TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**ĐỒ HỌA MÁY TÍNH**

**ĐỀ TÀI: VIẾT CHƯƠNG TRÌNH ĐƯỜNG ELIPSE THEO GIẢI THUẬT MIDPOINT VÀ TÔ MÀU LOANG CHO ELIPSE.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN CẢNH VINH** | |  | **LÊ VĂN LÂM**  **LÊ VĂN HIẾU** | | **Giảng viên hướng dẫn** | **: NGÔ QUỐC TẠO** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: TÍN CHỈ D13CNPM2** | | | **Khóa** | **: 2018-2023** | | |  |
|  |  |

***Hà Nội, tháng 6 năm 2020***

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Nguyễn Cảnh Vinh |  |  |  |
| 2 | Lê Văn Lâm |  |  |  |
| 3 | Lê Văn Hiếu |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

**LỜI CẢM MƠN**

Trên thực tế, không có sự thành công nào mà không gắn liền với những sự hỗ trợ, sự trợ giúp đỡ ít hay nhiều, dù là trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường Đại học đến nay em đã nhận được rất nhiều sự quan tâm, giúp đỡ của thầy cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến thầy cô ở Khoa Công Nghệ Thông Tin – Trường đại học Điện Lực đã cùng với tri thức và tâm huyết của mình để tuyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường và đặc biệt, trong kỳ này, em được tiếp cận với môn học rất hữu ích đối với sinh viên ngành Công Nghệ Thông Tin. Đó là môn: “*Đồ họa máy tính*”

Em xin chân thành cảm ơn thầy Ngô Quộc Tạo đã tận tâm hướng dẫn chúng em qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về môn học. Trong thời gian được học tập và thực hành dưới sự hướng dẫn của thầy, em không những thu được rất nhiều kiến thức bổ ích, mà còn được truyền sự say mê và thích thú đối với bộ môn “*đồ họa máy tính*”. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ báo cáo này rất khó có thể hoàn thành được.

Xin gửi lời cảm ơn chân thành đến gia đình, bạn bè là nguồn động viên to lớn, giúp em vượt qua những khó khăn trong quá trình học tập và thực hiện báo cáo.

Mặc dù đã rất cố gắng hoàn thiện báo cáo với tất cả sự nỗ lực, tuy nhiên, do bước đầu đi vào thực tế, tìm hiểu và xây dựng báo cáo trong thời gian có hạn, và kiến thức còn hạn chế, nhiều bỡ ngỡ, nên báo cáo *“*Vẽ hình elipse bằng thuật toán Midpoint, tô màu loang cho elip*”* chắc chắn sẽ không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự quan tâm, thông cảm và những đóng góp quý báu của các thầy cô và các bạn để báo cáo này được hoàn thiện hơn.

Một lần nữa, em xin chân thành cảm ơn và luôn mong nhận được sự đóng góp của mọi người.

Trân trọng.

**LỜI NÓI ĐẦU**

Trong xu thế tin học hóa toàn bộ cuộc sống đang diễn ra tấp nập hiện nay, lĩnh vực đồ họa đóng góp một vai trò cực kỳ to lớn. Ở đâu ta cũng có thể tìm thấy những minh họa cho việc áp dụng đồ hạo máy tính vào cuộc sống: xử lý ảnh, phim hoạt hình, nghiên cứu, xây dựng và tập hợp các công cụ (mô hình lý thuyết và phần mềm) khác nhau để *kiến tạo, xây dựng, lưu trữ và xử lý* các *mô hình* và *hình ảnh* của các đối tượng, sự vật, hiện tượng trong cuộc sống, sản xuất, nghiên cứu.

Đồ Họa Máy Tính góp phần quan trọng làm cho giao tiếp giữa con người và máy tính trở nên thân thiện hơn. Trong lĩnh vực kỹ thuật máy tính, Đồ Họa Máy Tính càng ngày càng phát triển mạng mẽ. Từ đồ họa trên máy tính chúng ta có nhiều lĩnh vực có ứng dụng rất quan trọng của Đồ Họa Máy Tính trong thực tế như: tạo mô hình, hoạt cảnh (game, giải trí,…), hỗ trợ thiết kế đồ họa, mô phỏng hình ảnh, chuẩn đoán hình ảnh (trong Y tế), huấn luyện đào tạo ảnh (quân sự, hàng không,…)

Trong Đồ Họa Máy Tính có nhiều thuật toán khác nhau. Xong mỗi thuật toán lại tỏ ra có những ưu việt và hạn chế riêng đối với từng bài toán cụ thể. Và để phục vụ cho việc tìm hiểu thêm về các thuật toán trong Đồ Họa Máy Tính, Nhóm chúng em với sự hướng dẫn của Giảng viên – – Khoa CNTT – Trường ĐH Điện Lực đã thực hiện đề tài: *““*Vẽ hình elipse bằng thuật toán Midpoint, tô màu loang cho elip*”.*

Chúng em xin gửi lời cảm ơn chân thành tới thầy giáo: *“***NGÔ QUỐC TẠO”** đã tận tâm chỉ bảo và giúp đỡ chúng em hoàn thành đề tài này. Trong quá trình làm đề tài sẽ không tránh khỏi những sai sót, khuyết điểm. Vì vậy, nhóm thực hiện chúng em hy vọng nhận được sự đánh giá và đóng góp nhiệt tình từ phía thầy cô và các bạn để bài của nhóm chúng em được hoàn thiện hơn

***Chúng em xin chân thành cảm ơn!***

Trường ĐH Điện Lực, tháng 06 năm 2020.

**PHẦN 1: GIỚI THIỆU CHUNG VỀ ĐỀ TÀI**

## Giới Thiệu Đề Tài.

Cùng với sự phát triển của nền kinh tế hiện nay là sự phát triển mạnh mẽ của ngành công nghệ thông tin. Các ứng dụng của công nghệ thông tin ngày càng nhiều và có mặt trong hầu hết các lĩnh vực của cuộc sống. Với đề tài: *“*Vẽ hình elipse bằng thuật toán Midpoint, tô màu loang cho elip *”*, nhóm Vinh + Hiếu+ Lâm – lớp D13CNPM2 sẽ trình bày về thuật toán Midpoint. Ý nghĩa của thuật toán Bresham trong công nghệ thông tin là không hề nhỏ, đặc biệt là đối với màn hình máy tính thì nó có ý nghĩa vô cùng quan trọng.

Qua đề tài này nhóm, mong sẽ giúp ích được cho mọi người hiểu thêm phần nào về thuật toán Bresham.

1. **Giới Thiệu Ngôn Ngữ Lập Trình Được Sử Dụng Trong Đề Tài**

Stroustrup đã bắt đầu làm việc với khái niệm lớp trong [1979](http://vi.wikipedia.org/wiki/1979). Ý tưởng tạo ra một ngôn ngữ mới bắt nguồn từ kinh nghiệm lập trình khi mà ông viết luận án tiến sĩ. Stroustrup nhận ra rằng [Simula](http://vi.wikipedia.org/w/index.php?title=Simula&action=edit&redlink=1)có nhiều tính năng hữu dụng cho việc phát triển một phần mềm lớn nhưng nó đã quá chậm trong ứng dụng thực tế, trong khi đó, [BCPL](http://vi.wikipedia.org/w/index.php?title=BCPL&action=edit&redlink=1) lại nhanh nhưng ở cấp quá thấp và không tiện cho việc phát triển phần mềm lớn. Đến khi làm việc ở [Bell Labs](http://vi.wikipedia.org/w/index.php?title=Bell_Labs&action=edit&redlink=1), thì ông gặp phải vấn đề trong việc phân tích [nhân Unix](http://vi.wikipedia.org/w/index.php?title=Nh%C3%A2n_Unix&action=edit&redlink=1) với việc [tính toán phân tán](http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n_ph%C3%A2n_t%C3%A1n). Dùng lại kinh nghiệm lúc làm luận án tiến sĩ, Stroustrup cài thêm các tính năng giống như Simula vào trong C để nâng cao. C được chọn là vì đó là ngôn ngữ tổng quát, nhanh và năng động. Lần đầu tiên, các chức năng như là lớp, lớp dẫn xuất, kiểm tra kiểu mạnh, nội tuyến (inline), và đối số mặc định đã được thêm vào trong C. Lần xuất bản đầu tiên vào thị trường xảy ra trong tháng 11/[1985](http://vi.wikipedia.org/wiki/1985).

Năm [1983](http://vi.wikipedia.org/wiki/1983), thì tên *C với các lớp* được đổi thành C++. các chức năng mới được thêm vào bao gồm hàm ảo, quá tải hàm và toán tử, tham chiếu, hằng, khả năng kiểm soát bộ nhớ của lưu trữ tự do, nâng cao việc kiểm soát kiểu, và lệnh chú giải kiểu (//).

Năm [1985](http://vi.wikipedia.org/wiki/1985), tác phẩm *The C++ Programming Language* được xuất bản lần đầu tiên, cung cấp một tài liệu tham khảo quan trọng cho ngôn ngữ nhưng đó chưa là một tiêu chuẩn chính thức.

Năm [1989](http://vi.wikipedia.org/wiki/1989) phiên bản C++ 2.0 phát hành. Các tính năng mới bao gồm đa kế thừa, lớp trừu tượng, hàm tĩnh, hàm thành viên hằng, và thành viên bảo tồn.

Năm [1990](http://vi.wikipedia.org/wiki/1990), cuốn *The Annotated C++ Reference Manual* được xuất bản cung cấp nền tảng cho tiêu chuẩn tương lai.

Phiên bản xuất bản sau đó có thêm các chức năng tiêu bản, ngoại lệ, không gian tên, chuyển kiểu cho toán tử new, và kiểu Boolean.

Khi C++ hình thành, thì thư viện chuẩn hoàn thiện với nó. Thư viện C++ đầu tiên thêm vào là iostream.h cung cấp cơ sở để thay thế các hàm C truyền thống như là printf và scanf. Sàu này, trong những thư viện chuẩn quan trọng nhất được thêm vào là [Thư viện Tiêu bản Chuẩn](http://vi.wikipedia.org/w/index.php?title=Th%C6%B0_vi%E1%BB%87n_Ti%C3%AAu_b%E1%BA%A3n_Chu%E1%BA%A9n&action=edit&redlink=1).

Sau nhiều năm làm việc, có sự cộng tác giữa [ANSI](http://vi.wikipedia.org/wiki/ANSI) và hội đồng tiêu chuẩn hoá C++ của [ISO](http://vi.wikipedia.org/wiki/ISO) để soạn thảo tiêu chuẩn [*ISO/IEC 14882*](http://vi.wikipedia.org/w/index.php?title=ISO/IEC_14882&action=edit&redlink=1)*:1998*. Phiên bản tiêu chuẩn này được phát hành năm [1989](http://vi.wikipedia.org/wiki/1989), hội đồng tiếp tục xử lí các báo cáo trục trặc, và ấn hành một phiên bản sửa sai của chuẩn C++ trong năm 2003.

Không ai là chủ nhân của ngôn ngữ C++, nó hoàn toàn miễn phí khi dùng. Mặc dù vậy, các văn bản tiêu chuẩn thì không miễn phí.

## III. Giới Thiệu Thư Viện Graphics.h Trong Dev-C++.

Tạo một chương trình đồ hoạ 2D trong DOS sữ dụng turbo C khá dễ dàng và ai cũng có thể làm được.Thư viện đựoc #include có tên là graphis.h. Nhưng các thư viện C/C++ của borland lại có 1 điều đặc biệt là không sữ dụng được đối với 1 số trình compiler khác, tuy nhiên so với các IDE hiện đại thì borland lại kém về mặt quản lý và tính thuận tiện.

Chính vì lẽ đó mà bản hack thư viện graphics cũng như các thư viện khác ra đời, và hôm nay thư viện mình muốn giới thiệu graphics.h của Micheal.

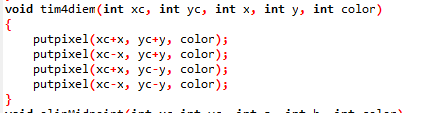
Micheal đã thay đỗi BGI library (thư viện BGI) thành thư viện có tên WinBGIm để có thể sữ dụng tốt trên windows. Và bây giờ bạn đã có thể sữ dụng tốt các hàm đặc biệt của borland bằng dev-c++

**IV.** **Cách sử dụng thư viện đồ họa trong Dev-C++.**

Đầu tiên bạn tải file đính kèm về máy mình, giải nén ra sẽ được 2 file là graphics.h và libbgi.a  
+ File graphics.h để ở thư mục C:\Dev-Cpp\include  
+ File libbgi.a để ở thư mục C:\Dev-Cpp\lib  
(Nếu bạn cài mặc đinh Dev C++ ở ổ C)  
  
Tiếp theo bạn khởi động Dev C++ lên, vào File-->New--> Project...-->Empty Project (Nhớ chọn C++ Project) -->OK  
Đặt tên cho Project của mình!  
  
Nhấn chuột phải lên cái project của bạn -->New File  
Nhấn Alt + P --->Chọn thẻ Parameters--> Copy và paste vào ô Linker những dìng dưới đây:  
-lbgi  
-lgdi32  
-lcomdlg32  
-luuid  
-loleaut32  
-lole32  
(Bao gồm cả dấu "-")  
-->OK

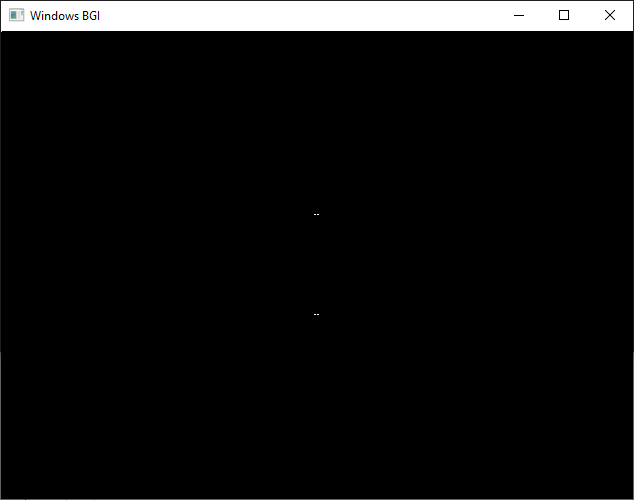
**Phần 2: Ý TƯỞNG XÂY DỰNG CHƯƠNG TRÌNH VÀ CÀI ĐẶT THUẬT TOÁN**

1. **CÁC HÀM SỬ DỤNG TRONG BÀI TẬP :**
   1. **TÌM 4 ĐIỂM**Code :



Trong phần ý tưởng của bài toán, để vẽ được hình elip hoàn hỉnh, chúng ta sẽ chia hình elip thành 4 phần. Và từ 1 phần để vẽ ra 3 phần còn lại.   
 Công dụng của hàm trên là để vẽ ra 4 điểm đối xứng nhau trên hình elip.

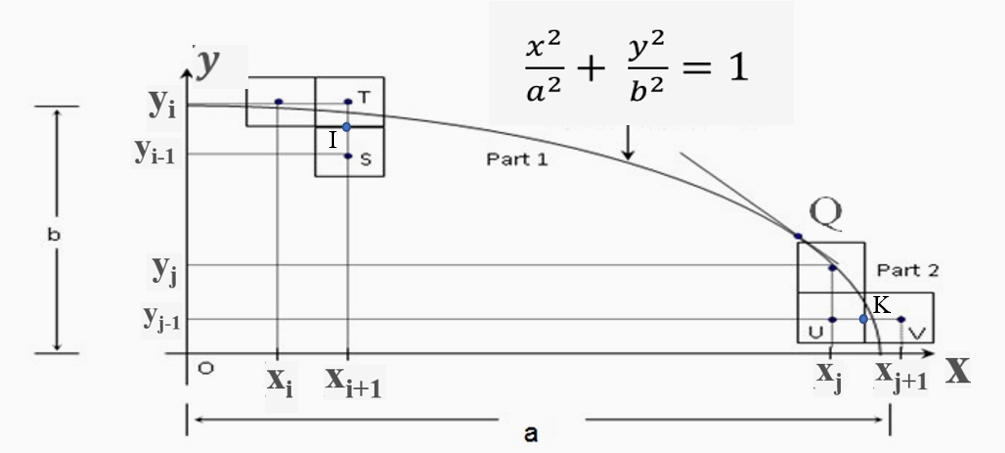
Ví dụ : khi ta tìm ra được 1 điểm trên hình elip . thì khi qua hàm trên, máy tính sẽ hiển thị tiếp 3 điểm đối xứng qua các trục Ox, Oy và đối của điểm đó qua tâm O.



*Hình 1 : Hàm Vẽ Hình Ellipse bằng thuật toán Mid-Point*

**2. Ý TƯỞNG THUẬT TOÁN :**

* Chia hình elip thành 2 phần tại tiếp điểm Q có hệ số góc của tiếp tuyến với elip bằng -1

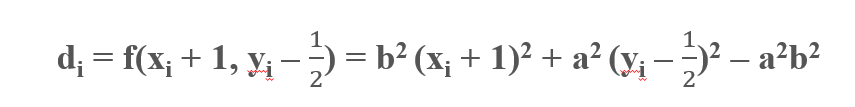


Tại vùng thứ 1, x biến thiên nhanh hơn y  
Tại vùng thứ 2, y biến thiên nhanh hơn x

1. Trong vùng thứ 1

Giả sử, ta vẽ được tọa độ (xi, yi), điểm tiếp theo trên hình elip được chọn trong bước nhảy i+1 là T hoặc là S.   
Trung điểm I của TS sẽ quyết định điểm nào được chọn (T hoặc S).

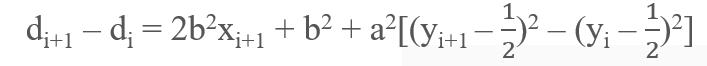
Giá trị của f(x,y) tại điểm I là :

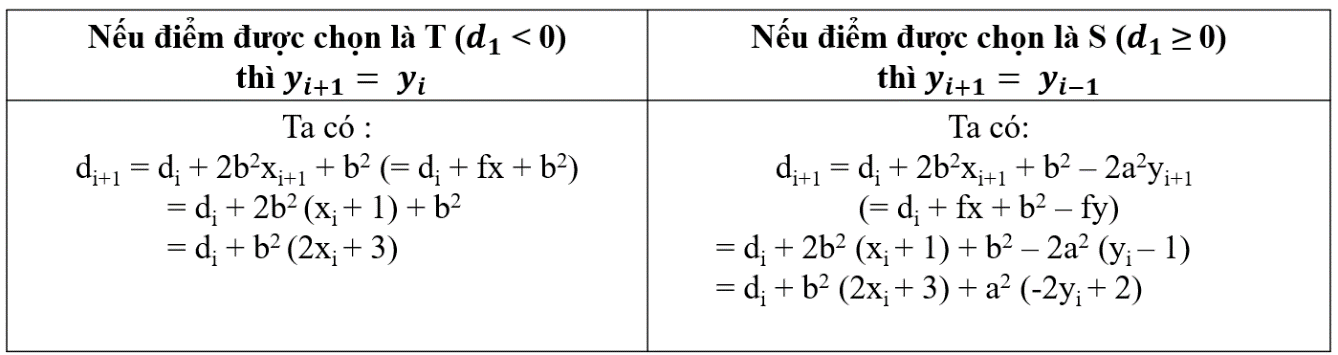
****+ Nếu di > 0 thì trung điểm I nằm ngoài đường elip 🡺 điểm được chọn là S.

+ Nếu di = 0 thì trung điểm I nằm trên đường elip   🡺 điểm được chọn là S.

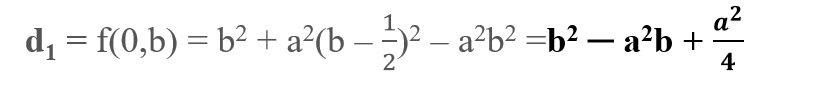
+ Nếu di < 0 thì trung điểm I nằm trong đường elip 🡺 điểm được chọn là T.

Tương tự, với bước nhảy i+1.   
Vì xi+1 =xi+1 nên ta suy ra được công thức chung :





Với điểm đầu tiên (0,b) ta có :



1. Trong vùng thứ 2

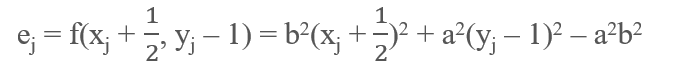
Chúng ta tính toán như vùng 1.

Giả sử xác định điểm (xj, yj), tiếp theo trên elip trong bước j+1.

Điểm được chọn là U hoặc V.

Trung điểm K của UV sẽ quyết định điểm được chọn (U hoặc V)

Giá trị của f(x,y) tại trung điểm K là :



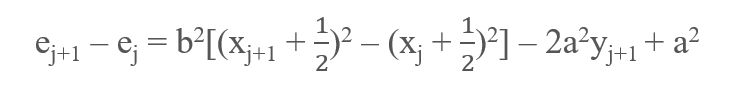
+ Nếu ej > 0 điểm được chọn là U.

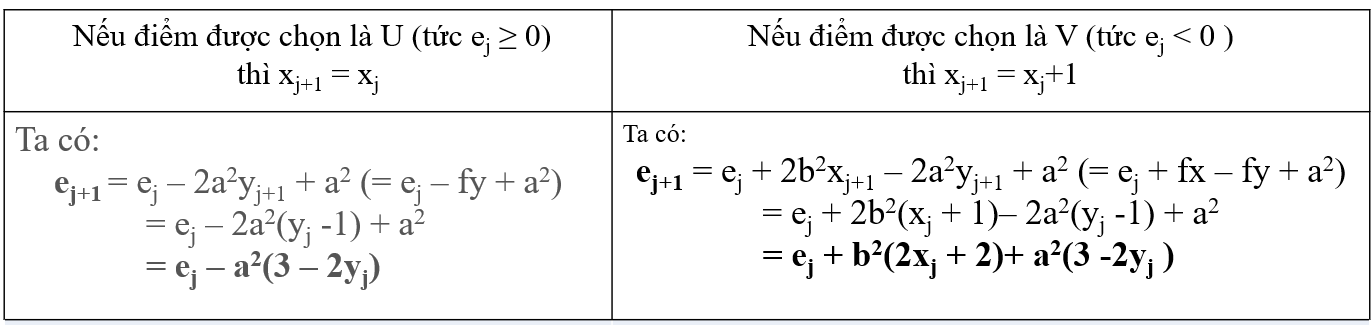
+ Nếu ej = 0 điểm được chọn là U

+ Nếu ej < 0 điểm được chọn là V.

Tương tự, với bước nhảy j+1

Vì yi+1 =yi -1 nên ta suy ra được công thức chung :



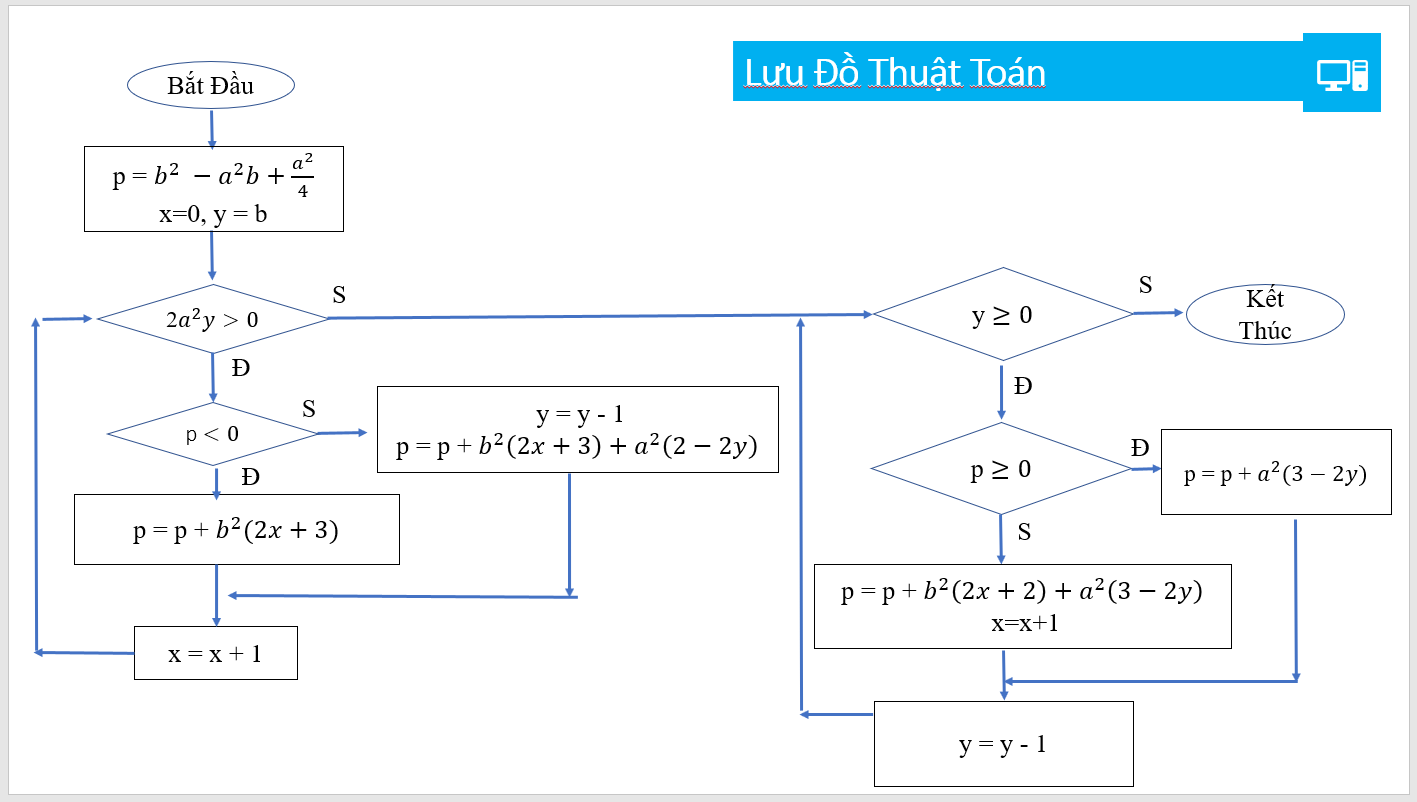


Bởi giá trị khởi tạo ban đầu trong phần 2 phụ thuộc vào vị trí cuối cùng của phần 1, giả sử là (xk ,yk)

Khi đó :



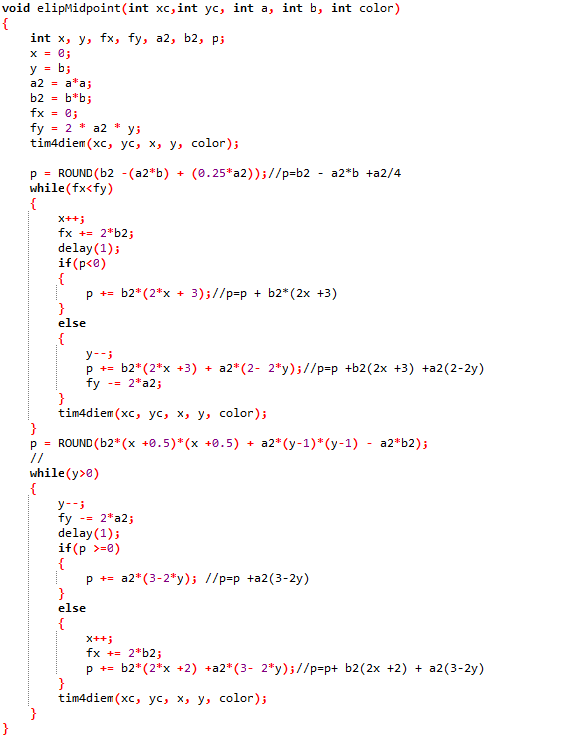
* Lưu đồ thuật toán



**CODE :**

|  |
| --- |
| #include<graphics.h>  #include<conio.h>//de dung man hinh dô hoa  #include<iostream>  #define ROUND(a) (int)(a+0.5)  using namespace std;    void plot(int xc, int yc, int x, int y, int color)  {  putpixel(xc+x, yc+y, color);  putpixel(xc-x, yc+y, color);  putpixel(xc+x, yc-y, color);  putpixel(xc-x, yc-y, color);  }  void elipMidpoint(int xc,int yc, int a, int b, int color)  {  int x, y, fx, fy, a2, b2, p;  x = 0;  y = b;  a2 = a\*a;  b2 = b\*b;  fx = 0;  fy = 2 \* a2 \* y;  plot(xc, yc, x, y, color);  p = ROUND(b2 -(a2\*b) + (0.25\*a2));//p=b2 - a2\*b +a2/4  while(fx<fy)  {  x++;  fx += 2\*b2;  delay(50);  if(p<0)  {  p += b2\*(2\*x + 3);//p=p + b2\*(2x +3)  }  else  {  y--;  p += b2\*(2\*x +3) + a2\*(2- 2\*y);//p=p +b2(2x +3) +a2(2-2y)  fy -= 2\*a2;  }  plot(xc, yc, x, y, color);  }  p = ROUND(b2\*(x +0.5)\*(x +0.5) + a2\*(y-1)\*(y-1) - a2\*b2);  //  while(y>0)  {  y--;  fy -= 2\*a2;  delay(50);  if(p >=0)  {  p += a2\*(3-2\*y); //p=p +a2(3-2y)  }  else  {  x++;  fx += 2\*b2;  p += b2\*(2\*x +2) +a2\*(3- 2\*y);//p=p+ b2(2x +2) + a2(3-2y)  }  plot(xc, yc, x, y, color);  }  }  int main()  {  int gd=DETECT, gm;  initgraph(&gd,&gm,"");  elipMidpoint(getmaxx()/2, getmaxy()/2,150,80,15);  getch();  closegraph();  return 0;  } |

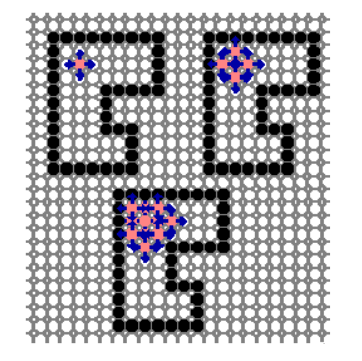
* Thuật toán Vẽ hình eclipse bằn thuật toán Midpoint



**3. HÀM HÀM TÔ LOANG**

**3.1 Ý TƯỞNG BÀI TOÁN :**

Thuật toán tô màu theo đường biên (hay còn gọi là tô màu loang đường biên, tô lân cận). Khác với thuật toán tô màu dựa theo dòng quét, đường biên của vùng tô màu ở thuật toán tô loang được xác định bởi tập các đỉnh của 1 đa giác, đường biên trong thuật toán được mô tả bằng một giá trị duy nhất, đó là màu của tất cả các điểm thuộc về đường biên (nói ngắn gọn là chúng ta sẽ tô đường biên một màu riêng).

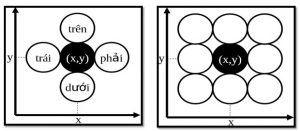


#### Hình 2 Thuật toán tô màu theo đường biên

Bắt đầu từ 1 điểm nằm bên trong vùng tô, ta sẽ kiểm tra các điểm lân cận của nó đã được tô màu hay có phải điểm biên hay không. Nếu không phải là điểm đã tô và không phải là điểm biên ta sẽ tô màu nó. Lặp lại cho tới khi nào không còn tô được điểm nào nữa thì dừng.

Thuật toán này có thể sẽ không hoạt động chính xác khi có 1 số điểm nằm trong vùng tô có màu là màu cần tô của vùng tô. Để khắc phục điều này, trước khi tô màu, cần phải đảm bảo rằng toàn bộ các điểm thuộc về vùng tô có màu khác màu tô.

Có hai quan điểm về cách tô này, đó là dùng 4 điểm lân cận hay 8 điểm lân cận đối với điểm đang xét được tô bằng màu trắng (chúng ta sẽ đi xét 4 điểm lân cận).



#### Hình 3 4 điểm lân cận và 8 điểm lân cận

Có hai hướng thực thi thuật toán: đó là sử dụng đệ quy và không đệ quy.

a) Thuật toán đệ quy:

-Bước 1: Kẻ vùng biên cần tô.

-Bước 2: Xác định một điểm (x, y) bên trong vùng cần tô.

-Bước 3: Tô điểm (x, y) sau đó tô loang những điểm lân cận.

Hàm đệ quy:

Ưu điểm:

+Có thể tô vùng có hình dạng bất kỳ.

+Cài đặt thuật toán dễ dàng.

Khuyết điểm:

+Không thể tô các vùng có kích thước lớn do tràn bộ nhớ khi sử dụng đệ quy.

+Việc gọi đệ quy, thuật toán cho 4 điểm lân cận của điểm hiện hành không quan tâm tới một trong bốn điểm đó đã được xét ở bước trước hay chưa (tức là sẽ có những điểm bị xét đi xét lại nhiều lần). Điều này khiến tốc độ chậm, không hiệu quả.

Chính vì đệ quy phức tạp như vậy nên hầu như ở các bài toán có tính đệ quy, chúng ta đều phải tìm cách khử đệ quy. Chúng ta sẽ tiến hành loang dần và lần lượt tô từng đoạn giao theo dòng quét ngang, thay vì tô theo 4 điểm lân cận.

b) Thuật toán khử đệ quy:

Áp dụng hàng đợi Queue, BFS

-Bước 1: Cất điểm hạt giống đầu tiên vào kho.

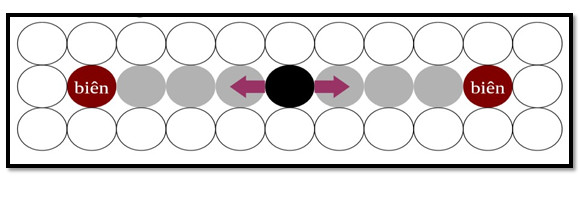
-Bước 2: Lặp nếu kho không rỗng

+Lấy điểm hạt giống.

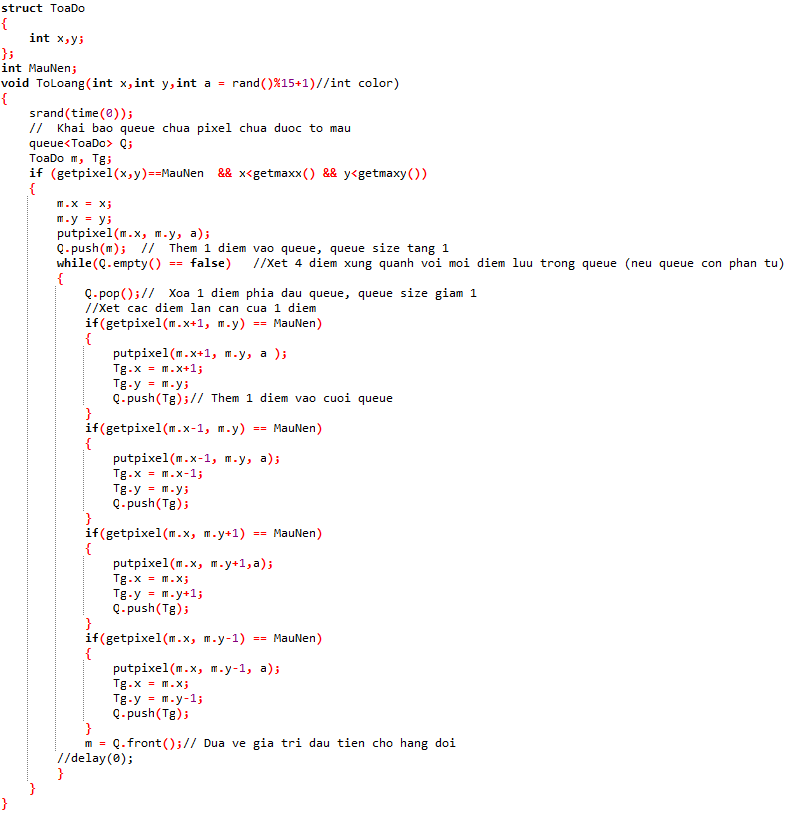
+Tô điểm hạt giống, sau đó tô loang sang 2 bên.

+Bổ sung những điểm hạt giống mới vào kho từ dòng trên và dòng dưới.

Tiêu chuẩn để làm điểm hạt giống: điểm này chưa được tô màu và không phải điểm biên.

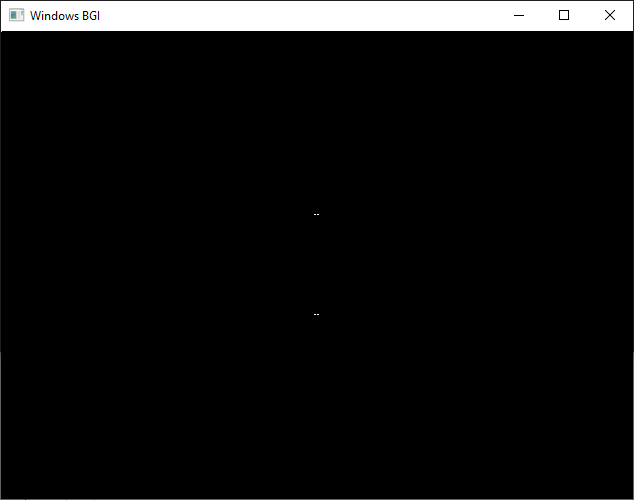


CODE :



**PHẦN 3: CHƯƠNG TRÌNH MINH HỌA**

* 1. **Hàm Vẽ Hình Ellipse bằng thuật toán Mid-Point**

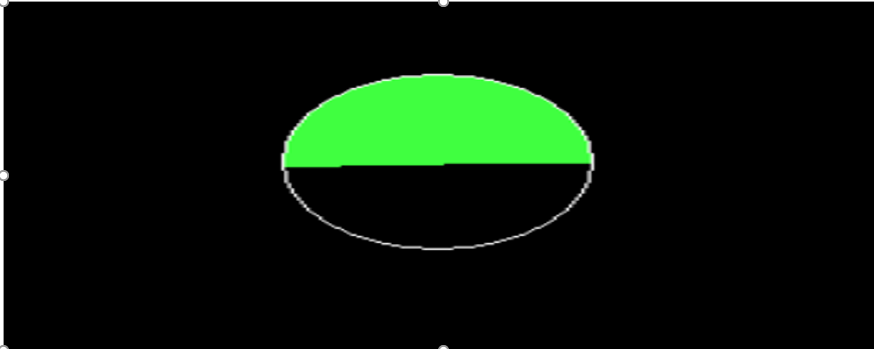


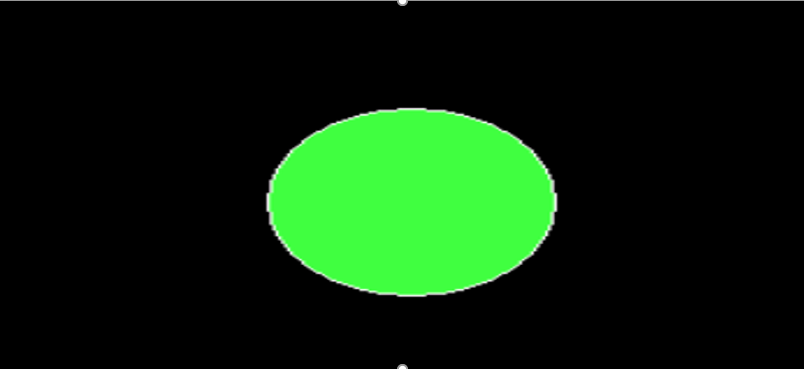
*Hình 4. Tìm 4 điểm*



*Hình 5. Hình Ellipse bằng thuật toán Mid-Point*

**3.2 Tô màu loang cho eclipse**



****

*Hình 6. Tô màu loang cho eclipse*

# **KẾT LUẬN**

Đề tài của nhóm chúng em có phần nào đó giúp mọi người hiểu rõ hơn về thuật toán Bresenham trong Đồ Họa Máy Tính và có thể ứng dụng trong học tập.

Qua việc thực hiện đề tài: **“***“*Vẽ hình elipse bằng thuật toán Midpoint, tô màu loang cho elip**”**, nhóm chúng em có thêm nhiều hiểu biết hơn về kiến thức Đồ Họa Máy Tính nói chung và Đồ Họa Máy Tính trên C nói riêng .

Trong quá trình thực hiện đề tài, vì còn giới hạn về kiến thức nên một số hạn chế của ba thuật toán xén hình nhóm chưa khắc phục được, và đề tài của nhóm khó có thể tránh khỏi thiếu sót. Nhóm rất mong nhận được ý kiến đóng góp của tất cả mọi người để đề tài của nhóm có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn**–** đã tận tình chỉ bảo và giúp đỡ chúng em thực hiện đề tài này.

***Một lần nữa chúng em xin chân thành cảm ơn !***

# **TÀI LIỆU THAM KHẢO**

1. Giáo trình “Kỹ thuật đồ họa”
2. Website tailieu.vn: <https://tailieu.vn/doc/giao-trinh-ky-thuat-do-hoa-1343460.html>
3. Website voer.edu.vn: <https://voer.edu.vn/c/giao-trinh-ky-thuat-do-hoa/6f8cfe5f>
4. Website ntu.edu.sg: <https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_Examples.html>