**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ**

**Đề tài 802**

**LIỆT KÊ TẤT CẢ CÁC SỐ NGUYÊN TỐ KHÔNG VƯỢT QUÁ N CÓ CHIỀU CAO H**

****

**SINH VIÊN THỰC HIỆN:**

**Họ và tên: Hồ Văn Vy**

**MSV: 102190200**

**Lớp: 19TCLC\_DT4**

**Nhóm: 19.13A**

**Giảng viên hướng dẫn: Nguyễn Văn Nguyên**

**Đà Nẵng 06/2020**

LỜI MỞ ĐẦU

Số nguyên tố là số nguyên lớn hơn 1 gồm 2 ước là 1 và chính nó. Với tính chất đặc biệt này khiến số nguyên tố có các ứng dụng quan trọng trong các lĩnh vực mã hóa, bảo mật thông tin đặc biệt là trong thời số, khi mà hầu hết các thông tin đều được tổ chức và lưu trữ thông qua các cloud, server. Do vậy, có rất nhiều bài toán liên quan đến số nguyên tố được đặt ra, và đồ án lần này sẽ nghiên cứu cách tìm những số nguyên tố có chiều cao cho trước trong một khoảng xác định.

Và với đồ án này, trước tiên em xin cảm ơn đến quý cô ThS Đỗ Thị Tuyết Hoa, ThS Trần Hồ Thủy Tiên, và đặc biệt là thầy ThS Nguyễn Văn Nguyên đã tận tình chỉ dẫn giúp em hoàn thành được đồ án.

MỤC LỤC

[1](#_Toc19885203)

[LỜI MỞ ĐẦU 2](#_Toc19885204)

[MỤC LỤC 3](#_Toc19885205)

[DANH MỤC HÌNH VẼ 4](#_Toc19885206)

[1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc19885207)

[2. CƠ SỞ LÝ THUYẾT 5](#_Toc19885208)

[2.1. Ý tưởng 5](#_Toc19885210)

[2.2. Cơ sở lý thuyết 5](#_Toc19885211)

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 6](#_Toc19885212)

[3.1. Phát biểu bài toán 6](#_Toc19885213)

[3.2. Cấu trúc dữ liệu 6](#_Toc19885214)

[3.3. Thuật toán 7](#_Toc19885215)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 9](#_Toc19885216)

[4.1. Tổ chức chương trình 9](#_Toc19885217)

[4.2. Kết quả 10](#_Toc19885218)

[4.2.1. Giao diện chính của chương trình 10](#_Toc19885219)

[4.2.2. Kết quả thực thi của chương trình 14](#_Toc19885220)

[4.2.3. Nhận xét 23](#_Toc19885221)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc19885222)

[a. Kết luận 24](#_Toc19885223)

[b. Hướng phát triển 24](#_Toc19885224)

[TÀI LIỆU THAM KHẢO 25](#_Toc19885225)

[[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản 25](#_Toc19885226)

[[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập 25](#_Toc19885227)

DANH MỤC HÌNH VẼ

Hình 3.1 Sơ đồ khối thuật toán độ cao của một số tự nhiên…………………7

Hình 3.2 Sơ đồ khối thuật toán sàng Eratosthenes…………………………..8

Hình 4.0 Luồng chạy của chương trình……………………………………...9

# Hình 4.1 Giao diện ban đầu, nhấn Enter để vào menu chính……………...10

# Hình 4.2 Chương trình đưa ra 2 lựa chọn cho người dùng………………...11

# Hình 4.3 Người dùng nhập số nguyên N và H……………………………..11

# Hình 4.4 Chương trình đưa ra 2 lựa chọn cho việc xuất kết quả …………..12

# Hình 4.5 Với dữ liệu lớn in ra màn hình chương trình tự động

# phân trang…………………………..……………………………..12

# Hình 4.6 Trường hợp chọn nhập dữ liệu từ file…………………………....13

# Hình 4.7 Người dùng phải nhập tên file đầu vào mong muốn……………..13

# Hình 4.8 Khi người dùng nhập đúng tên file và thỏa mãn………………....14

# Hình 4.9 File output.txt nơi chứa kết quả ………………………………….14

# GIỚI THIỆU ĐỀ TÀI

Với sự phát triển mạnh mẽ của lĩnh vực công nghệ thông tin và mạng Internet, số nguyên tố có các ứng dụng quan trọng trong lĩnh vực mã hóa, bảo mật thông tin số. Vì vậy, nhóm chúng em chọn đề tài 802: viết chương trình C liệt kê các số nguyên tố không có chiều cao là H không vượt quá N ( 10 ≤ N ≤ 106, 1 ≤ H ≤ 54) có sử dụng file.

# CƠ SỞ LÝ THUYẾT

## Ý tưởng

* Sử dụng sàng số nguyên tố Eratosthenes giúp tìm tất cả các số nguyên tố nhỏ hơn một số N cho trước để đảm bảo tốc độ thực thi chương trình, thay vì sử dụng phương pháp đơn giản với tốc độ rất chậm là lặp qua các số trong đoạn [2, N], ở mỗi bước lặp tiến hành kiểm số x (x ϵ [2, N]) có phải là nguyên tố không bằng phương pháp kiểm tra xem x có là bội số của bất số nguyên nào giữa 2 và căn N.
* Dùng thư viện graphics.h giúp tạo giao diện đồ họa hỗ trợ cho việc, nhập xuất dữ liệu từ file hoặc từ bàn phím thuận tiện và trực quan hơn.

## Cơ sở lý thuyết

**2.2.1 Số nguyên tố**

Số nguyên tố là số nguyên lớn hơn 1 và có đúng hai ước là 1 và chính nó.

Do vậy, nếu xác định số x là số nguyên tố, có thể kết luận rằng mọi số nguyên là bội của x đều không phải là số nguyên tố.

**2.2.2 Sàng Eratosthenes**

Sàng Erathosthenes dùng để tìm các số nguyên tố nhỏ hơn hoặc bằng N hoặc có thể dùng để kiểm tra một số nguyên nhỏ hơn hoặc bằng N có phải là số nguyên tố không.

Lý thuyết hoạt động của sàng: tạo 1 danh sách các số tự nhiên từ 2 đến n và giả sử các số đó đều là số nguyên tố. Vào mỗi lần duyệt, chọn một số nguyên tố và đánh dấu tất cả các số là bội của số nguyên tố đó vì không phải là số nguyên tố. Sau khi duyệt xong, các số chưa bị đánh dấu là các số nguyên tố cần tìm.

**2.2.3 Độ cao của một số**

Độ cao của một số tự nhiên là tổng các chữ số của số đó.

Để tìm độ cao của một số số nguyên N, tiến hành lấy số N chia lấy phần dư nhằm lấy chữ số cuối cùng N để tính tổng, sau đó chia số N cho 10 lấy phần nguyên để loại bỏ chữ số cuối cùng vừa cộng; lặp lại như vậy cho đến khi số nguyên N bằng 0.

# TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## Phát biểu bài toán

Độ cao của một số tự nhiên là tổng các chữ số của số đó. Với mỗi cặp số tự nhiên N và H cho trước hãy liệt kê các số nguyên tố không vượt quá N và có chiều cao là H, 10 ≤ N ≤ 106, 1 ≤ H ≤ 54. Hãy viết chương trình C (có sử dụng file) để thực hiện yêu cầu trên.

Đầu vào: Hai số nguyên N và H ( 10 ≤ N ≤ 106, 1 ≤ H ≤ 54) được tùy chọn nhập vào từ bàn phím hoặc từ file

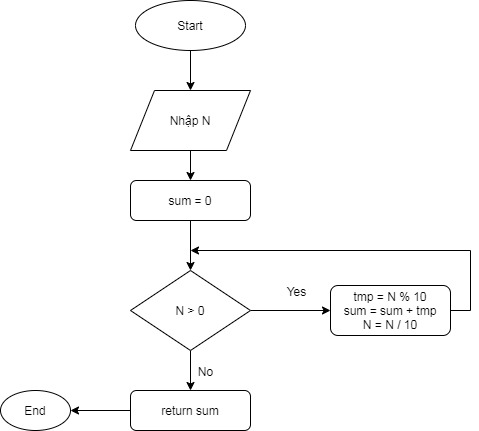
Đầu ra: Danh sách các số nguyên tố trong đoạn [2, N] có chiều cao là H được tùy chọn in trên giao diện đồ họa hoặc file

## Cấu trúc dữ liệu

Đề tài sử dụng danh sách liên kết đặc (mảng một chiều) bằng cách cấp phát động để đánh dấu các số nguyên tố dựa trên thuật toán sàng Eratosthenes.

## Thuật toán

**3.3.2 Độ cao số tự nhiên**



Hình 3.1 Sơ đồ khối thuật toán độ cao của một số tự nhiên

***Psuede code:***

* Nhập N, khởi tạo sum = 0
* Lặp
  + tmp = N % 10
  + sum = sum + tmp
  + N = N / 10

Đến khi N = 0

* Trả về sum: tổng các chữ số của N

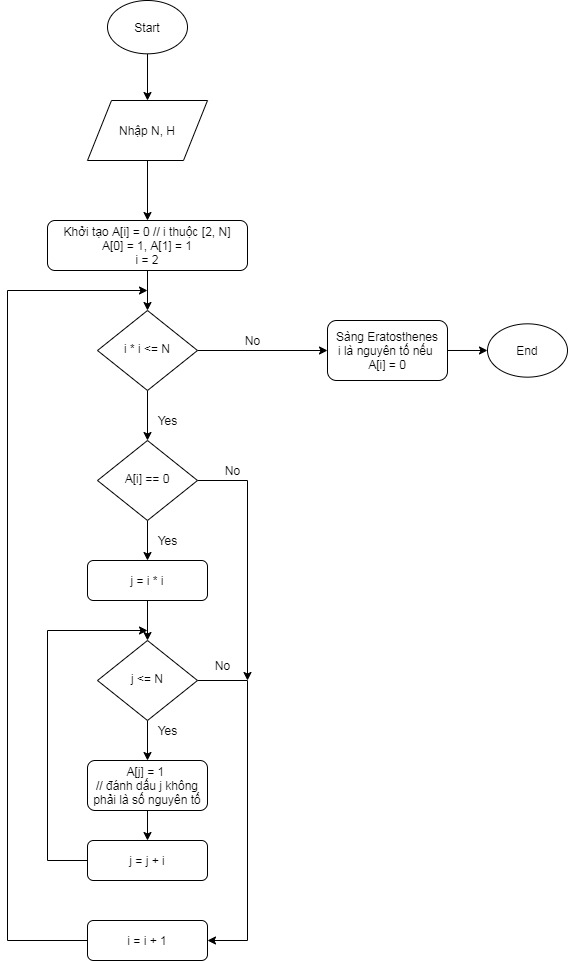
***Độ phức tạp thuật toán***

Giả sử d là số chữ số của N thì độ phức tạp T tỉ lệ với d.

Ta lại có: 10d – 1 ≤ N < 10d ⬄ d – 1 ≤ logN < d ⬄logN < d ≤ logN + 1

* T = O(logN)

**3.3.2 Sàn Eratosthenes**



Hình 3.2 Sơ đồ khối thuật toán sàng Eratosthenes

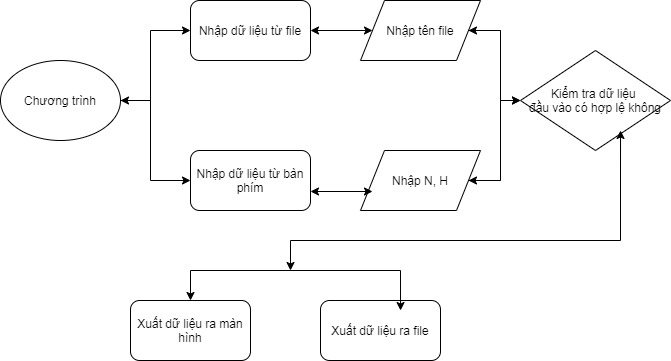
* ***Psude code:***
* Nhập N, H.
* Khởi tạo một mảng A[0] = 1, A[1] = 1, A[i] = 0 với i thuộc [2, N]
* Lặp i = 2; i \* i <= N; i++
  + Nếu A[i] == 0
    - Lặp j = i \* i; j <= N; j = j + i
      * A[j] = 1 // đánh dấu j không phải là số nguyên tố
* Trả về mảng A đã đánh dấu, A[i] == 0 🡺 i là số nguyên tố, A[i] == 1 🡺 i không phải là số nguyên tố
* ***Độ phức tạp thuật toán:***
* Khi i = 2, vòng lặp j lặp N/2 lần
* Khi i = 3, vòng lặp j lặp N/3 lần
* Khi i = 5, vòng lặp j lặp N/5 lần

…..

* Độ phức tạp T = N/2 + N/3 + N /5 + … = N.(1/2 + 1/3 + 1/5 + …) = O(N logN)

# CHƯƠNG TRÌNH VÀ KẾT QUẢ

## Tổ chức chương trình



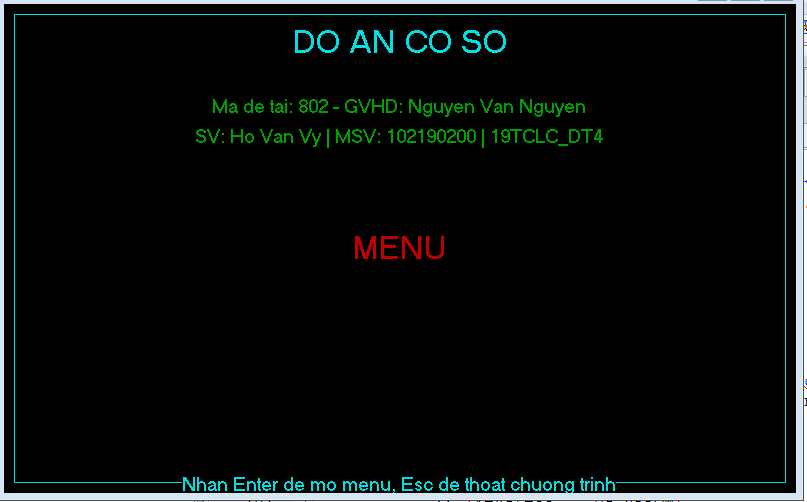
Hình 4.0 Luồng chạy của chương trình

Source code bao gồm:

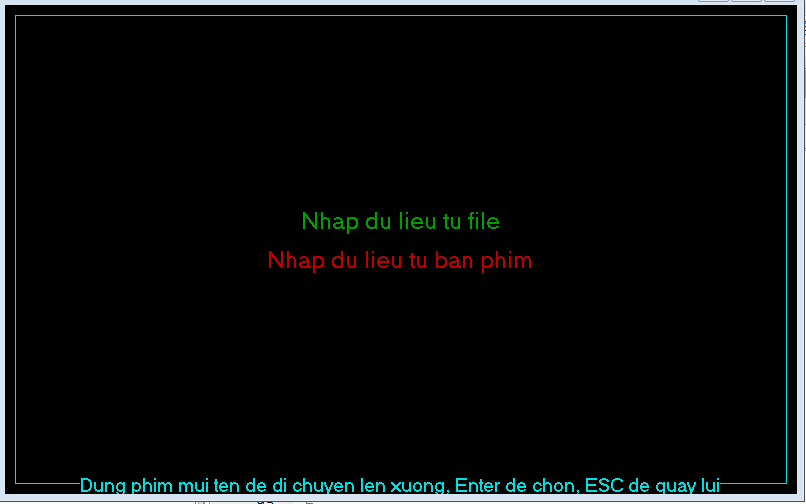
* 3 file header:
* **myConstant.h**: định nghĩa các hằng số về kích thước cửa sổ đồ họa, đường dẫn file output, mã ASCII của các phím Tab, Enter, ESC, mũi tên lên, xuống.
* **myViewFunc.h**: chứa các hàm vẽ khung viền, vẽ các box nhập dữ liệu, viết chữ bằng đồ họa, menu chương trình.
* **calculate-utility.h**: gồm các hàm nhập, xuất file, sàn nguyên tố Eratosthenes, chuyển chuỗi kí tự sang số, chuyển số sang chuỗi kí tự.
* **main.c**: nơi chạy chính của cả chương trình, xử lí các sự kiện bàn phím để hiển thị menu hay gọi các hàm tính toán thích hợp.
* **output.txt**: file chứa kết quả đầu ra của bài toán.

## Kết quả

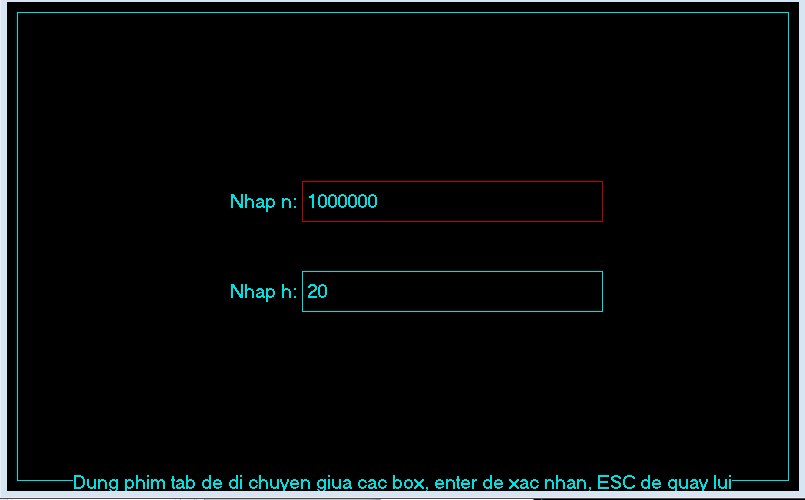
### Giao diện chính của chương trình



Hình 4.1. Giao diện ban đầu, nhấn Enter để vào menu chính

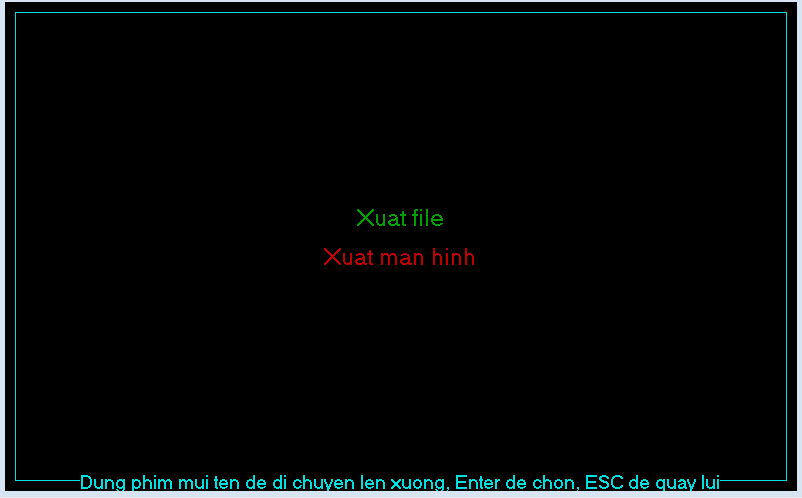


Hình 4.2. Chương trình đưa ra 2 lựa chọn cho người dùng

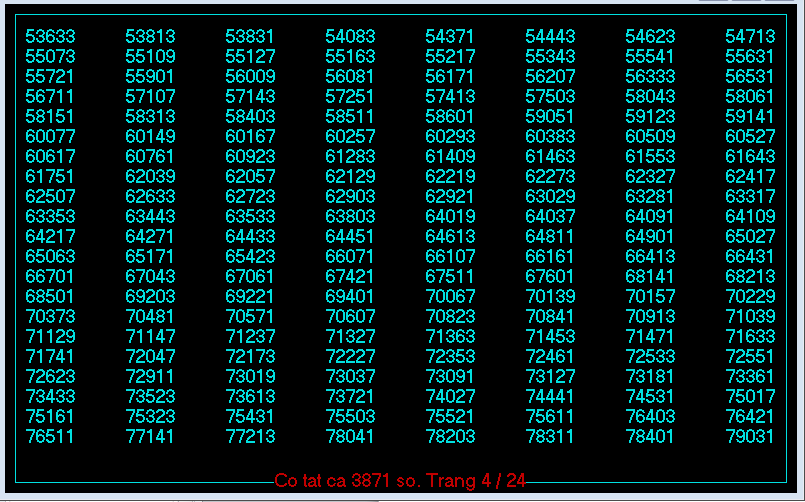


Hình 4.3. Người dùng nhập số nguyên N và H

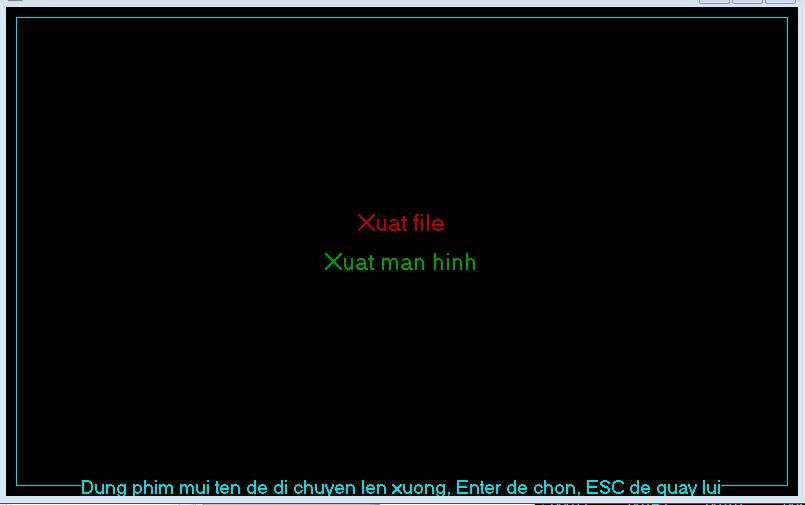
(chương trình có kiểm tra điều kiện đầu vào của N và H)



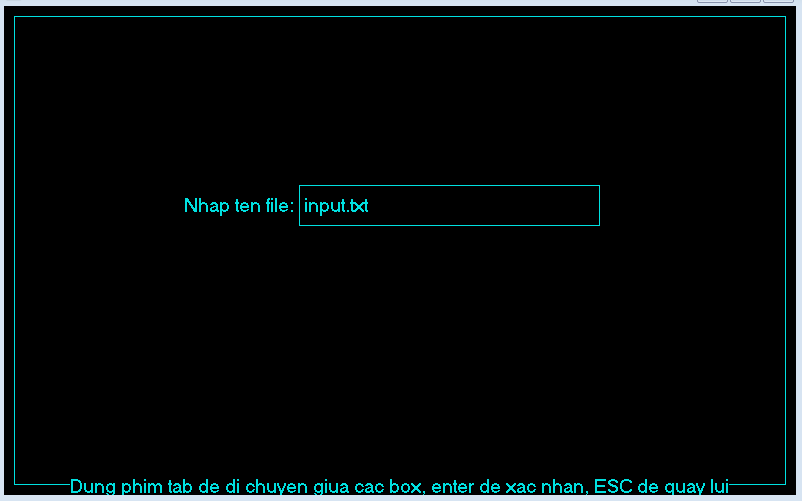
Hình 4.4. Chương trình đưa ra 2 lựa chọn cho việc xuất kết quả



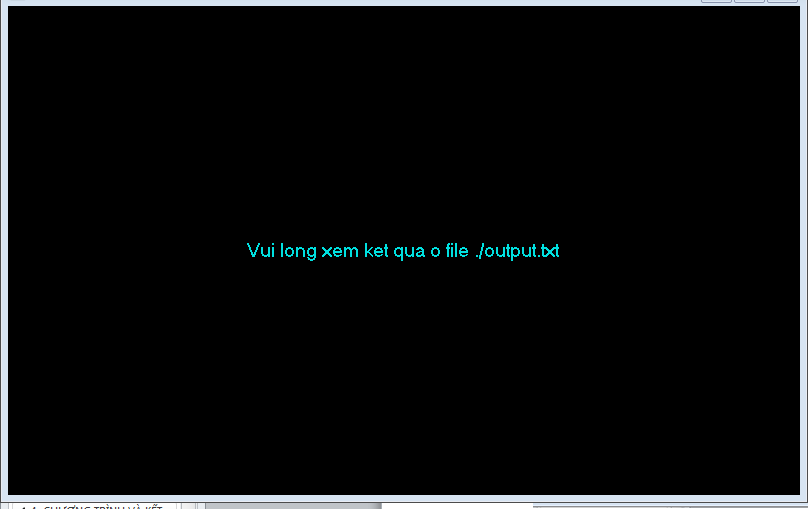
Hình 4.5. Với dữ liệu lớn in ra màn hình được chương trình tự động phân trang

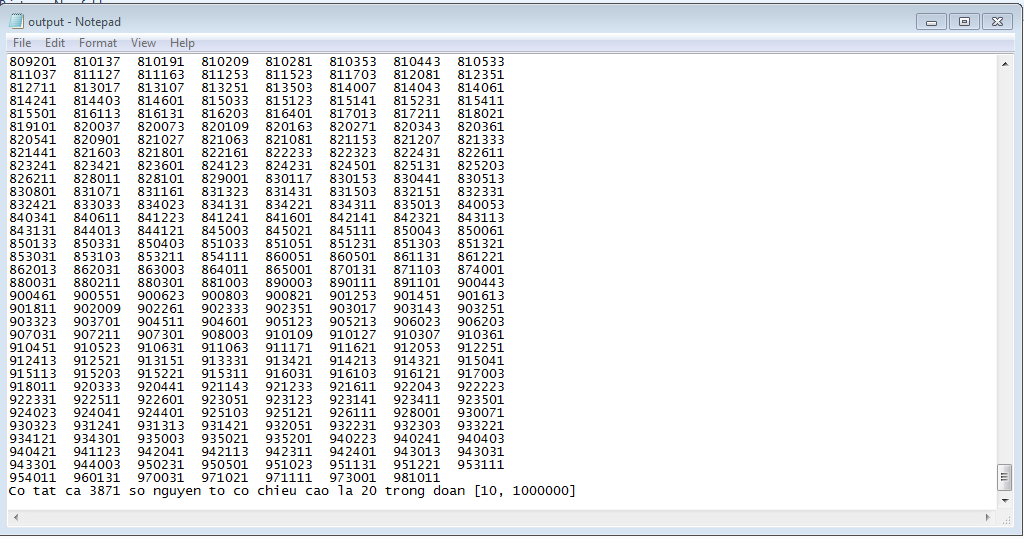


Hình 4.6. Trường hợp chọn nhập dữ liệu từ file



Hình 4.7. Người dùng phải nhập tên file đầu vào mong muốn

Hình 4.8. Khi người dùng nhập đúng tên file và thỏa mãn



Hình 4.9. File output.txt nơi chứa kết quả

### Kết quả thực thi của chương trình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | N | H | Kết quả | Số lượng số nguyên tố có chiều cao H |
| 11 | 1000 | 11 | 29 47 83 137 173 191 227 263 281 317 353 443 461 641 821 911 | 16 |
| 22 | 1000 | 10 | 19 37 73 109 127 163 181 271 307 433 523 541 613 631 811 | 10 |
| 33 | 10000 | 13 | 67 139 157 193 229 283 337 373 409 463 571 607 643 661 733 751 823 1039 1093 1129 1237 1291 1327 1381 1453 1471 1543 1723 1741 1831 2029 2083 2137 2281 2371 2551 2713 2731 2803 3019 3037 3109 3163 3181 3217 3253 3271 3307 3343 3361 3433 3541 3613 3631 4027 4153 4243 4261 4423 4441 4513 4603 4621 4801 5107 5233 5323 5413 5431 5503 5521 5701 6007 6043 6133 6151 6421 7213 7321 7411 8221 8311 9013 9103 | 84 |
| 44 | 10000 | 17 | 89 179 197 269 359 449 467 557 593 647 683 719 773 809 827 863 881 953 971 1097 1187 1259 1277 1367 1439 1493 1583 1619 1637 1709 1871 1907 2069 2087 2267 2339 2357 2393 2447 2591 2609 2663 2753 2843 2861 3167 3257 3329 3347 3491 3527 3581 3617 3671 3761 3833 3851 3923 4049 4139 4157 4229 4283 4337 4373 4391 4409 4463 4481 4517 4643 4733 4751 4931 5039 5147 5237 5273 5309 5381 5417 5471 5507 5651 5741 5813 5903 6029 6047 6173 6263 6317 6353 6551 6803 6911 7019 7109 7127 7253 7307 7433 7451 7523 7541 7703 7901 8009 8081 8117 8171 8243 8423 8513 9161 9323 9341 9413 9431 9521 | 120 |
| 55 | 10000 | 29 | 2999 3989 4799 4889 5879 5897 5987 6599 6689 6779 6869 6959 6977 7499 7589 7877 7949 8597 8669 8849 8867 9479 9497 9587 9677 9749 9767 9839 9857 9929 | 30 |
| 66 | 100000 | 13 | 67 139 157 193 229 283 337 373 409 463 571 607 643 661 733 751 823 1039 1093 1129 1237 1291 1327 1381 1453 1471 1543 1723 1741 1831 2029 2083 2137 2281 2371 2551 2713 2731 2803 3019 3037 3109 3163 3181 3217 3253 3271 3307 3343 3361 3433 3541 3613 3631 4027 4153 4243 4261 4423 4441 4513 4603 4621 4801 5107 5233 5323 5413 5431 5503 5521 5701 6007 6043 6133 6151 6421 7213 7321 7411 8221 8311 9013 9103 10039 10093 10273 10453  ……………………  53401 54013 54121 55021 55201 56101 60133 60223 60331 60601 61051 61141 61231 62131 62311 62401 63031 63103 63211 65011 65101 70051 70123 70141 70321 70501 71023 72031 72103 72211 74101 80221 81013 81031 82003 82021 83101 90031 90121 93001 | 320 |
| 77 | 100000 | 25 | 997 1699 1789 1879 1987 2689 2797 2887 3499 3697 3769 3877 3967 4597 4759 4957 4993 5479 5569 5659 5749 5839 5857 6199 6379 6397 6469 6577 6793 6829 6883 6991 7297 7369 7459 7477 7549 7639 7873 7927 7963 8089 8179 8269 8287 8377 8467 8539 8629 8647 8719 8737 8863 8971 9187 9277 9349 9439 9547 9619 9781 9817 9871 9907  ……………………  93913 94057 94219 94273 94291 94309 94327 94543 94561 94651 94723 94903 95083 95191 95317 95443 95461 95713 95731 95803 95911 96181 96451 96703 97081 97117 97171 97423 97441 97711 98017 98143 98251 98323 99133 99223 99241 | 805 |
| 88 | 1000000 | 23 | 599 797 887 977 1499 1697 1787 1877 1949 2399 2579 2687 2777 2939 2957 3299 3389 3659 3677 3767 3929 3947 4289 4397 4649 4793 4919 4937 4973 5099 5189 5279 5297 5387 5477 5639 5657 5693 5783 5927 5981 6089 6197 6269 6287 6359 6449 6719 6737 6791 6827 6863 6917 6971 7079 7187 7349 7457 7529 7547 7583 7673 7691 7727  ……………………  954221 955103 955211 956003 960017 960053 960251 960341 960521 961133 961151 961241 961313 961511 961601 962033 962051 962303 963311 965201 966011 970043 970061 970133 970313 970421 971051 971141 971501 972113 972131 972221 973031 974003 975011 980321 981023 981221 981311 982103 982211 982301 990023 991031 992021 992111 993011 | 5503 |
| 99 | 1000000 | 17 | 89 179 197 269 359 449 467 557 593 647 683 719 773 809 827 863 881 953 971 1097 1187 1259 1277 1367 1439 1493 1583 1619 1637 1709 1871 1907 2069 2087 2267 2339 2357 2393 2447 2591 2609 2663 2753 2843 2861 3167 3257 3329 3347 3491 3527 3581 3617 3671 3761 3833 3851 3923 4049 4139 4157 4229 4283 4337  ……………………  842111 844001 850121 850211 850301 852011 852101 871001 880001 900143 900161 900233 900701 901007 901133 901403 902141 902303 902501 903311 904103 904121 905111 906011 910421 911033 911231 911303 911321 913013 913103 914021 920123 920411 921203 922211 923201 924101 930113 931013 932003 932021 934001 940031 940301 941201 942101 950111  951101 952001 | 2570 |
| 110 | 1000000 | 20 | 389 479 569 587 659 677 839 857 929 947 983 1289 1487 1559 1667 1847 1973 2099 2297 2459 2477 2549 2657 2693 2729 2819 2837 2909 2927 2963 3089 3359 3449 3467 3539 3557 3593 3719 3863 3881 3917 4079 4259 4349 4457 4493 4547 4583 4637 4673 4691 4817 4871 4943 5087 5393 5483 5519 5573 5591 5717 5807 5843 5861  ……………………  924023 924041 924401 925103 925121 926111 928001 930071 930323 931241 931313 931421 932051 932231 932303 933221 934121 934301 935003 935021 935201 940223 940241 940403 940421 941123 942041 942113 942311 942401 943013 943031 943301 944003 950231 950501 951023 951131 951221 953111 954011 960131 970031 971021 971111 973001 981011 | 3871 |

### Nhận xét

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

…………………………………………………………………………………......

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Chương trình chạy đúng và đảm bảo thời gian đối với các test case số lớn. Ngoài ra, chương trình còn có kiểm tra xử lý lỗi với những trường hợp file không tồn tại, dữ liệu đầu vào không đúng yêu cầu của bài toán.

Chương trình có sử dụng thư viện đồ họa graphics.h giúp thao tác người dùng với chương trình một cách thuận tiện, dễ dàng.

## Hướng phát triển

Phát triển và tìm hiểu các thuật toán khác giúp tính toán các số nguyên tố với kích thước lớn hơn. Có thể dùng các thư viện đồ họa khác như Qt giúp nâng cao trải nghiệm người dùng, dễ tương tác hơn so với graphics.h được sử dụng trong đồ án lần này.

TÀI LIỆU THAM KHẢO

[1] VNOI, *Số học 2 - Số nguyên tố, Sàng Eratosthenes*, [*https://vnoi.info/wiki/translate/he/Number-Theory-2*](https://vnoi.info/wiki/translate/he/Number-Theory-2)

[2] Wikipedia, *Sàng Eratosthenes*, <https://vi.wikipedia.org/wiki/S%C3%A0ng_Eratosthenes>

[3] Nguyễn Văn Hiếu, Đọc ghi file trong C, <https://nguyenvanhieu.vn/doc-ghi-file-trong-c/>