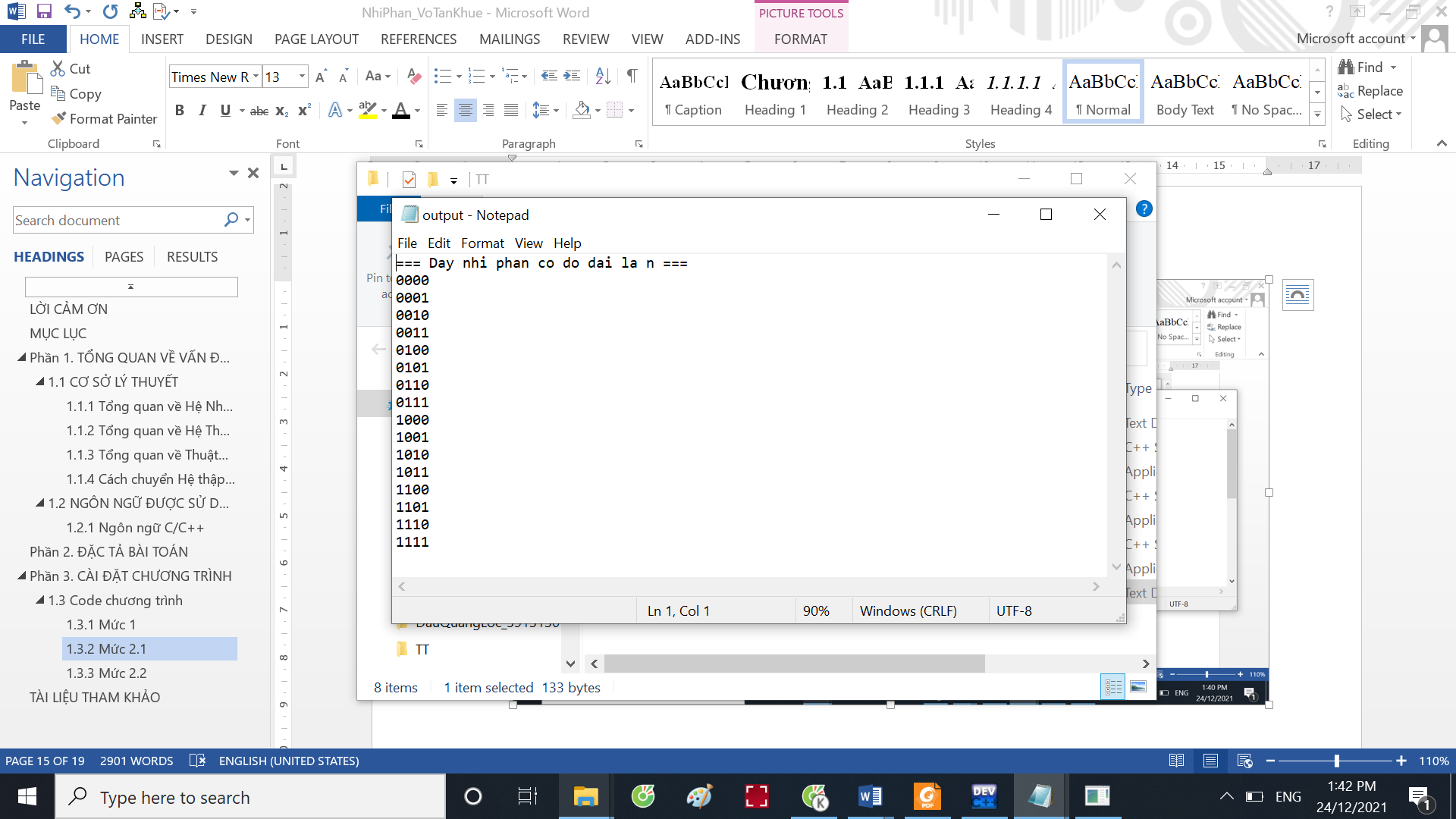
Hình 3.2.1. Nhập file input n=4

Màn hình Console:



Hình 3.2.2. Kết quả màn hình Console

Kết quả trong file output:



Hình 3.2.3. Kết quả trong file output

### 3.1.3 Mức 2.2

Khai báo thư viện C++ và khai báo hằng số ll có kiểu dữ liệu là long long:

|  |
| --- |
| *#include <bits/stdc++.h>*  *#include <iostream>*  *#define ll long long*  *using namespace std;* |

Hàm trả về độ dài chuỗi và đảo ngược chuỗi tương tự như mức 1

|  |
| --- |
| *// Ham tra ve do dai chuoi*  *int* ***doDai****(string s)*  *{*  *int cnt = 0;*  *for (int i = 0; s[i] != NULL; i++)*  *cnt++;*  *return cnt;*  *}* |

|  |
| --- |
| *// Ham dao nguoc chuoi*  *string* ***daoNguoc****(string s)*  *{*  *string dao("");*  *int l =* ***doDai****(s);*  *for (int i = l - 1; i >= 0; --i)*  *dao += s[i];*  *return dao;*  *}* |

Xây dựng hàm **muc\_02()** có chức năng in ra dãy nhị phân độ dài n có “01” xuất hiện 2 lần:

|  |
| --- |
| *void* ***muc\_02****(int n)*  *{*  *for (int i = 0; i < n - 1; i++)*  *{*  *for (int j = i + 2; j < n - 1; j++)*  *{*  *string h = "";*  *for (int e = 0; e < n; e++)*  *{*  *if (e == i || e == j)*  *h = h + "1";*  *else*  *h = h + "0";*  *}*  *h = daoNguoc(h);*  *cout << h << "\n";*  *}*  *}*  *if (n > 4)*  *for (int i = 0; i < n - 1; i++)*  *{*  *for (int j = i + 2; j < n - 1; j++)*  *{*  *string s = "";*  *for (int e = 0; e < n; e++)*  *{*  *if (e == i || e == j)*  *s = s + "0";*  *else*  *s = s + "1";*  *}*  *cout << s << "\n";*  *}*  *}*  *}* |

Hàm chương trình chính:

Đọc giá tri n trong file input.txt, Nếu:

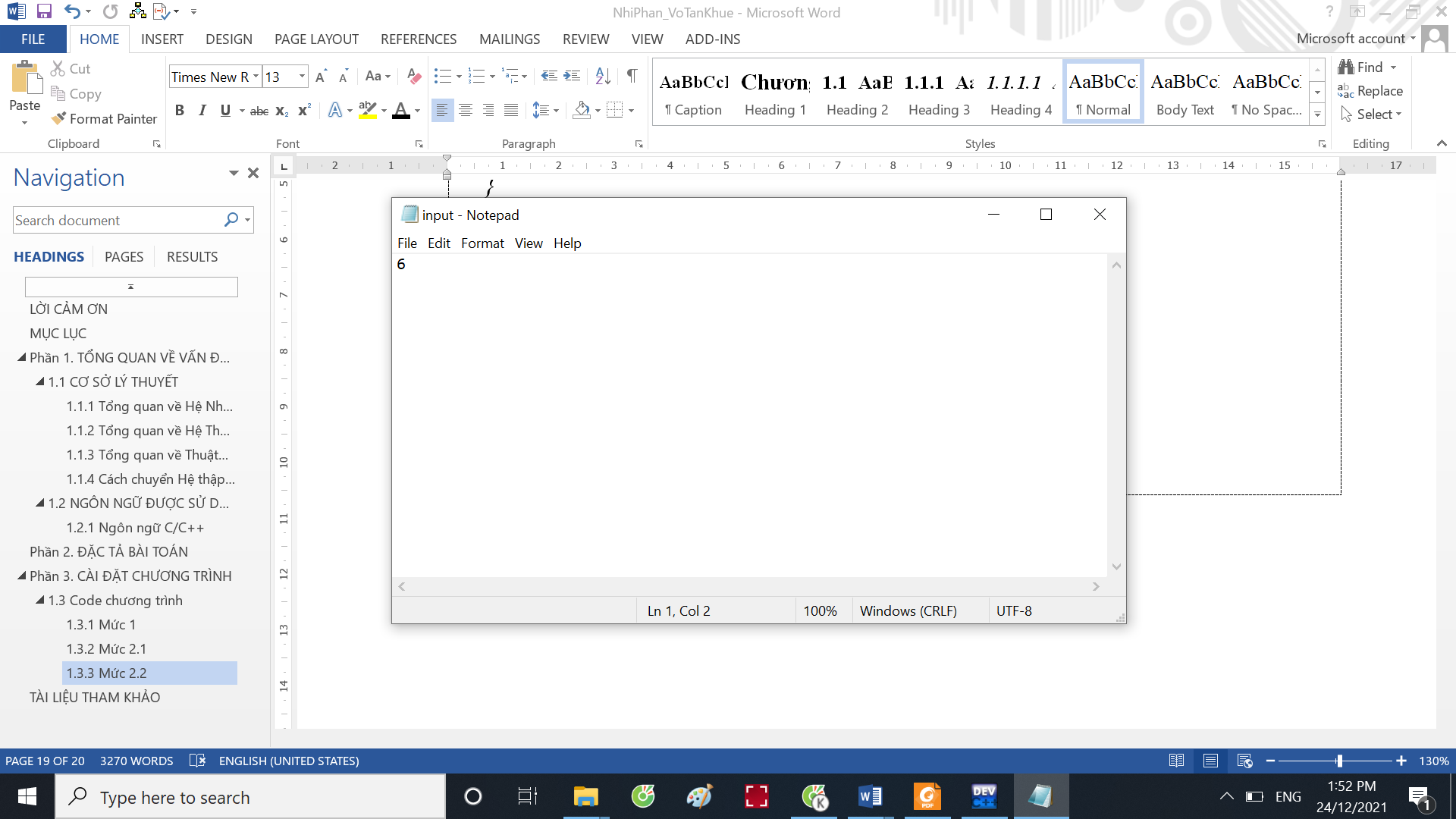
* Giá trị n > 100 thông báo nhập lại n và n <3 thông báo độ dái n không đủ để thực hiện bài toán
* N < 100 thì thực hiện chương trình

|  |
| --- |
| *int* ***main****() {*  *int n;*  *freopen("input.txt", "r", stdin);*  *cin >> n;*  *if (n > 100)*  *{*  *cout << "Nhap lai tham so n<=100 trong file";*  *}*  *else {*  *if (n <= 3)*  *cout << "Do dai n khong du de thuc hien bai toan";*  *else {*  *cout << "Chuoi 01 xuat hien 2 lan \n";*  ***muc\_02****(n);*  *}*  *}*  *return 0;*  *}* |

Tổng hợp code chương trình hoàn chỉnh:

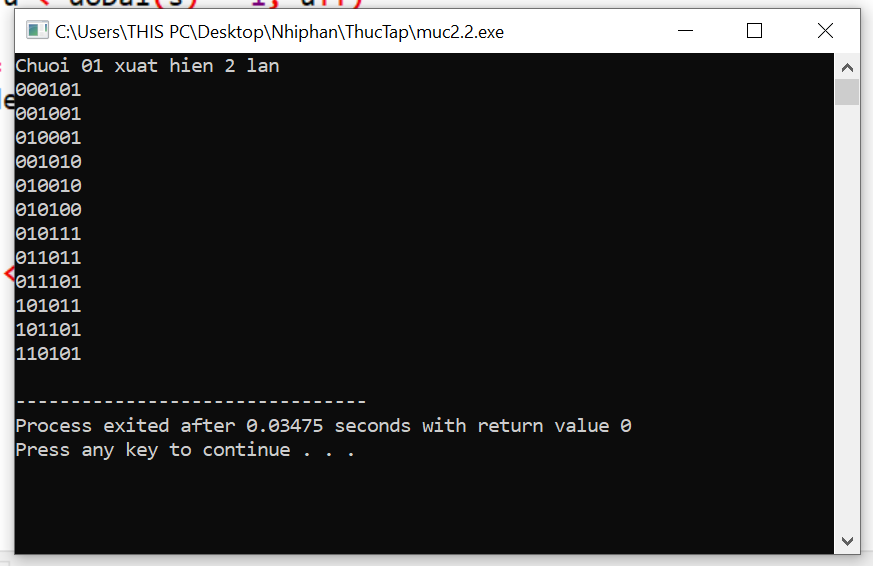
|  |
| --- |
| *#include <bits/stdc++.h>*  *#include <iostream>*  *#define ll long long*  *using namespace std;*  *int doDai(string s)*  *{*  *int cnt = 0;*  *for (int i = 0; s[i] != NULL; i++)*  *cnt++;*  *return cnt;*  *}*  *string daoNguoc(string s)*  *{*  *string dao("");*  *int l = doDai(s);*  *for (int i = l - 1; i >= 0; --i)*  *dao += s[i];*  *return dao;*  *}*  *void muc\_02(int n)*  *{*  *for (int i = 0; i < n - 1; i++)*  *{*  *for (int j = i + 2; j < n - 1; j++)*  *{*  *string h = "";*  *for (int e = 0; e < n; e++)*  *{*  *if (e == i || e == j)*  *h = h + "1";*  *else*  *h = h + "0";*  *}*  *h = daoNguoc(h);*  *cout << h << "\n";*  *}*  *}*  *if (n > 4)*  *for (int i = 0; i < n - 1; i++)*  *{*  *for (int j = i + 2; j < n - 1; j++)*  *{*  *string s = "";*  *for (int e = 0; e < n; e++)*  *{*  *if (e == i || e == j)*  *s = s + "0";*  *else*  *s = s + "1";*  *}*  *cout << s << "\n";*  *}*  *}*  *}*  *int main() {*  *int n;*  *freopen("input.txt", "r", stdin);*  *cin >> n;*  *if (n > 100)*  *{*  *cout << "Nhap lai tham so n<=100 trong file";*  *}*  *else {*  *if (n <= 3)*  *cout << "Do dai n khong du de thuc hien bai toan";*  *else {*  *cout << "Chuoi 01 xuat hien 2 lan \n";*  *muc\_02(n);*  *}*  *}*  *return 0;*  *}* |

Kết quả chạy thử chương trình khi cho file input là n = 6 như sau:



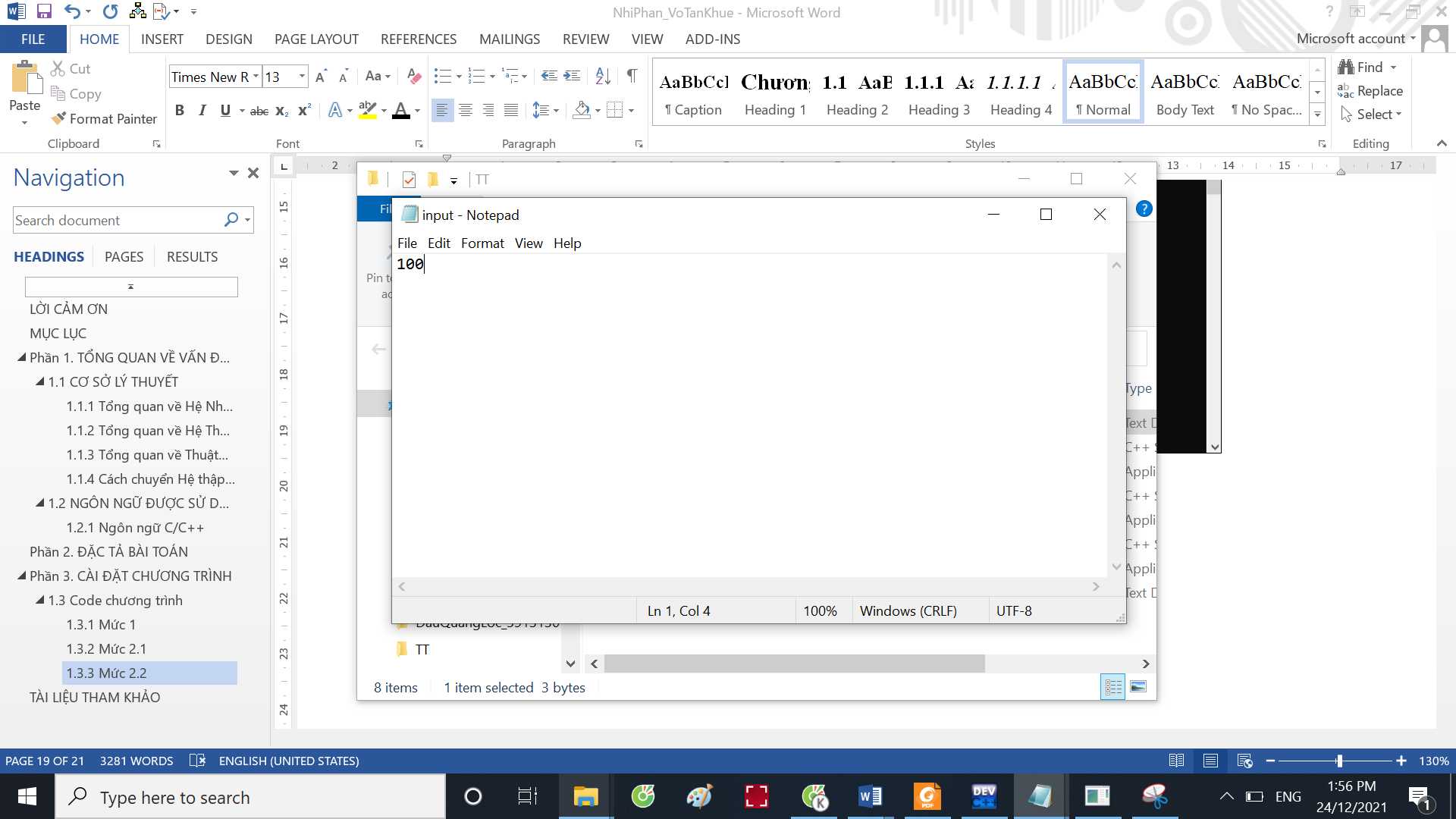
Hình 3.3.1. Nhập vào input n=6

Kết quả trên màn hình Console:



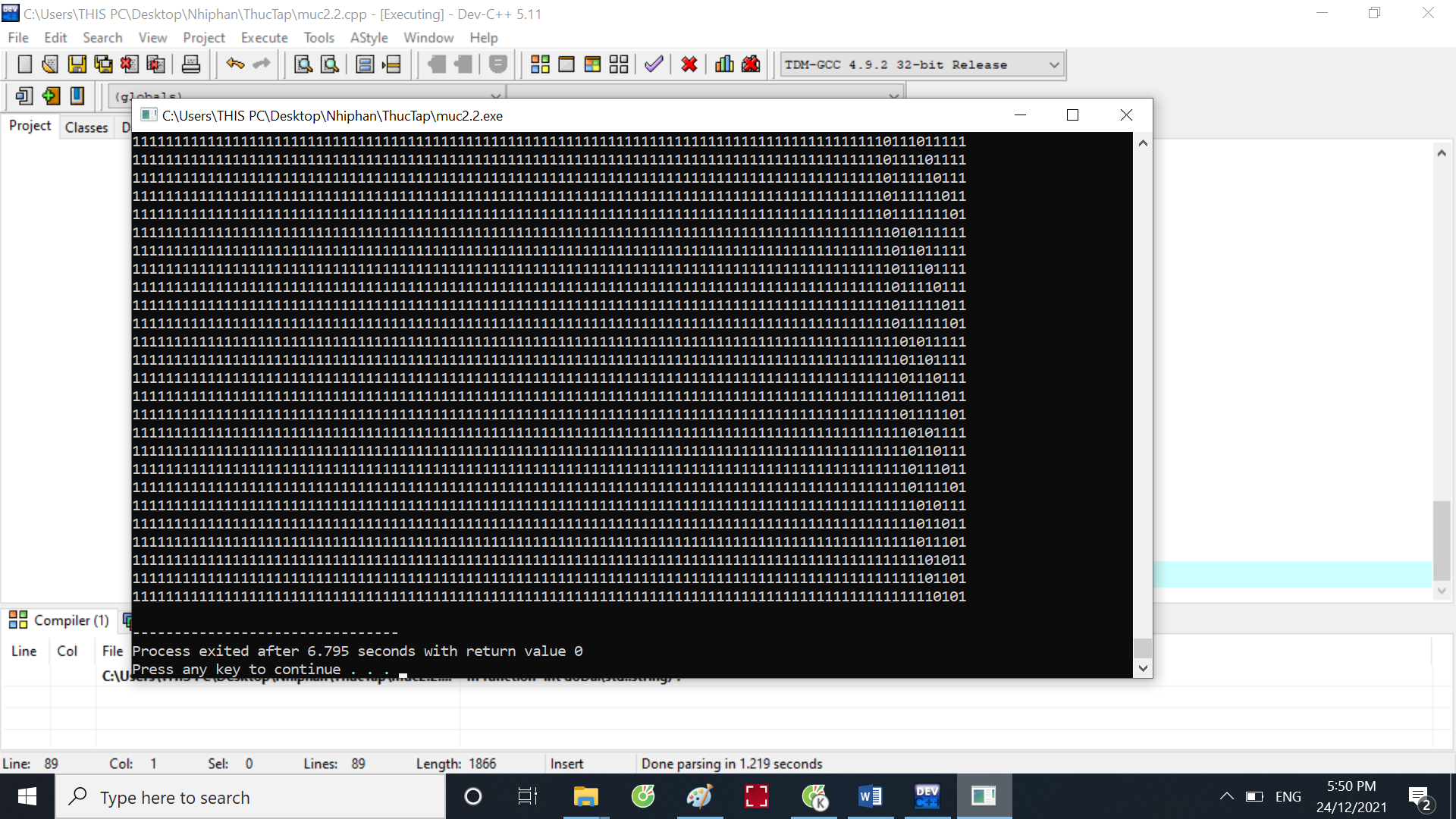
Hình 3.3.2. Kết quả màn hình Console

Kết quả chạy thử chương trình khi cho file input là n = 100 như sau:



Hình 3.3.3. Nhập vào input n=100

Kết quả trên màn hình Console:



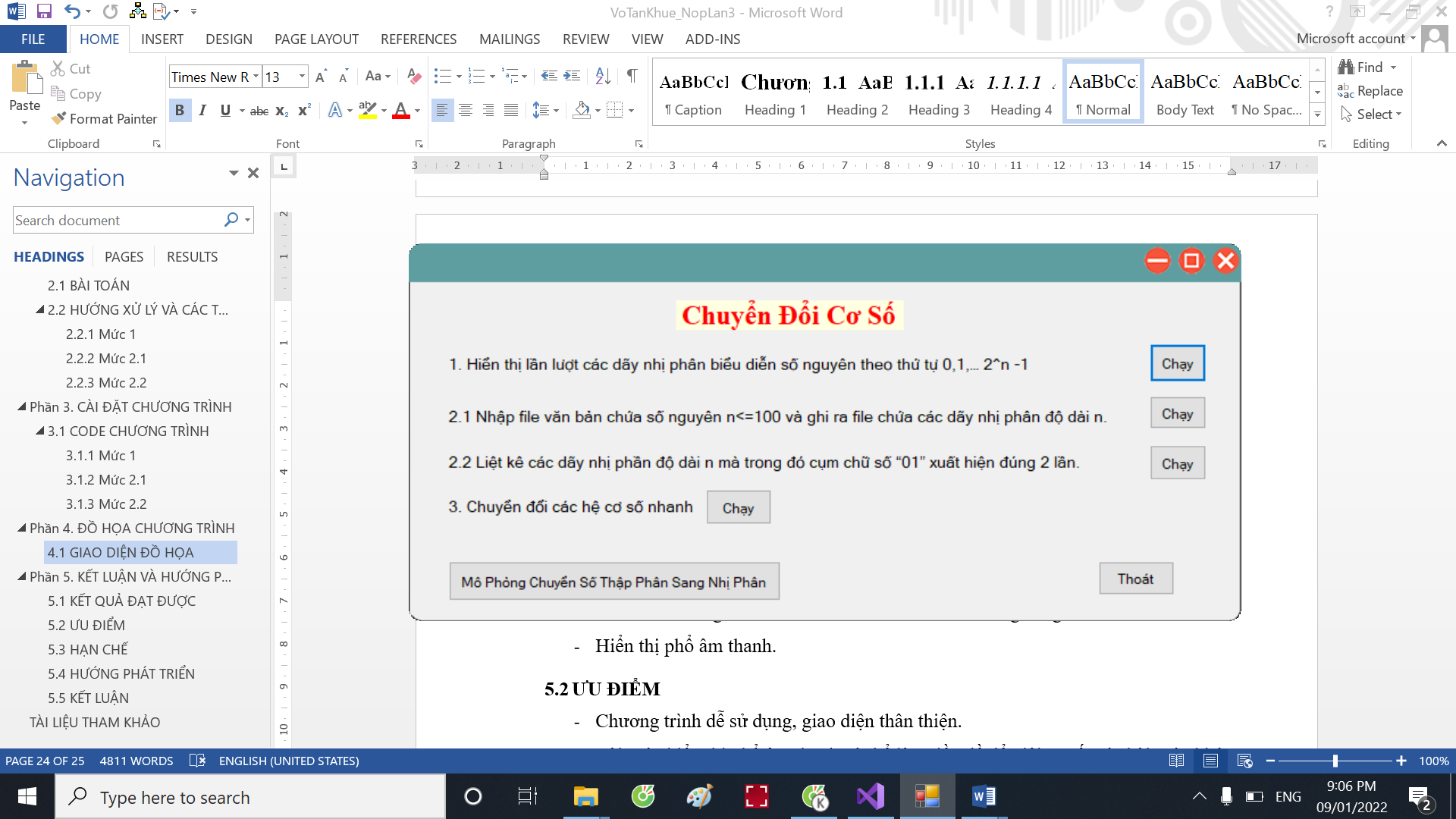
Hình 3.3.4. Kết quả màn hình Console

# Phần 4. ĐỒ HỌA CHƯƠNG TRÌNH

## 4.1 GIAO DIỆN ĐỒ HỌA

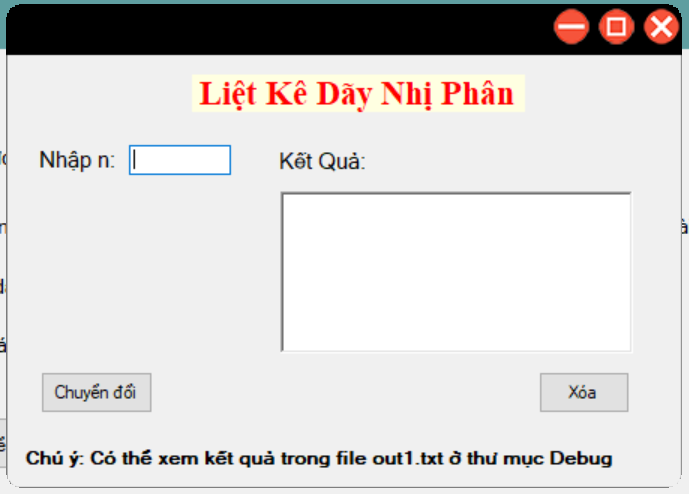
* **Giao diện Menu**

Giao diện Menu hiển thị các nội dung bài toán ở các mức 1, 2.1, 2.2, 3, để chạy các mức người dùng nhấn nút “Chạy”, trong đó mức 3 là mức bổ sung thêm. Ngoài ra có thêm nút hiển thị Mô Phỏng Chuyển Số Thập Phân Sang Nhị Phân.



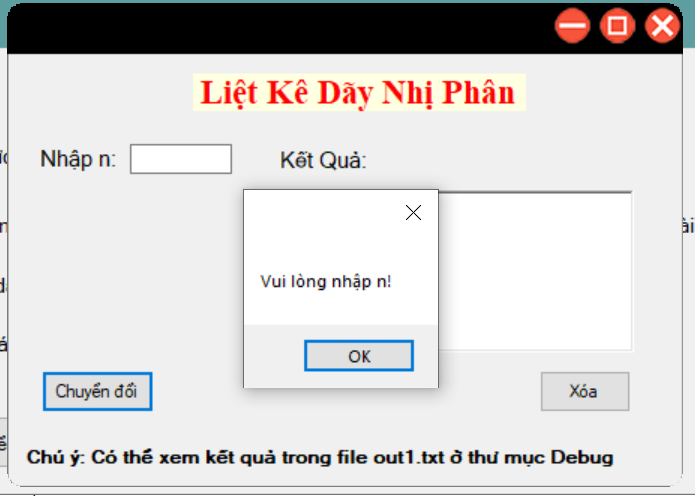
Hình 4.1.1. Giao diện màn hình Menu

* **Giao diện mức 1**



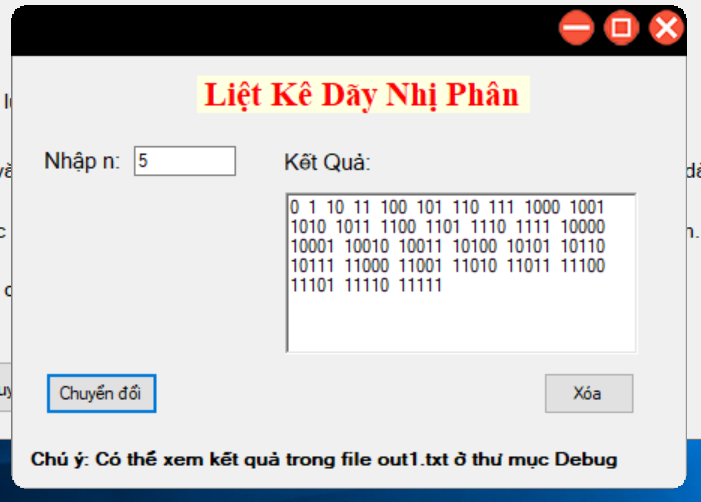
Hình 4.1.2. Giao diện màn hình mức 1

Người dùng khi nhập số nguyên n sẽ được kiểm tra đã nhập dữ liệu hay chưa, trường hợp nếu chưa nhập thì in ra thông báo “Vui lòng nhập n!”.



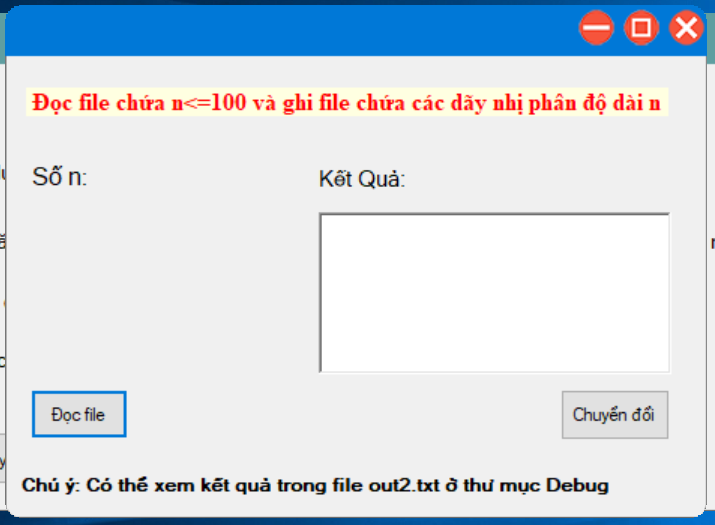
Hình 4.1.3. Kiểm tra khi chưa nhập n

Sau khi người dùng nhập n hợp lệ và nhấn “Chuyển đổi” thì sẽ in kết quả ra màn hình, có thể xem kết quả trong file out1.txt ở thư mục Debug.



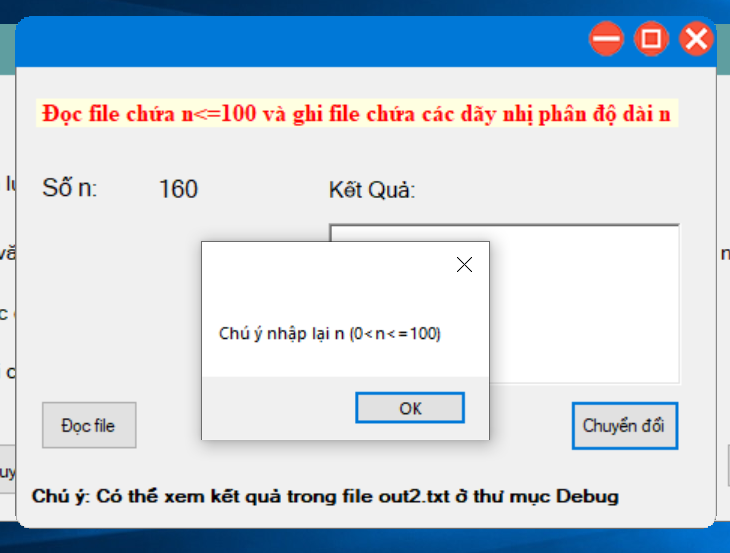
Hình 4.1.4. Kết quả nhập đúng

* **Giao diện mức 2.1**



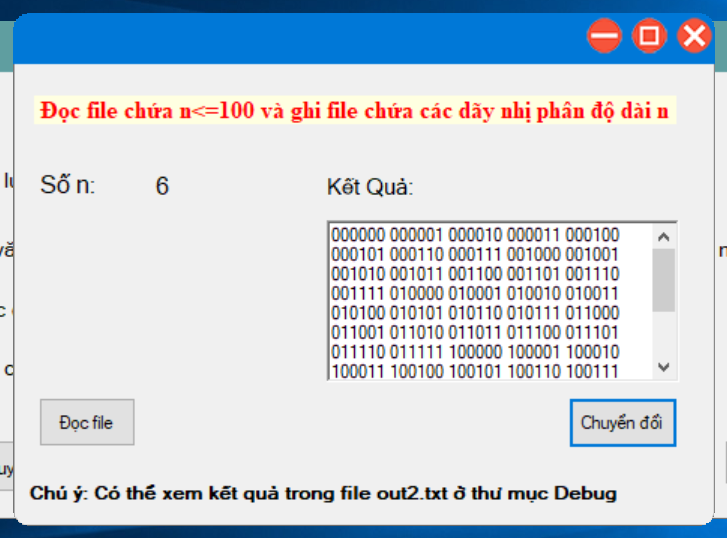
Hình 4.1.5. Giao diện màn hình mức 2.1

Người dùng khi chọn “Đọc file” sẽ chọn file text có chứa số nguyên n, hệ thống sẽ kiểm tra số nguyên n thõa điều kiện hay chưa, nếu sai thì in ra thông báo “Chú ý nhập lại n (0 < n <= 100)”.



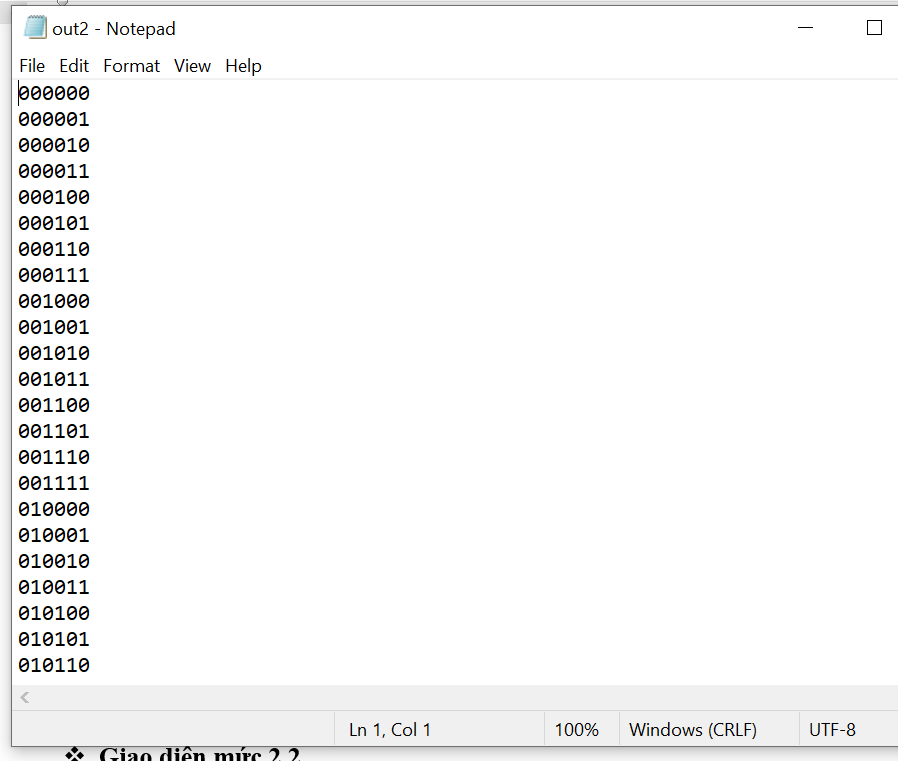
Hình 4.1.6. Kiểm tra điều kiện n

Nếu n thõa điều kiện và người dùng nhấn “Chuyển đổi” thì sẽ in kết quả ra màn hình, có thể xem kết quả trong file out2.txt ở thư mục Debug.



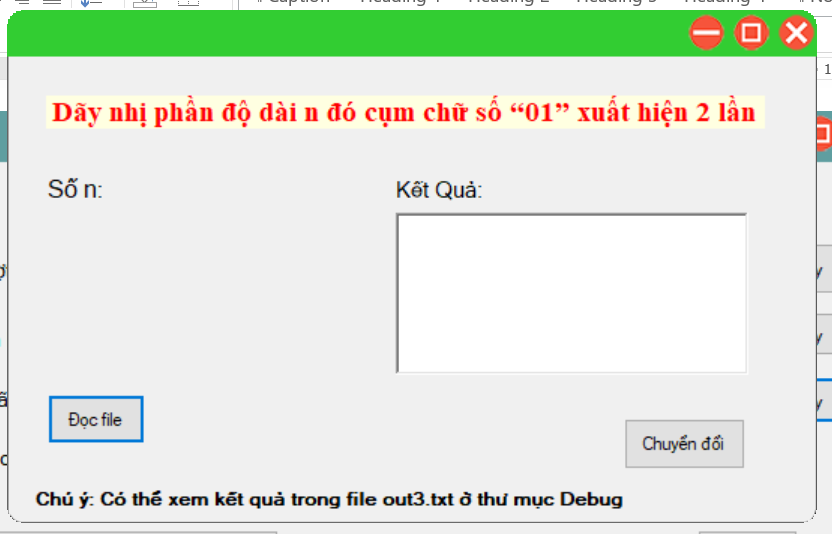
Hình 4.1.7. Kết quả nhập đúng

Kết quả ở file out2.txt ở thư mục Debug



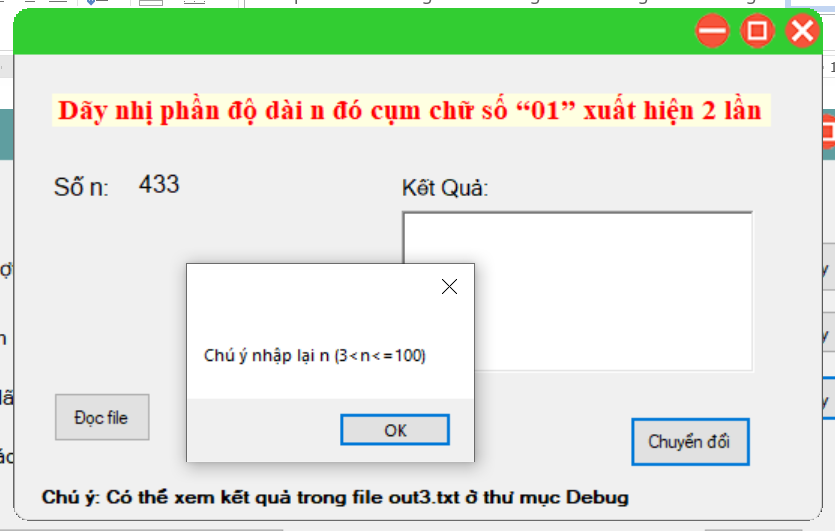
Hình 4.1.8. Kết quả ở file out2.txt

* **Giao diện mức 2.2**



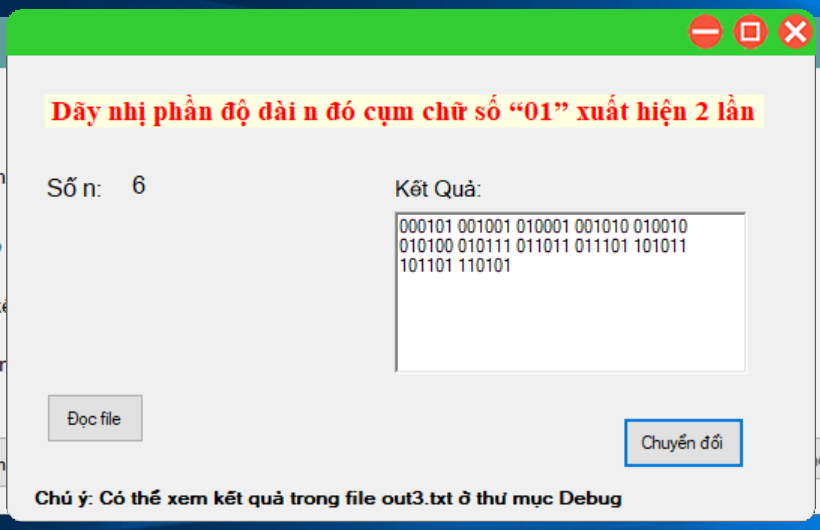
Hình 4.1.9. Giao diện màn hình mức 2.2

Người dùng khi chọn “Đọc file” sẽ chọn file text có chứa số nguyên n, hệ thống sẽ kiểm tra số nguyên n thõa điều kiện hay chưa, nếu sai thì in ra thông báo “Chú ý nhập lại n (0 < n <= 100)”.



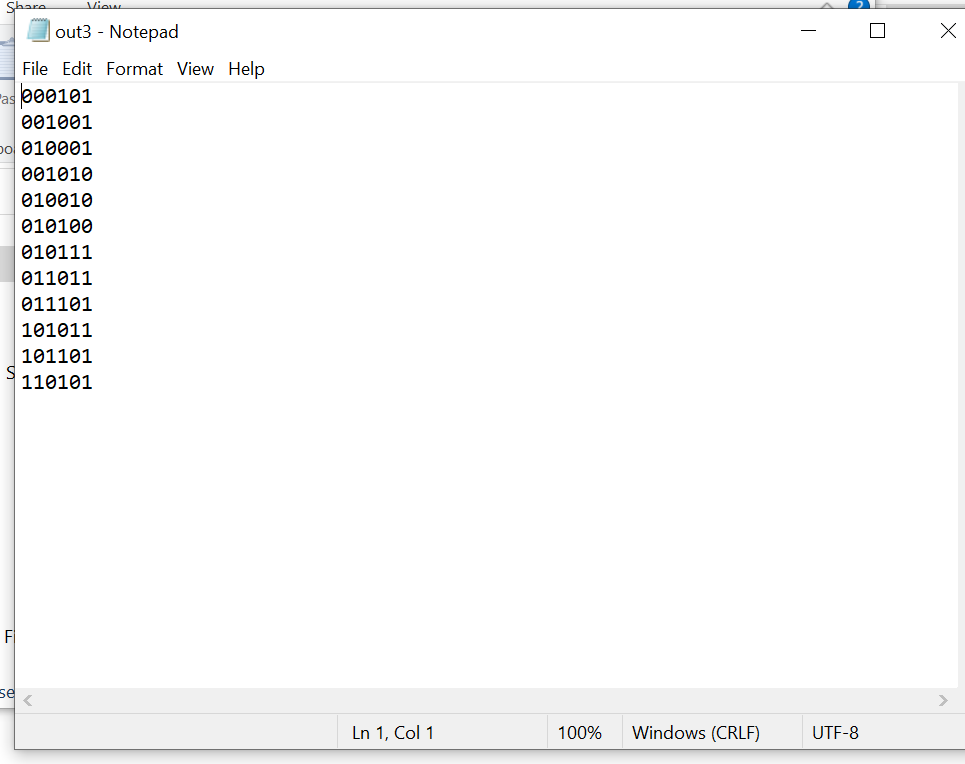
Hình 4.1.10. Kiểm tra điều kiện n

Nếu n thõa điều kiện và người dùng nhấn “Chuyển đổi” thì sẽ in kết quả ra màn hình, có thể xem kết quả trong file out3.txt ở thư mục Debug.



Hình 4.1.11. Kết quả nhập đúng

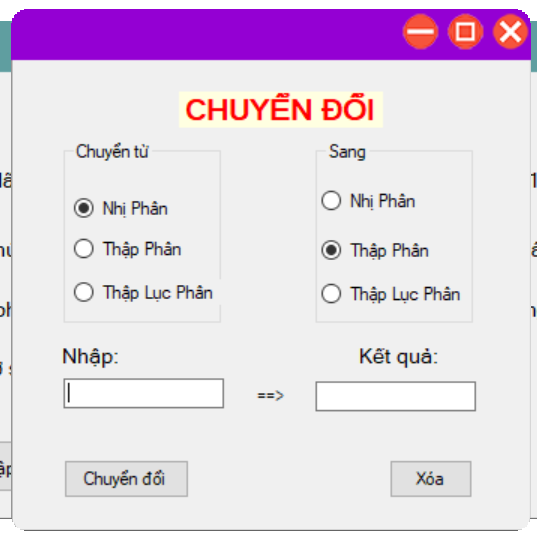
Kết quả ở file out3.txt ở thư mục Debug



Hình 4.1.12. Kết quả ở file out3.txt

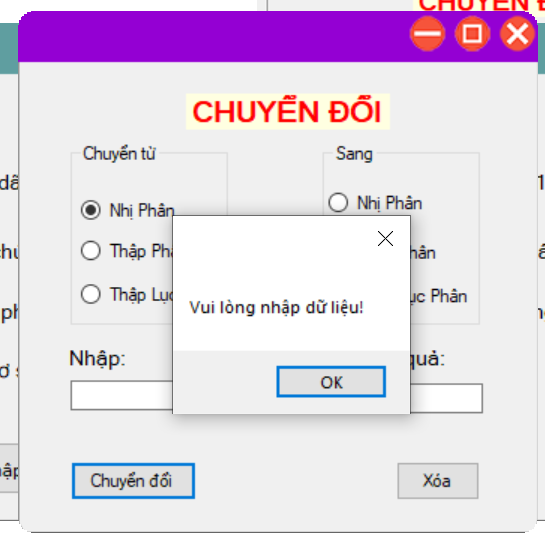
* **Giao diện mức 3 (Làm thêm)**

Mức 3: chuyển đổi các hệ cơ số nhanh bao gồm hệ nhị phân, thập phân, thập lục phân.



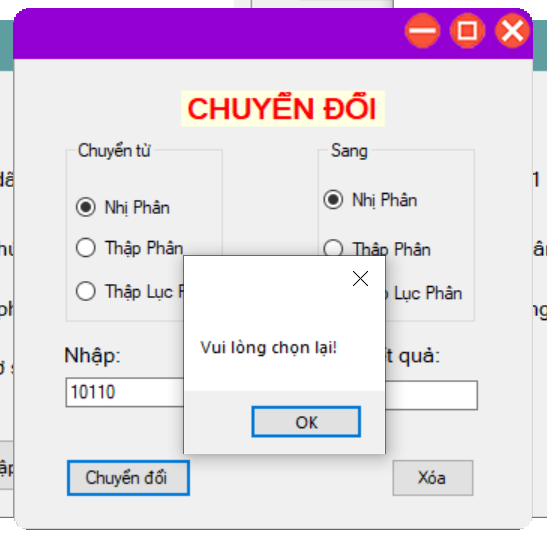
Hình 4.1.13. Giao diện màn hình mức 3

Người dùng khi nhập số nguyên n sẽ được kiểm tra đã nhập dữ liệu hay chưa, trường hợp nếu chưa nhập thì in ra thông báo “Vui lòng nhập dữ liệu!”.



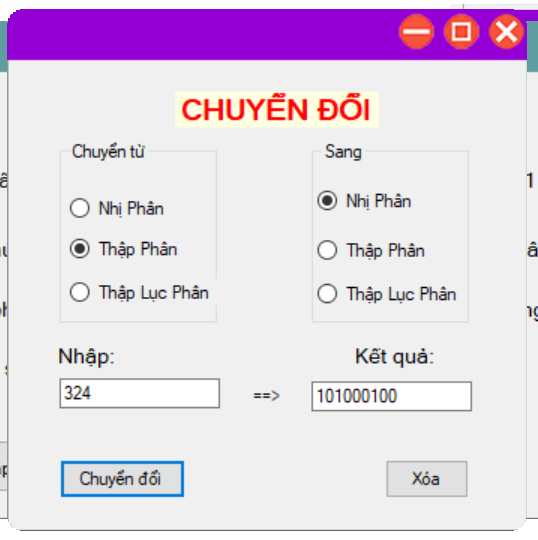
Hình 4.1.14. Kiểm tra khi chưa nhập n

Người dùng khi nhập số nguyên n thõa màn điều kiện, nhưng chọn hai hệ cơ số giống nhau như (Nhi phân – Nhị phân) thì hệ thống in ra thông báo màn hình “Vui lòng chọn lại!”.



Hình 4.1.15. Chọn trùng lại hệ cơ số

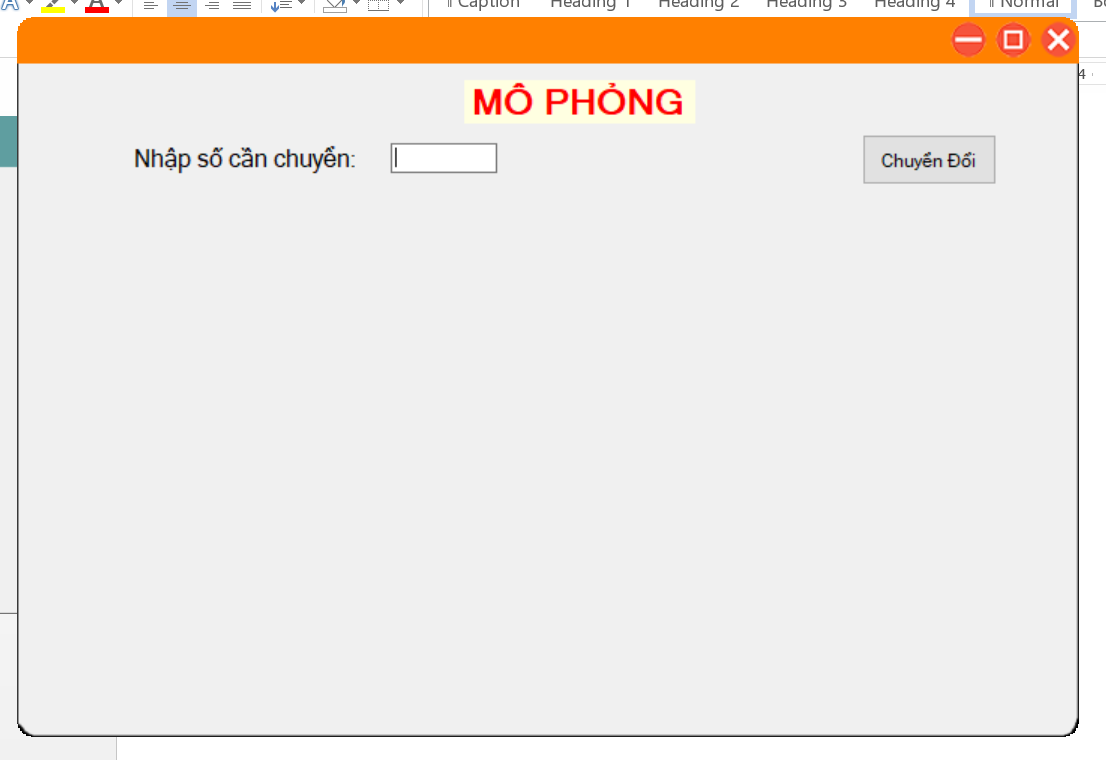
Nếu thõa điều kiện và người dùng nhấn “Chuyển đổi” thì sẽ in kết quả ra màn hình như 4.1.16 bên dưới.



Hình 4.1.16. Kết quả nhập đúng

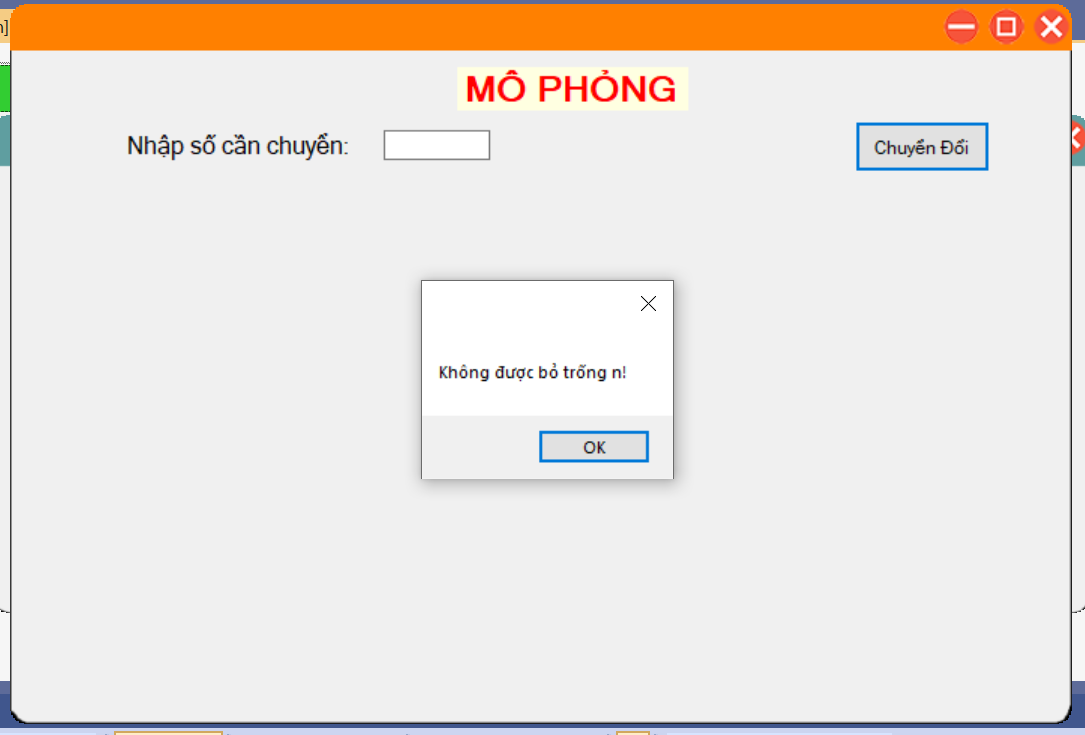
* **Giao diện mô phỏng chuyển số thập phân sang nhị phân**

Giao diện sẽ mô phỏng chuyển số thập phân sang nhị phân dựa trên phương pháp chuyển ở mục 1.1.3.



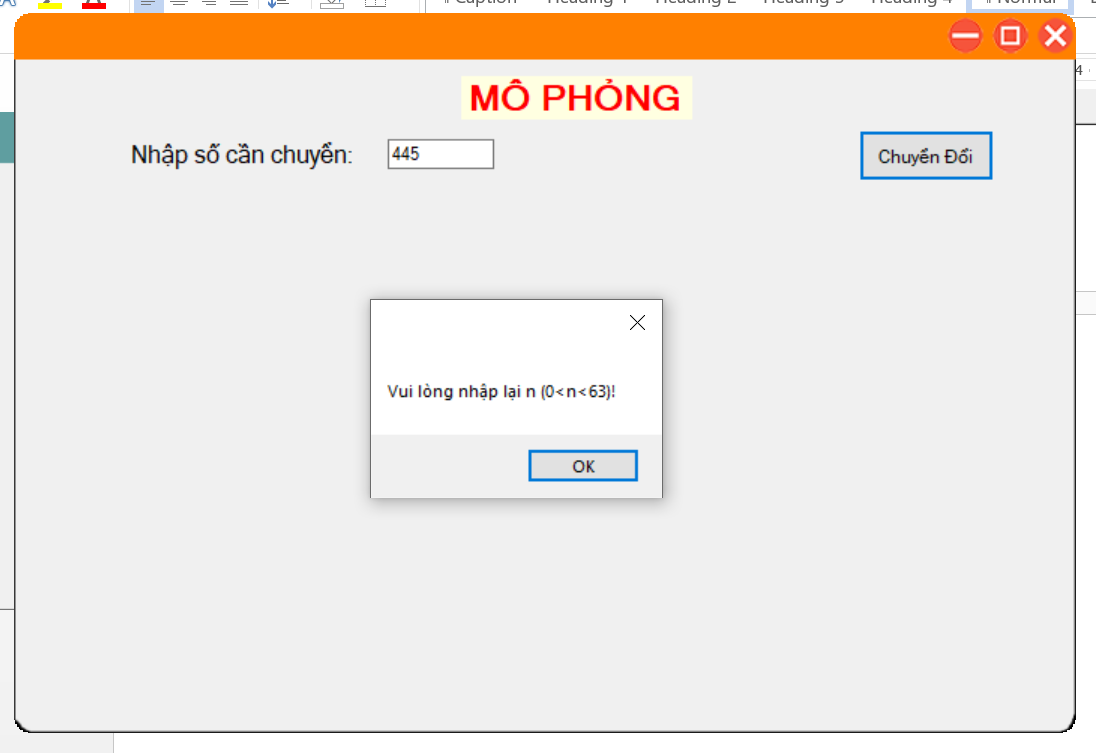
Hình 4.1.17. Giao diện mô phỏng chuyển số thập phân sang nhị phân

Người dùng khi nhập số nguyên n sẽ được kiểm tra đã nhập dữ liệu hay chưa, trường hợp nếu chưa nhập thì in ra thông báo “Không được bỏ trống n!”.



Hình 4.1.18. Kiểm tra khi chưa nhập n

Người dùng khi nhập số nguyên n, hệ thống sẽ kiểm tra số nguyên n thõa điều kiện hay chưa, nếu sai thì in ra thông báo “Chú ý nhập lại n (0 < n <= 63)!”. Vì code giao diện đồ họa chưa tối ứu nên chỉ mô phỏng n từ 0 đến 63.



Hình 4.1.19. Kiểm tra điều kiện n

Nếu n thõa điều kiện và người dùng nhấn “Chuyển đổi” thì sẽ in kết quả ra màn hình như 4.1.20 bên dưới.



Hình 4.1.20. Kết quả nhập đúng

# Phần 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT QUẢ ĐẠT ĐƯỢC

* + Đã thực hiện hoàn chỉnh yêu cầu mức 1, mức 2 của đề.
  + Bổ sung thêm mức 3.
  + Thêm giao diện đồ họa được làm winform, mô phỏng chuyển số thập phân sang nhị phân.

## ƯU ĐIỂM

* + Chương trình dễ sử dụng, giao diện thân thiện.

## HẠN CHẾ

* Giao diện hệ thống chưa được tối ưu cho trải nghiệm người dùng.
* Code chương trình còn dài, còn nhiều chỗ chưa được tối ưu.

## HƯỚNG PHÁT TRIỂN

* Hoàn thiện thêm những chức năng hệ thống hiện có, cải thiện về giao diện, trải nghiệm người.
* Mô phỏng thêm thuật toán đệ quy quay lui và phương pháp sinh.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Wikipedia tiếng Việt, “Hệ nhị phân,” 23 12 2021. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Hệ\_nhị\_phân. [Đã truy cập 23 12 2021]. |
| [2] | Wikipedia tiếng việt, “Hệ thập phân,” 23 12 2021. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Hệ\_thập\_phân. [Đã truy cập 23 12 2021]. |
| [3] | E-Magazine, "Chuyển đổi số từ thập phân sang nhị phân," 05 01 2021. [Online]. Available: https://expressmagazine.net/posts/view/1253/chuyen-doi-so-tu-thap-phan-sang-nhi-phan. [Accessed 05 01 2021]. |
| [4] | Wikipedia tiếng việt, "Thuật toán đệ quy," 23 12 2021. [Online]. Available: ttps://vi.wikipedia.org/wiki/Đệ\_quy\_(tin\_học). [Accessed 23 12 2021]. |
| [5] | Bài giảng chuyên đề, Giải thuật và Lập trình, Đại học Sư phạm Hà Nội: Lê Minh Hoàng, 1999-2002. |
| [6] | Wikipedia tiếng việt, "Dev-C++," 31 12 2021. [Online]. Available: https://vi.wikipedia.org/wiki/Dev-C%2B%2B. [Accessed 21 12 2021]. |
| [7] | Wikipedia tiếng việt, "C++," 23 12 2021. [Online]. Available: https://vi.wikipedia.org/wiki/C%2B%2B. [Accessed 23 12 2021]. |
| [8] | Wikipedia tiếng việt, "C Sharp (ngôn ngữ lập trình)," 09 01 2022. [Online]. Available: https://vi.wikipedia.org/wiki/C\_Sharp\_(ngôn\_ngữ\_lập\_trình). [Accessed 01 01 2022]. |
| [9] | Wikipedia tiếng việt, "Windows Forms," 01 09 2022. [Online]. Available: https://vi.wikipedia.org/wiki/Windows\_Forms. [Accessed 01 09 2022]. |

efrfvr