**1. Khái niệm lập trình và ngôn ngữ lập trình.**

* Lập trình là quá trình tạo ra các chương trình máy tính bằng cách viết các mã lệnh để điều khiển hành vi của máy tính. Nó là quá trình tạo ra một chuỗi các câu lệnh mà máy tính có thể hiểu và thực thi để thực hiện một tác vụ cụ thể.
* Ngôn ngữ lập trình là một tập hợp các quy tắc và cú pháp được sử dụng để viết các chương trình máy tính. Nó cung cấp các khái niệm, cú pháp và cấu trúc để diễn đạt ý nghĩa và chỉ dẫn cho máy tính. Ngôn ngữ lập trình giúp lập trình viên giao tiếp với máy tính thông qua việc viết mã lệnh.

**2. Khái niệm thuật toán/giải thuật?**

* Thuật toán là một tập hợp các bước rõ ràng và cụ thể để giải quyết một vấn đề hoặc thực hiện một tác vụ. Nó là một quy trình hoặc phương pháp được thiết kế để đạt được kết quả mong muốn thông qua việc thực hiện một loạt các thao tác, tính toán hoặc xử lý.
* Một thuật toán định nghĩa cách giải quyết một vấn đề từ một trạng thái ban đầu đến trạng thái kết thúc, bằng cách sử dụng một chuỗi các bước logic và tính toán. Nó có thể được biểu diễn dưới dạng một thuật toán chung hoặc thuật toán cụ thể cho từng vấn đề cụ thể.
* Mục tiêu của một thuật toán là tối ưu hóa thời gian thực thi, tối ưu hóa không gian lưu trữ và/hoặc tối ưu hóa các yếu tố khác tùy thuộc vào vấn đề cụ thể. Một thuật toán hiệu quả sẽ thực hiện tác vụ một cách chính xác, hiệu quả và đưa ra kết quả đúng theo yêu cầu.
* Giải thuật là một thuật toán cụ thể được thiết kế để giải quyết một vấn đề hoặc thực hiện một tác vụ cụ thể. Giải thuật thường là một phần của một hệ thống lớn hơn hoặc một ứng dụng cụ thể. Các giải thuật có thể được phân loại dựa trên các khía cạnh như độ phức tạp, phương pháp tiếp cận, mục tiêu tối ưu, và các yếu tố khác.

**3. Pseudo code là gì?**

Pseudo code là một ngôn ngữ mô tả thuật toán bằng cách sử dụng một ngôn ngữ tự nhiên gần gũi với ngôn ngữ con người. Nó không phải là một ngôn ngữ lập trình cụ thể, mà chỉ là một cách mô tả thuật toán một cách rõ ràng và dễ hiểu cho con người.

Pseudo code giúp lập trình viên mô tả thuật toán một cách trừu tượng, không phụ thuộc vào ngôn ngữ lập trình cụ thể. Nó cho phép mô tả các bước và luồng đi của thuật toán một cách dễ dàng và trực quan hơn so với việc sử dụng ngôn ngữ lập trình thực tế.

Pseudo code thường sử dụng các từ khóa và cú pháp đơn giản, giống với ngôn ngữ tự nhiên, để biểu thị các cấu trúc lặp, câu lệnh điều kiện, khai báo biến, và các phép toán cơ bản. Ví dụ, trong pseudo code, lặp có thể được biểu diễn bằng cú pháp như "while", "for" hoặc "repeat-until", và câu lệnh điều kiện có thể sử dụng cú pháp như "if-else".

Một ví dụ về pseudo code cho thuật toán tính tổng của hai số là:

```

Input: a, b (hai số nguyên)

Output: sum (tổng của a và b)

sum = a + b

```

Pseudo code giúp lập trình viên xác định rõ ràng cấu trúc và logic của thuật toán trước khi triển khai nó bằng một ngôn ngữ lập trình cụ thể. Nó là một công cụ hữu ích để diễn giải và trao đổi ý tưởng thuật toán giữa các thành viên trong nhóm phát triển.

**4. Flowchart là gì?**

Flowchart là một biểu đồ hình vẽ được sử dụng để biểu diễn một quy trình hoặc thuật toán bằng các hình khối và các mũi tên. Nó là một công cụ trực quan và đồ họa để mô tả dòng chảy của các bước và quyết định trong một quy trình.

Flowchart được sử dụng để biểu diễn một loạt các bước hoặc hành động theo một thứ tự cụ thể. Mỗi bước hoặc hành động được biểu diễn bằng một hình khối, và mũi tên được sử dụng để chỉ ra luồng dòng chảy của các bước. Các hình khối trong flowchart có thể biểu diễn các loại hành động khác nhau như câu lệnh, quyết định, lặp lại, đầu vào/đầu ra, và các hành động khác.

Các hình khối phổ biến trong flowchart bao gồm:

1. Hình khối hình chữ nhật: Đại diện cho một bước thực hiện hoặc một câu lệnh.

2. Hình khối hình tròn: Đại diện cho một quyết định hoặc một điều kiện.

3. Hình khối hình hình bầu dục: Đại diện cho một quy trình đầu vào/đầu ra.

4. Hình khối hình chữ nhật với góc bị bo tròn: Đại diện cho một bước kết thúc hoặc một bước thoát khỏi quy trình.

Flowchart giúp lập trình viên hiểu và diễn giải thuật toán một cách trực quan và dễ hiểu. Nó cũng là một công cụ hữu ích để giao tiếp và trao đổi ý tưởng giữa các thành viên trong nhóm phát triển.

**5. Dùng pseudocode và flowchart để thực hiện những bài sau**

**a. Hiển thị: "Hello World!"**

**b. Tính tổng 2 số a, b**

**c. Kiểm tra một số là có phải là chẵn hay không**

**d. Hiển thị 1000 lần: "Hello World!"**

a. Pseudocode:

```

Output "Hello World!"

```

Flowchart:

```

[Start] --> [Output "Hello World!"] --> [End]

```

b. Pseudocode:

```

Input a, b (hai số nguyên)

sum = a + b

Output sum

```

Flowchart:

```

[Start] --> [Input a, b] --> [sum = a + b] --> [Output sum] --> [End]

```

c. Pseudocode:

```

Input num (số nguyên)

IF num % 2 == 0 THEN

Output "Số là chẵn"

ELSE

Output "Số là lẻ"

ENDIF

```

Flowchart:

```

[Start] --> [Input num] --> [num % 2 == 0?] --(Yes)--> [Output "Số là chẵn"] --> [End]

|

+--(No)--> [Output "Số là lẻ"]

```

d. Pseudocode:

```

counter = 0

WHILE counter < 1000 DO

Output "Hello World!"

counter = counter + 1

ENDWHILE

```

Flowchart:

```

[Start] --> [counter = 0] --> [counter < 1000?] --(Yes)--> [Output "Hello World!"] --> [counter = counter + 1] --> [counter < 1000?]

| |

+--------------------------------------------(No)----------------------------------+

```

Lưu ý: Các pseudocode và flowchart trên chỉ là ví dụ và không chứa cú pháp ngôn ngữ lập trình cụ thể. Chúng chỉ mô tả logic và dòng chảy của các bài toán tương ứng.

**6. So sánh ưu nhược điểm của pseudocode và flowchart**

Ưu điểm của Pseudocode:

1. Độ linh hoạt: Pseudocode không bị ràng buộc bởi ngôn ngữ lập trình cụ thể, cho phép lập trình viên diễn đạt thuật toán một cách tự nhiên và linh hoạt.

2. Dễ đọc và hiểu: Pseudocode sử dụng ngôn ngữ tự nhiên, gần gũi với con người, giúp lập trình viên và người đọc dễ dàng hiểu thuật toán mà không cần kiến thức chuyên sâu về ngôn ngữ lập trình.

3. Dễ dùng và thay đổi: Pseudocode cho phép lập trình viên dễ dàng chỉnh sửa và thay đổi thuật toán mà không cần phải mắc bẫy của cú pháp ngôn ngữ lập trình cụ thể.

4. Tích hợp tốt với tài liệu: Pseudocode có thể được sử dụng để mô tả thuật toán trong tài liệu kỹ thuật, sách giáo trình hoặc tài liệu hướng dẫn mà không cần sử dụng ngôn ngữ lập trình thực tế.

Nhược điểm của Pseudocode:

1. Mơ hồ và không chính xác: Pseudocode có thể gây hiểu lầm nếu không được mô tả rõ ràng và chính xác. Nếu không được sử dụng một cách cẩn thận, nó có thể dẫn đến sự hiểu lầm trong việc triển khai thuật toán thực tế.

2. Không chạy được trực tiếp: Pseudocode là một mô tả trừu tượng của thuật toán và không thể chạy trực tiếp trên máy tính. Nó chỉ là một công cụ để mô tả logic và dòng chảy của thuật toán.

Ưu điểm của Flowchart:

1. Trực quan và dễ hiểu: Flowchart sử dụng biểu đồ hình vẽ, giúp người đọc dễ dàng nhìn thấy luồng dòng chảy và các quyết định trong thuật toán.

2. Mô phỏng dễ dàng: Các biểu đồ trong flowchart có thể được mô phỏng dễ dàng bằng các công cụ và phần mềm chuyên dụng, giúp kiểm tra và thử nghiệm thuật toán trước khi triển khai thực tế.

3. Giao tiếp và trao đổi ý tưởng: Flowchart là công cụ hữu ích để giao tiếp và trao đổi ý tưởng thuật toán giữa các thành viên trong nhóm phát triển.

Nhược điểm của Flowchart:

1. Cấu trúc phức tạp: Flowchart có thể trở nên phức tạp và rối rắm nếu thuật toán có nhiều điều kiện, lặp lại hoặc dòng chảy phức tạp.

2. Phụ thuộc vào ngôn ngữ hình vẽ: Flowchart yêu cầu người đọc biết đọc và hiểu các biểu đồ hình vẽ, đòi hỏi sự quen thuộc với các ký hiệu và quy ước sử dụng trong flowchart.

3. Không linh hoạt: Flowchart thường được vẽ dựa trên quy tắc và cú pháp cố định, không linh hoạt như pseudocode. Việc thay đổi và chỉnh sửa flowchart có thể tốn thời gian và công sức.