

SỬ DỤNG PHẦN MỀM CROCODILE ICT 6.05 HỖ TRỢ THIẾT KẾ THUẬT TOÁN TIN HỌC

Biên soạn: Thạc sĩ Nguyễn Hữu Thành

Giáo viên trường THCS Lạc Hồng, Q.10

Nội dung:

1. Cài đặt phần mềm Crocodile ICT 6.05
2. Màn hình làm việc
3. Cấu trúc chương trình và qui trình xây dựng thuật toán
4. Sử dụng các khối chức năng cơ bản vẽ sơ đồ thuật toán
5. Ví dụ minh họa và bài tập áp dụng
6. Đề thi tham khảo

I. Cài đặt phần mềm Crocodile ICT 6.05

1.1. Cấu hình yêu cầu

- Windows 2000, XP, Vista hoặc Windows 7
- Pentium III 500 MHz hoặc cao hơn
- 128 MB RAM
- 16 bit soundcard optional
- Web browser

Sau khi cài đặt xong, phần mềm sẽ được lưu vào thư mục “**C:\Program Files\Crocodile Clips\Crocodile ICT 605**” với dung lượng 88 MB.

1.2. Cài đặt phần mềm

- Mở thư mục có chứa phần mềm Crocodile ICT.
- Sau đó, double-click vào file **Crocodile_ICT_CI_605.exe** (có biểu tượng



để bắt đầu cài đặt).

- Thực hiện theo các chỉ dẫn trên màn hình và cuối cùng nhấn nút Finish để hoàn thành.

1.3. Nhập mã đăng ký phần mềm

- Ngắt kết nối mạng Internet.
- Chạy chương trình vừa cài xong, xuất hiện một hộp thoại, nhấp chọn nút **Edit License**.
- Nhập lần lượt vào 2 khung với giá trị sau:

Licensee Name	melatonin		
Licence Code	CI000SS	605	VUNGR

Click chọn **Next** để chương trình xác định các thông số vừa nhập là có thể sử dụng vĩnh viễn.

II. Màn hình làm việc

2.1. Các thành phần chính: gồm có:

- Thanh tiêu đề, thanh bảng chọn, thanh công cụ, thanh cuộn dọc - cuộn ngang
- Các cửa sổ (nằm bên trái).
- Vùng thiết kế sơ đồ: vùng trắng lớn bên phải, phía dưới có tối đa 5 Scene. Nhấn nút +/- để thêm/xóa Scene, nhấp đúp vào tên Scene để đặt tên Scene (mỗi Scene có thể thiết kế một hoặc nhiều sơ đồ, nhưng ta chỉ nên tạo 1 sơ đồ để tiện quản lý).



2.2. Thanh bảng chọn:

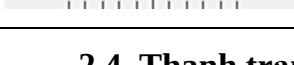
Mỗi bảng chọn gồm các lệnh có chức năng cùng nhóm. Thanh bảng chọn chứa tên các bảng chọn: **File, Edit, View, Scenes, Help**.

Bảng chọn	Mô tả
File	Các lệnh xử lý tệp chương trình như: New (tạo tệp mới), Open (mở tệp), Save (lưu tệp), Save as (lưu tệp với tên khác), Quit (thoát), ...
Edit	Các lệnh biên tập chương trình như: Copy (sao chép), Paste (dán), Delete (xóa), ...
View	Các lệnh hiển thị như: Zoom In (phóng to), Zoom Out (thu nhỏ), ...
Scenes	Các lệnh xử lý một scene như: New Scene (thêm mới), Remove Scene (xóa), ...
Help	Các hướng dẫn trợ giúp sử dụng phần mềm

2.3. Thanh công cụ:

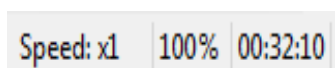
Thanh công cụ chứa biểu tượng của một số lệnh thường dùng. Để thực hiện lệnh, chỉ cần nhấp chuột vào biểu tượng tương ứng trên thanh công cụ.

Nút lệnh	Tính năng
 Delete part(s)	Xóa đối tượng đã chọn
 New	Tạo tệp mới
 Open	Mở một tệp đã có
 Save	Lưu tệp đang mở
 Print	In tệp đang mở
 Cut	Xóa đối tượng được chọn nhưng lưu vào vùng nhớ đệm

	Copy	Sao chép đối tượng được chọn vào vùng nhớ đệm
	Paste	Dán các đối tượng từ vùng nhớ đệm vào sơ đồ
	Undo	Hủy bỏ thao tác vừa làm
	Redo	Khôi phục thao tác vừa hủy bỏ
	Zoom In	Phóng to màn hình
	Zoom Out	Thu nhỏ màn hình
	Space Properties	Mở cửa sổ thuộc tính của scene hiện thời
	Pause	Tạm ngừng hoạt động của sơ đồ
	Show/Hide flowchart part labels	Hiện thị/ẩn nhãn của các khối
	Set flowchart speed	Thiết lập tốc độ chạy chương trình cho sơ đồ
	Simulation speed	Sử dụng con trượt để điều chỉnh tốc độ sơ đồ

2.4. Thanh trạng thái:

Thanh trạng thái là một thanh ngang đặt tại góc dưới bên phải màn hình làm việc, cho phép cung cấp thông tin về trạng thái hiện thời của một scene.



Ba mục luôn hiển thị ở bên trái của thanh trạng thái là:

- **Speed:** Hệ số tốc độ
- **Zoom:** Tỷ lệ phóng đại
- **Simulation time:** Khoảng thời gian (giờ, phút, giây) kể từ khi sơ đồ được tạo, mở ra hoặc tải về.

III. Cấu trúc chương trình và quy trình xây dựng thuật toán

3.1. Cấu trúc chương trình

Chương trình được lập trình trên Crocodile ICT là tập hợp các khối lệnh được tổ chức, sắp xếp sao cho thực hiện được ý tưởng của người lập trình. Vì thế người lập trình phải sử dụng những cấu trúc sau đây:

- *Cấu trúc tính toán:* Trong chương trình nếu có biểu thức tính toán thì sau khi gán giá trị cho các biến, hằng, tùy theo độ ưu tiên của các toán tử để tính toán kết quả và gán giá trị trả lại cho biến ở bên trái khối lệnh.

- *Cấu trúc tuần tự:* Các khối lệnh được sắp xếp để thực hiện tuần tự hết lệnh này đến lệnh khác theo thứ tự từ trên xuống (top-down).

- *Cấu trúc quyết định:* Thông qua khối lệnh **Decision \ Test**.

- *Cấu trúc lặp*: Dùng để lặp một khối lệnh hay một nhóm khối lệnh.

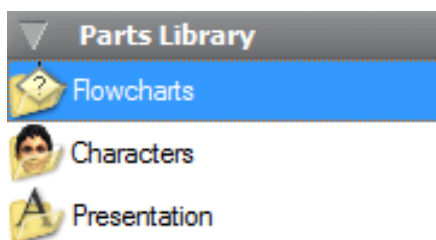
3.2. Quy trình xây dựng thuật toán

- Đọc yêu cầu bài toán, xác định dữ liệu vào và kết quả bài toán (Input/ Output).
- Phân tích và mô tả các bước giải bài toán.
- Vẽ sơ đồ thuật toán.
- Nhập dữ liệu đầu vào và chạy thử chương trình.
- Kiểm tra và sửa lỗi thiết kế (nếu có).
- Kiểm tra kết quả và hiệu chỉnh thuật toán (nhập nhiều bộ dữ liệu để kiểm tra các trường hợp của thuật toán).
- Hoàn tất chương trình.

IV. Sử dụng các khối chức năng cơ bản vẽ sơ đồ thuật toán

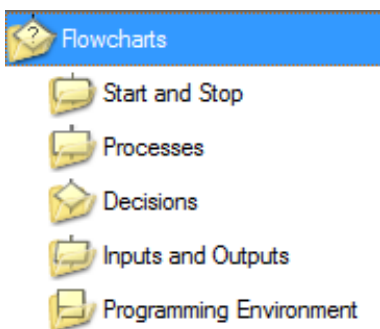
4.1. Các khối lệnh thường sử dụng

Các khối lệnh để xây dựng chương trình trên Crocodile ICT đều được đặt tại các thư mục trong **Parts Library**. Nó gồm có 3 thư mục chính sau:



- **Flowcharts**: các khối dùng để vẽ sơ đồ khối.
- **Characters**: các nhân vật để thực hiện hành động đã được lập trình.
- **Presentation**: gồm các hộp thoại, hình ảnh, ...

4.1.1. Flowcharts: Cung cấp các thành phần để tạo sơ đồ khối.

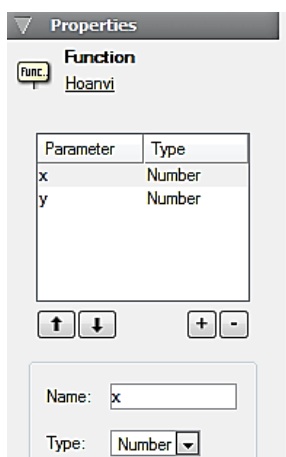


- **Start and Stop**: gồm các khối để bắt đầu hay kết thúc sơ đồ.
- **Processes**: quản lý tiến trình chạy.
- **Decisions**: kiểm tra điều kiện.
- **Inputs and Outputs**: quản lý nhập, xuất.
- **Programming Environment**: thể hiện môi trường lập trình.

4.1.1.1. Start and Stop

-  **Start**: Sử dụng để bắt đầu một sơ đồ. Khi sơ đồ đã được tạo hoàn chỉnh, nhấn nút  để thiết lập chạy chương trình. Khi sơ đồ đang chạy, ngưng lại bằng việc nhấn nút .
-  **Stop**: Dùng để kết thúc một chương trình. Bất kỳ một chương trình chính nào cũng đều phải kết thúc bằng thành phần Stop.

c. **Function**



Hoanvi (x , y)

Sử dụng để xây dựng một hàm. Tên hàm là duy nhất, có thể được thay đổi bằng việc click vào và sửa tên ở phần function. Danh sách tham số có thể được thêm vào bằng nút  hoặc .

- Name: Tên của tham số.

- Type: Kiểu tham số.

d. **Return**: Kết thúc một hàm.

4.1.1.2. Processes

a. **Set variable**: Dùng để gán giá trị cho biến.

Click vào **Set Variable** rồi kéo rê ra vùng soạn thảo. Tại đây ta có thể nhập giá trị cho biến: (lưu ý tên biến có phân biệt chữ in, chữ thường và không chứa khoảng trắng).

x ← 0

+ Click vào tên biến (bên trái mũi tên) để thay đổi tên biến.

+ Click vào giá trị của biến (bên phải mũi tên) để nhập giá trị hoặc biểu thức cho biến.

b. **Increment variable**: Tăng giá trị cho biến.

c. **Random**: Phát sinh 1 giá trị là số nguyên ngẫu nhiên (từ 0 đến 100) cho biến.

d. **Call function**: Dùng để gọi tên một hàm. Chương trình chính sẽ tạm ngưng cho đến khi giá trị cuối cùng của hàm được trả về, sau đó chương trình chính sẽ được tiếp tục.

4.1.1.3. Decisions

 Test: Giống với ô kiểm tra điều kiện trong sơ đồ khối, dùng để so sánh giá trị của hai biểu thức.

4.1.1.4. Inputs and Outputs

a. **Get property**: Nhập giá trị cho biến.

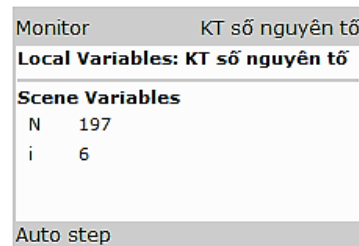
- b.  **Set property** : Đưa giá trị biến ra màn hình.

Hai khối này thường kết hợp với khối  **Edit box** trong thư mục **Presentation** để nhập giá trị hoặc xuất kết quả.

4.1.1.5. Programming Environment

-  **Monitor box**

Hộp thoại này thể hiện toàn bộ hoạt động và sự kiện của scene. Nhờ đó, học sinh có thể thấy được sự thay đổi giá trị của các biến qua từng bước thực hiện trong chương trình.



-  **Scene (global) variables**

Hộp thoại này thể hiện sự thay đổi giá trị của các biến toàn cục (global) trong chương trình. Nhấn nút + để đưa tên biến cần theo dõi vào và nhấn nút - để loại bỏ tên biến.

Scene (global) variables		
Variable	Initial	Current
- N	0	197
- i	0	6
+		

4.1.2. Presentation

Inputs-Outputs:

-  **Text** : Dùng để chèn thêm văn bản vào các mô hình và có thể tác động đến như một thành phần thông thường.
-  **Edit box** : Dùng để liên kết với các thành phần khác, chẳng hạn như liên kết với *Get property* để nhập giá trị cho biến, liên kết với *Set property* để đưa giá trị của biến ra màn hình.

4.2. Thư viện hàm có sẵn

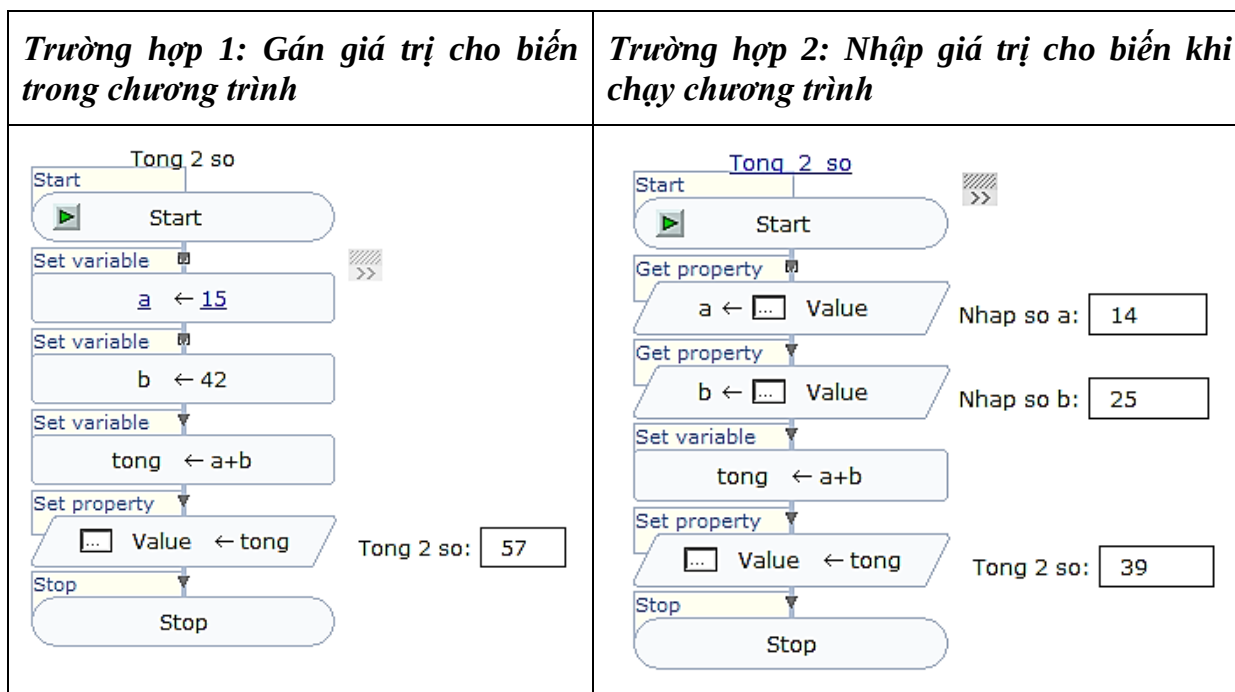
Cú pháp	Ý nghĩa
$\text{sqrt}(x)$	Trả về giá trị là căn bậc 2 của x ($x \geq 0$)
$\text{fact}(x)$	Trả về giá trị là giai thừa của x
$\text{pow}(x,y)$	Trả về giá trị x lũy thừa y
$\text{mod}(a,b)$	Trả về giá trị là phần dư của phép chia a cho b
$\text{rand}()$	Trả về một số ngẫu nhiên trong khoảng $[0,1]$
$\text{abs}(x)$	Trả về giá trị là trị tuyệt đối của x
$\text{floor}(x)$	Trả về giá trị của x được làm tròn xuống số nguyên gần nhất
$\text{round}(x)$	Trả về giá trị của x được làm tròn đến số nguyên gần nhất

V. Ví dụ minh họa và bài tập áp dụng

5.1. Ví dụ minh họa:

5.1.1. Dùng cấu trúc tuần tự:

Ví dụ 1: Vẽ sơ đồ thuật toán nhập vào 2 số nguyên a, b. Tính tổng 2 số và xuất kết quả.



Ví dụ 2: Vẽ sơ đồ thuật toán đổi giá trị của 2 biến x và y.

- Xác định bài toán:

INPUT: Hai biến x, y có giá trị tương ứng là a và b (a, b có giá trị xác định).

OUTPUT: Hai biến x, y có giá trị tương ứng là b và a.

- Thuật toán:

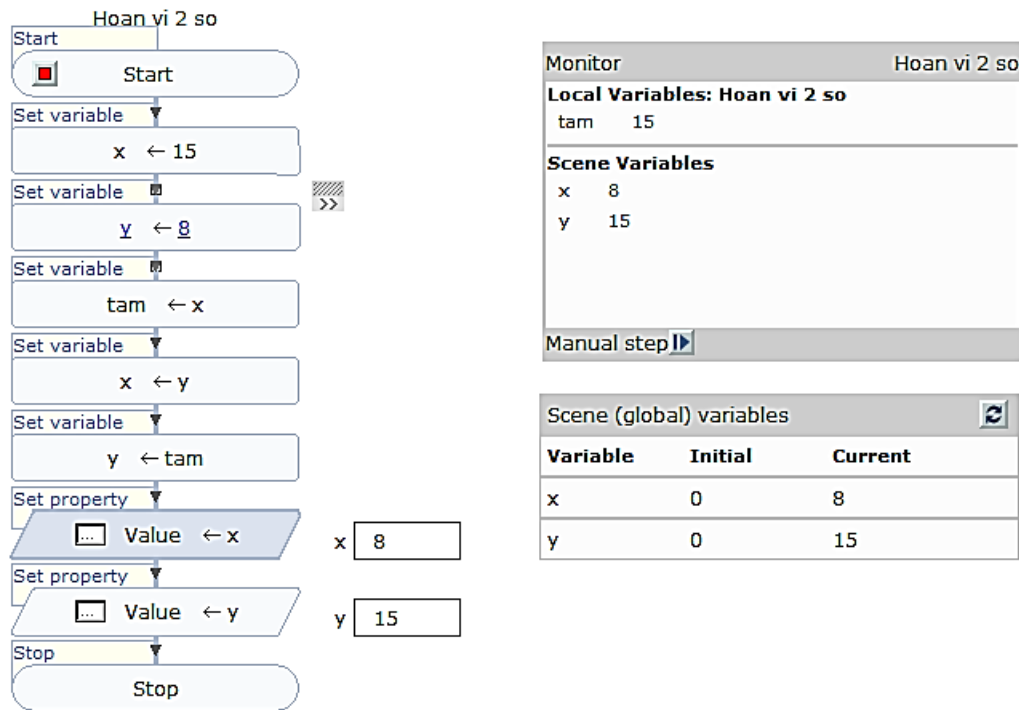
Bước 1: tam $\leftarrow$ x

Bước 2: x $\leftarrow$ y

Bước 3: y $\leftarrow$ tam

- Vẽ sơ đồ thuật toán:

Trường hợp 1: Gán giá trị cho biến trong chương trình



Trường hợp 2: Nhập giá trị cho biến khi chạy chương trình

Học sinh tự vẽ sơ đồ.

5.1.2. Dùng cấu trúc rẽ nhánh:

Ví dụ 1: Cho hai số thực a và b. Hãy cho biết kết quả so sánh hai số đó dưới dạng “a lớn hơn b”, “a nhỏ hơn b” hoặc “a bằng b”.

- Xác định bài toán:

INPUT: Hai biến số thực a và b (a, b có giá trị xác định).

OUTPUT: Kết quả so sánh.

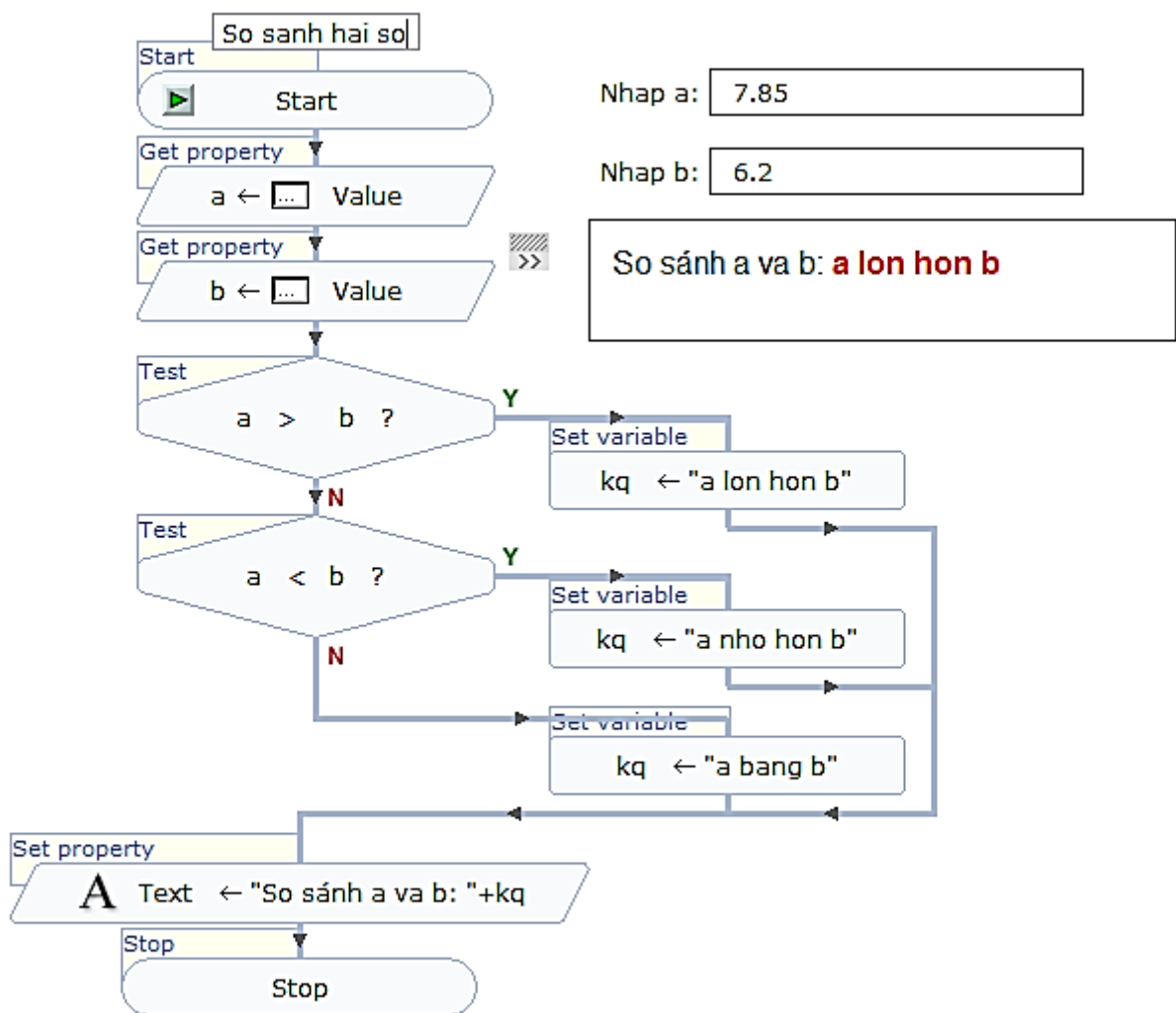
- Thuật toán:

Bước 1: Nếu $a > b$, kết quả là “a lớn hơn b” và chuyển đến bước 3.

Bước 2: Nếu $a < b$, kết quả là “a nhỏ hơn b”; ngược lại, kết quả là “a bằng b”.

Bước 3: Kết thúc thuật toán.

- Vẽ sơ đồ thuật toán: (giá trị a, b được nhập khi chạy chương trình)



Ví dụ 2: Viết chương trình tìm và đưa ra nghiệm của phương trình bậc hai tổng quát: $ax^2 + bx + c = 0$ ($a \neq 0$).

- Xác định bài toán:

INPUT: Ba số thực a, b, c ($a \neq 0$).

OUTPUT: Kết luận về nghiệm của pt bậc hai: $ax^2 + bx + c = 0$.

- Thuật toán:

Bước 1: Tính $\Delta \leftarrow b^2 - 4ac$

Bước 2: Xét giá trị của Delta trong 3 trường hợp sau:

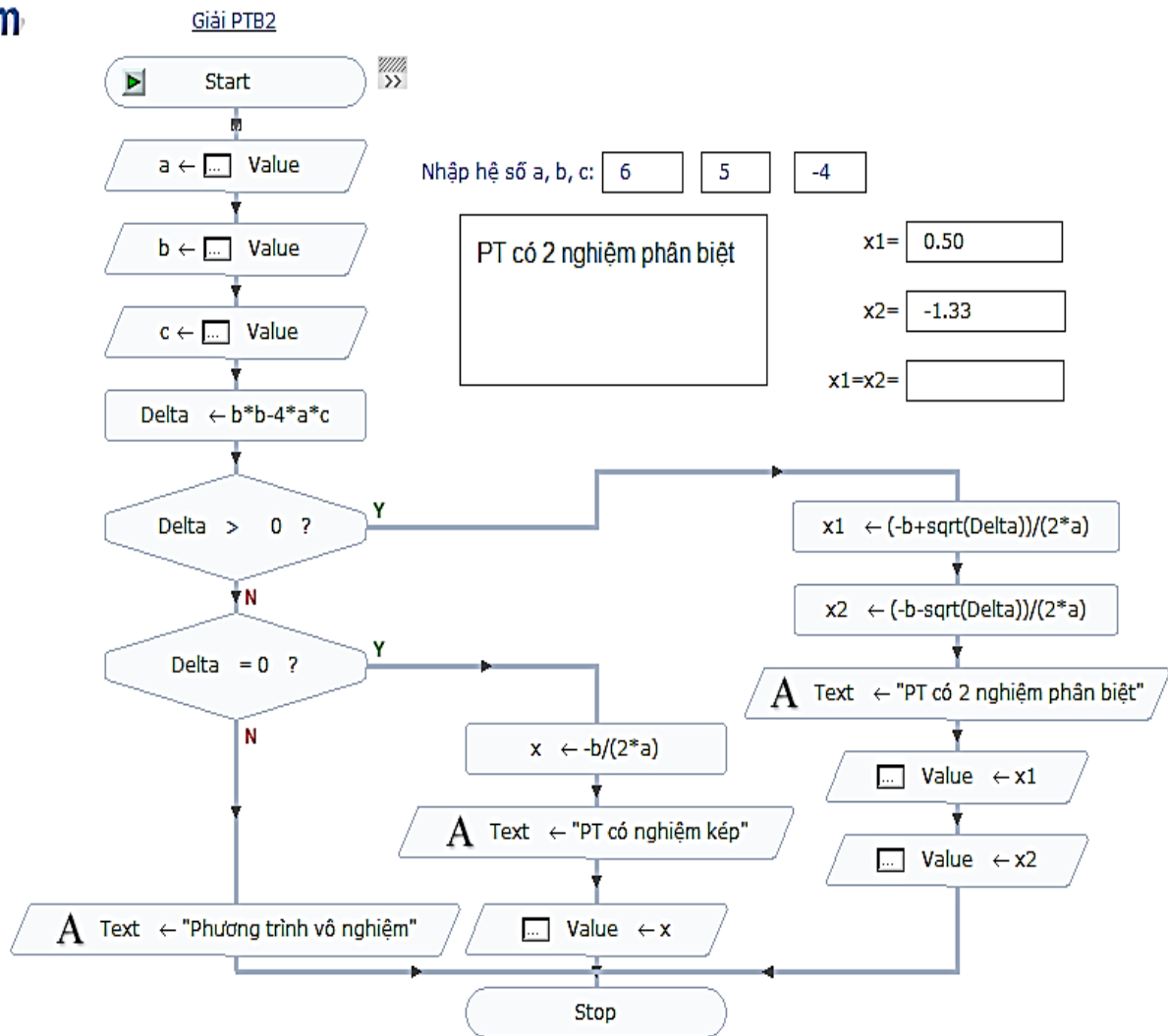
+ Nếu $\Delta > 0$: Tính giá trị 2 nghiệm: $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$; $x_2 = \frac{-b - \sqrt{\Delta}}{2a}$, thông báo “Phương trình có 2 nghiệm phân biệt”

+ Nếu $\Delta = 0$: Tính giá trị nghiệm kép: $x_1 = x_2 = -b/2a$, thông báo “Phương trình có nghiệm kép”.

+ Nếu $\Delta < 0$: Thông báo “Phương trình vô nghiệm”

Bước 3: Xuất thông báo trong từng trường hợp của Delta và các nghiệm của phương trình (nếu có nghiệm). Kết thúc thuật toán.

- Vẽ sơ đồ thuật toán:



5.1.3. Dừng cấu trúc lặp:

Ví dụ 1: Viết chương trình tìm ước chung lớn nhất (UCLN) của hai số nguyên dương M và N

- Xác định bài toán:

INPUT: Hai số nguyên dương M và N.

OUTPUT: UCLN(M,N).

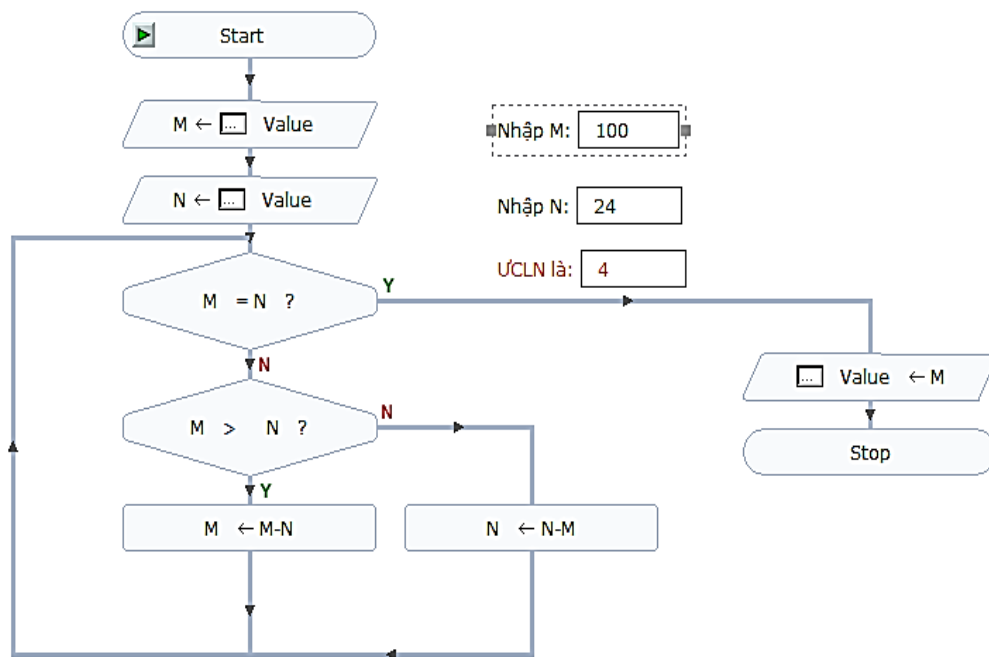
- Thuật toán:

Bước 1: Nếu $M = N$ thì chuyển đến bước 3.

Bước 2: Nếu $M > N$ thì gán $M \leftarrow M - N$; Ngược lại, gán $N \leftarrow N - M$

Bước 3: Thông báo giá trị M và kết thúc thuật toán.

- Vẽ sơ đồ thuật toán:



Ví dụ 2: Viết chương trình kiểm tra tính nguyên tố của một số nguyên dương N.

- Xác định bài toán:

INPUT: Số nguyên dương N.

OUTPUT: Thông báo “N là số nguyên tố” hoặc “N không là số nguyên tố”.

- Thuật toán:

Dựa theo tính chất của số nguyên tố: “Số nguyên tố là số nguyên lớn hơn 1, không có ước số ngoài 1 và chính nó”.

Như vậy, ta có ý tưởng: nếu $N > 1$ thì xét lần lượt các số từ 2 đến N (dùng biến đếm i), nếu có số i nào là ước số của N ($N \bmod i = 0$) thì dừng lại, ngược lại tăng biến đếm i lên 1 đơn vị.

Xét giá trị của i:

+ Nếu $i = N$ thì kết luận “N là số nguyên tố”.

+ Ngược lại, kết luận “N không phải số nguyên tố”.

Thuật toán được mô tả như sau:

Bước 1: Gán $i \leftarrow 2$. Nếu $N \leq 1$ thì nhảy đến bước 5.

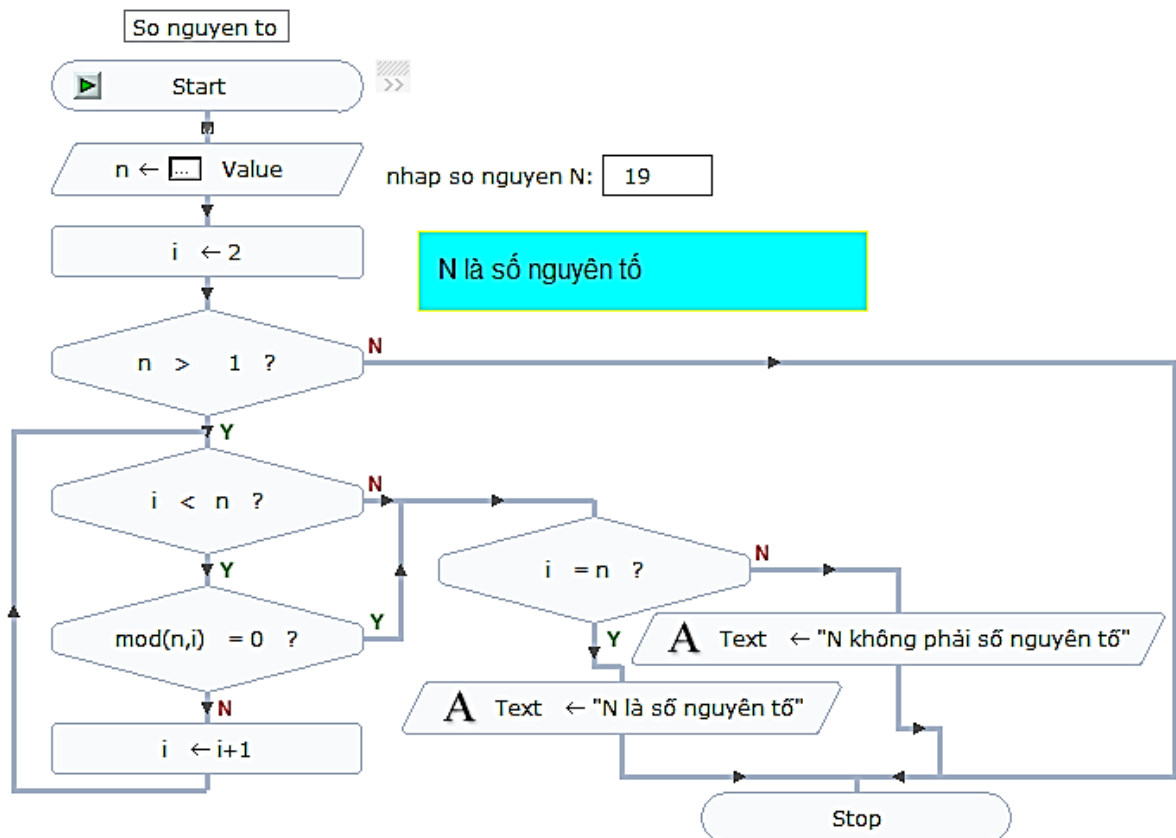
Bước 2: Nếu $N \bmod i = 0$ thì nhảy đến bước 4.

Bước 3: Gán $i \leftarrow i + 1$. Nếu $i \leq n$ thì quay lại bước 2.

Bước 4: Nếu $i = N$, kết luận “N là số nguyên tố”; ngược lại, kết luận “N không phải số nguyên tố”.

Bước 5: Kết thúc thuật toán.

- Vẽ sơ đồ thuật toán:



Ví dụ 3: Viết chương trình tính tổng các chữ số của một số nguyên.

- Xác định bài toán:

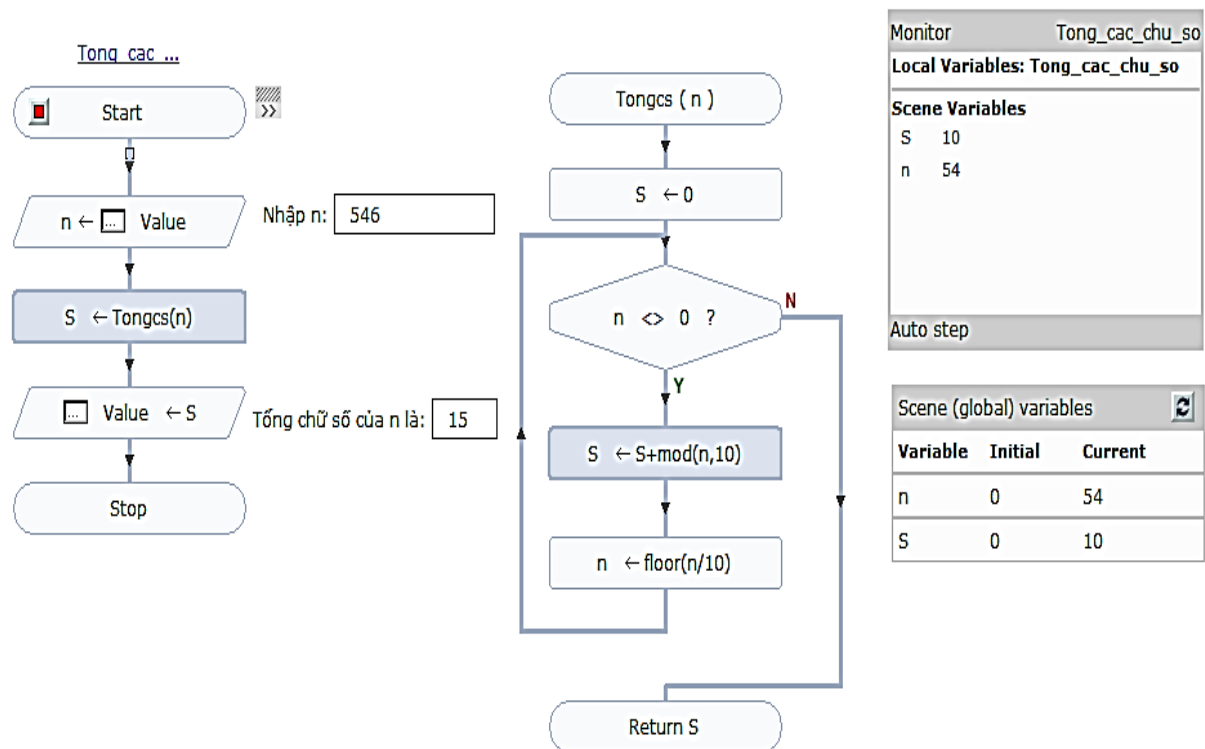
INPUT: Số nguyên dương N.

OUTPUT: Tổng các chữ số của N.

- Thuật toán:

Ý tưởng là ta chia số đó cho 10 lấy dư (**mod**) thì được chữ số hàng đơn vị, và lấy số đó **div** 10 thì sẽ được phần còn lại. Do đó sẽ chia liên tục cho đến khi không chia được nữa (số đó bằng 0), mỗi lần chia thì được một chữ số và ta cộng dồn chữ số đó vào tổng.

- Vẽ sơ đồ thuật toán:



* **Bài tương tự:** Viết chương trình tính tích các chữ số của một số nguyên.

Tính tích các chữ số cũng tương tự, chỉ cần chú ý ban đầu gán S là 1 và thực hiện phép nhân S với $n \bmod 10$.

5.1.4. Xử lý dãy số (mảng):

Ví dụ 1: Tìm số lớn nhất trong dãy A các số $a_1, a_2, \dots, a_n$ cho trước.

- Xác định bài toán:

INPUT: Dãy A các số $a_1, a_2, \dots, a_n$ ($n \geq 1$).

OUTPUT: Giá trị $MAX = \max\{a_1, a_2, \dots, a_n\}$.

- Thuật toán:

Ta sẽ dùng biến MAX để lưu số lớn nhất của dãy A. Việc xác định MAX có thể được thực hiện như sau: Đầu tiên gán giá trị a_1 cho biến MAX. Tiếp theo, lần lượt so sánh các số $a_2, \dots, a_n$ của dãy A với MAX. Nếu $a_i > MAX$, ta gán a_i cho MAX.

Từ đó, ta có thuật toán sau:

Bước 1: $MAX \leftarrow a_1; i \leftarrow 1$.

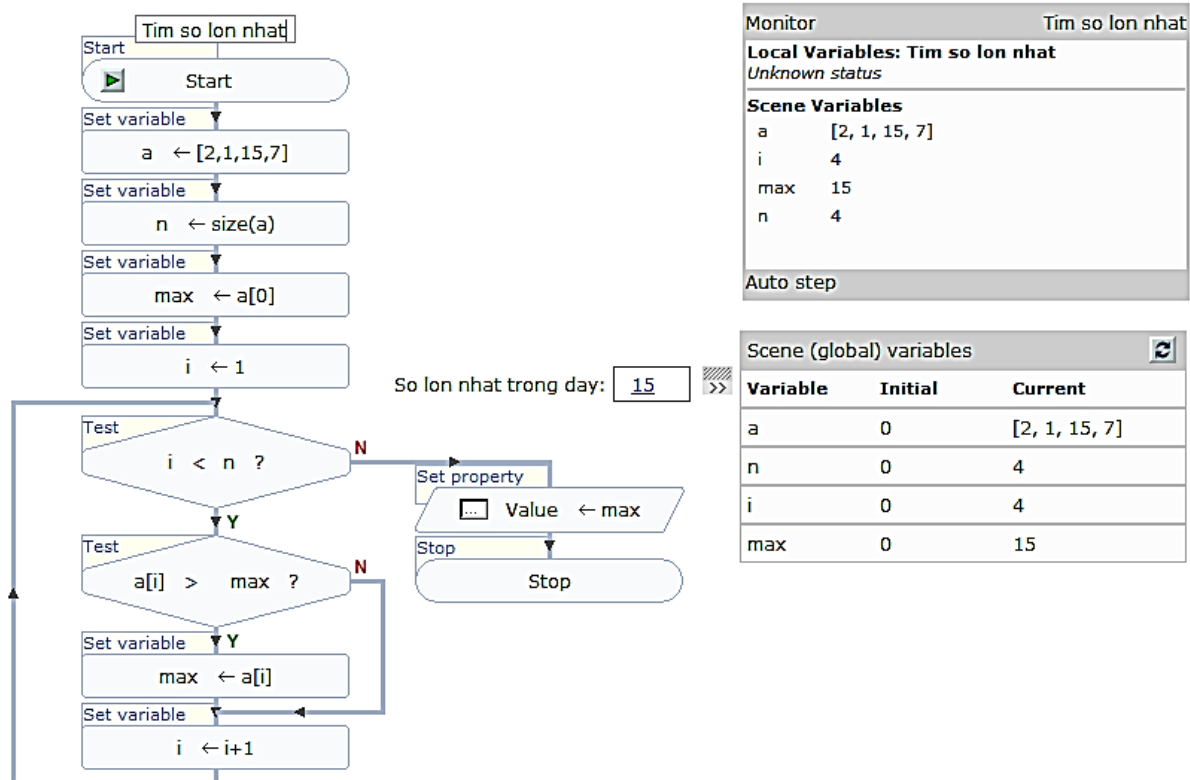
Bước 2: Nếu $a_i > MAX$, $MAX \leftarrow a_i$.

Bước 3: $i \leftarrow i+1$.

Bước 4: Nếu $i \leq n$ thì quay lại bước 2.

Bước 5: Thông báo giá trị MAX và kết thúc thuật toán.

- Vẽ sơ đồ thuật toán:



Ví dụ 2: Cho dãy A gồm N số nguyên khác nhau: $a_1, a_2, \dots, a_N$ và một số nguyên k. Cần biết có hay không chỉ số i ($1 \leq i \leq N$) mà $a_i = k$, nếu có hãy cho biết chỉ số đó. Sử dụng thuật toán tìm kiếm tuần tự, viết chương trình để giải bài toán.

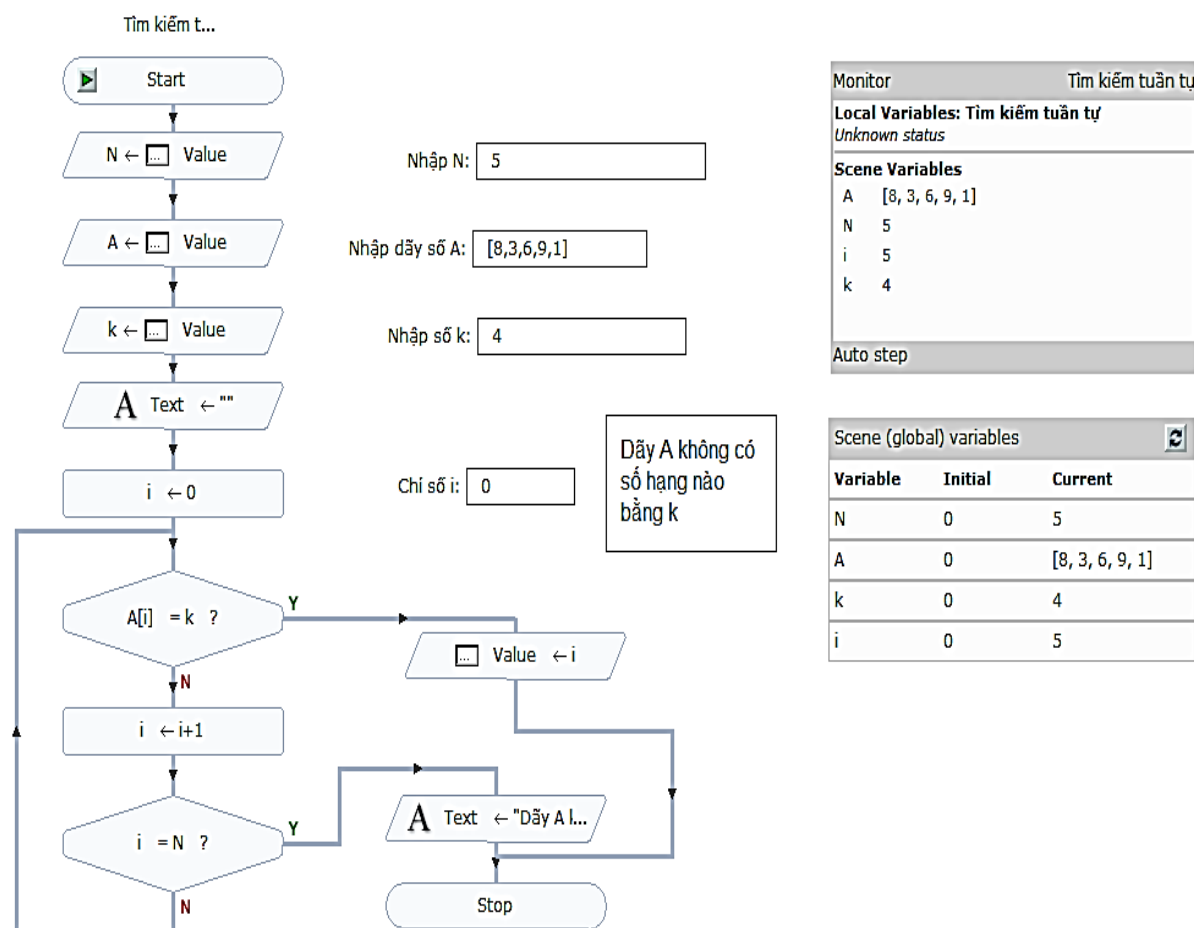
- Xác định bài toán:

INPUT: Dãy A gồm N số nguyên khác nhau: $a_1, a_2, \dots, a_N$ và số nguyên k.

OUTPUT: Chỉ số i mà $a_i = k$ hoặc thông báo không có số hạng nào của dãy A có giá trị bằng k.

- Thuật toán:

- Về sơ đồ thuật toán:



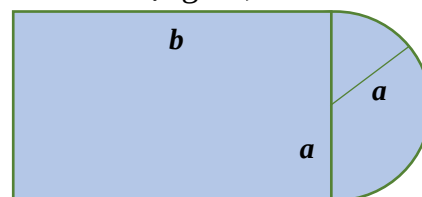
5.2. Bài tập áp dụng:

5.2.1. Bài 1: Giải phương trình bậc nhất dạng tổng quát $bx + c = 0$

- Input: Các số b và c .
- Output: Nghiệm của phương trình bậc nhất.

5.2.2. Bài 2: Tính diện tích ghép bởi các hình cơ bản

Một hình **A** được ghép từ một hình chữ nhật với chiều rộng $2a$, chiều dài b và một hình bán nguyệt bán kính a như hình sau đây:



- Input: Các số a và b .
- Output: Diện tích của hình **A**.

5.2.3. Bài 3: Tính tổng của 100 số tự nhiên đầu tiên

- Input: Dãy **A** chứa 100 số tự nhiên đầu tiên.
- Output: Giá trị của tổng $S = 1 + 2 + \dots + 100$.

5.2.4. Bài 4: Tính tổng các phân tử lớn hơn 0 trong dãy số **A** có n phân tử.

- Input: Dãy số **A** có n phân tử.
- Output: Tổng các phân tử lớn hơn 0 trong dãy số **A**.

5.2.5. Bài 5: Tìm các số có giá trị nhỏ nhất trong dãy số **A** có n phân tử.

- Input: Dãy số **A** có n phân tử.
- Output: Số lượng các số có giá trị nhỏ nhất trong dãy số **A**.

5.2.6. Bài 6: Kiểm tra 3 số nguyên dương a, b, c có hợp thành một tam giác không?

- Input: Ba số nguyên dương a, b, c.
- Output: Kết quả kiểm tra, xuất thông báo: “Ba số a, b, c là độ dài ba cạnh của một tam giác” hoặc “Ba số a, b, c không là độ dài ba cạnh của một tam giác”.

Áp dụng định lý: “Trong một tam giác, tổng hai cạnh bất kỳ luôn lớn hơn cạnh còn lại”.

5.2.7. Bài 7: Tính tổng các ước số của một số nguyên

- Input: Số nguyên N.
- Output: Tổng các ước số của N.

Ý tưởng: Để tính tổng các ước số của số N, ta cho i chạy từ 1 đến $\sqrt{N}$, nếu N chia hết cho số nào thì ta cộng số đó vào tổng. (Cách tính này chưa xét $\sqrt{N}$ cũng là ước số của N).

Dựa trên thuật toán tính tổng ước số, ta có thể kiểm tra được một số nguyên có là số hoàn thiện không: “Số nguyên gọi là số hoàn thiện nếu nó bằng tổng các ước số của nó”.

VI. Đề thi tham khảo

Đề thi Khéo tay kỹ thuật môn Crocodile ICT năm học 2014-2015 cấp thành phố (thời gian: 60 phút), thi ngày 08/02/2015 tại trường THCS Hồng Bàng, Q.5.

Tên file lưu trữ: ICT_sbd.CXI tại ổ đĩa qui định với sbd là số báo danh thực tế.

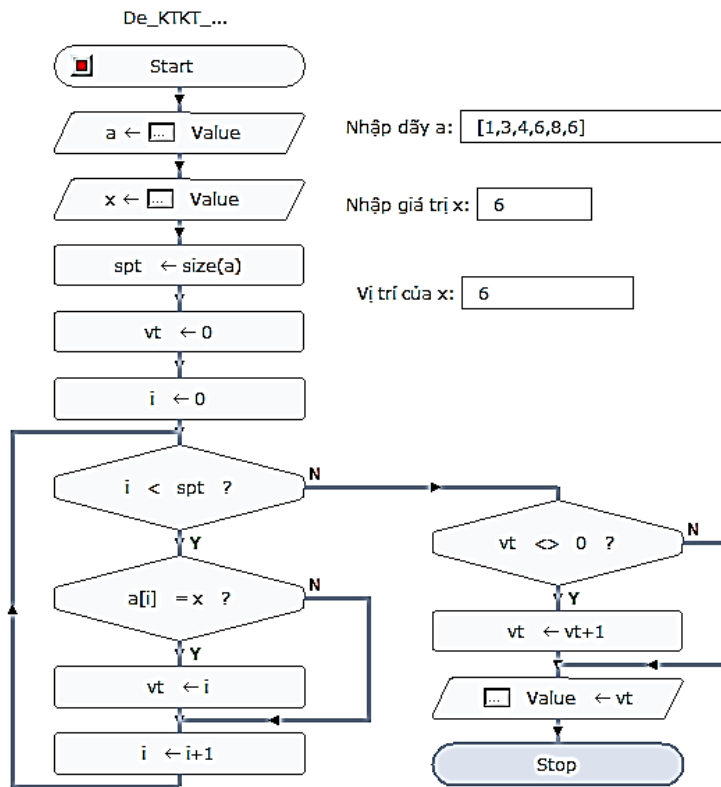
Đề: *Xác định vị trí phần tử có giá trị x của một dãy cho trước. Trường hợp không có giá trị x thì xuất vị trí 0, trong trường hợp có nhiều giá trị x thì xuất vị trí x cuối cùng của dãy. Lưu ý vị trí trong dãy theo thứ tự từ 1 đến n.*

Input: Cho trước dãy a gồm các số nguyên.

Output: Xác định vị trí phần tử có giá trị x cho trước của dãy.

Ví dụ 1:	Ví dụ 2:	Ví dụ 3:
- Dãy a: 1, 3, 4, 6, 8	- Dãy a: 1, 3, 4, 6, 8	- Dãy a: 1, 3, 3, 6, 3
- Giá trị x: 6	- Giá trị x: 9	- Giá trị x: 3
Xuất: 4	Xuất: 0	Xuất: 5

Bài giải mẫu:



DE THI KTKT CẤP TP

Ngày 08/02/2015

(Nhập bằng Input box)

Monitor De_KTKT_cap_TP_8_2_2015_2

Local Variables:

De_KTKT_cap_TP_8_2_2015_2

a [1, 3, 4, 6, 8, 6]

i 6

spt 6

Scene Variables

vt 6

x 6

Auto step

Scene (global) variables

Variable	Initial	Current
x	0	6
vt	0	6

- HẾT -