

CẤU TRÚC DỮ LIỆU GIẢI THUẬT

Câu 001: Giải thuật là ... câu lệnh chặt chẽ, rõ ràng và xác định một trình tự các thao tác trên các đối tượng dữ liệu.

A. một B. hai **C. dãy.**

Câu 002: Sau một số ... bước thực hiện giải thuật cho chúng ta đạt được kết quả mong muốn.

A.vô hạn B. giới hạn

C. hữu hạn D.thao tác

Câu 003: Đánh giá độ phức tạp của giải thuật là việc xác định ... và ... mà giải thuật cần để thực hiện giải một bài toán.

A. Khoảng thời gian, độ khó.

B. Khoảng thời gian, độ khó.

C. Khoảng thời gian, dung lượng bộ nhớ máy tính.

D. Độ khó, dung lượng bộ nhớ máy tính.

Câu 004: Các kiểu dữ liệu cơ bản là ...

A. các kiểu dữ liệu mà người lập trình được cung cấp sẵn từ máy tính

B. các kiểu dữ liệu mà người lập trình được cung cấp sẵn từ ngôn ngữ tự nhiên

C. các kiểu dữ liệu mà người lập trình được cung cấp sẵn từ ngôn ngữ lập trình.

D. các kiểu dữ liệu mà người lập trình được cung cấp sẵn từ ngôn ngữ máy

Câu 005: Chỉ ra kiểu dữ liệu cơ bản

A. Sinh viên

B. float.

C. Họ tên.

D. Ngày sinh.

Câu 006: Chỉ ra kiểu dữ liệu không cơ bản

A. Char.

B. int.

C. long.

D. struct.

Câu 007: Kiểu dữ liệu trừu tượng là ...

A. kiểu dữ liệu mà người lập trình phải tự xây dựng không dựa trên các kiểu dữ liệu cơ bản được cung cấp từ ngôn ngữ lập trình.

B. kiểu dữ liệu mà người lập trình phải tự xây dựng dựa trên các kiểu dữ liệu không cơ bản được cung cấp từ ngôn ngữ lập trình.

C. kiểu dữ liệu mà người lập trình phải tự xây dựng dựa trên các kiểu dữ liệu cơ bản được cung cấp từ ngôn ngữ máy.

D. kiểu dữ liệu mà người lập trình phải tự xây dựng dựa trên các kiểu dữ liệu cơ bản được cung cấp từ ngôn ngữ lập trình.

Câu 008: Chỉ ra kiểu dữ liệu trừu tượng

A. float.

B. int.

C. char.

D. mảng 1 chiều.

Câu 009: Chỉ ra kiểu dữ liệu cơ bản

A. mảng 1 chiều.

B. mảng 2 chiều.

C. struct.

D. pointer.

Câu 010: Cấu trúc dữ liệu là ...

A. cách lưu trữ dữ liệu trong bộ nhớ máy tính (ROM), sao cho nó có thể được sử dụng một cách hiệu quả.

B. cách lưu trữ dữ liệu trong bộ nhớ máy tính (HDD), sao cho nó có thể được sử dụng một cách hiệu quả.

C. cách lưu trữ dữ liệu trong bộ nhớ máy tính (USB), sao cho nó có thể được sử dụng một cách hiệu quả.

D. cách lưu trữ dữ liệu trong bộ nhớ máy tính (RAM), sao cho nó có thể được sử dụng một cách hiệu quả.

Câu 011: Mối quan hệ giữa cấu trúc dữ liệu và giải thuật có thể minh họa bằng đẳng thức

A. Chương trình = Cấu trúc dữ liệu

B. Giải thuật + Chương trình = Cấu trúc dữ liệu

C. Cấu trúc dữ liệu + Chương trình = Giải thuật.

D. Cấu trúc dữ liệu + Giải thuật = Chương trình.

Câu 012: Phát biểu sau đúng hay sai: Khi cấu trúc dữ liệu thay đổi thì giải thuật cũng thay đổi theo

A. Đúng. B. Sai.

Câu 013: Phát biểu sau đúng hay sai: khi nói tới dữ liệu thì cũng phải xem xét dữ liệu đó cần được thực hiện bằng giải thuật gì để đạt được kết quả mong muốn

A. Đúng. B. Sai.

Câu 014: Phát biểu sau đúng hay sai: Khi nói tới giải thuật phải xem xét nó sẽ tác động trên dữ liệu nào

A. Đúng. B. Sai.

Câu 015: Phát biểu sau đúng hay sai: Giải thuật thể hiện hành động của các bước để giải bài toán

A. Đúng. B. Sai.

Câu 016: Phát biểu sau đúng hay sai: Dữ liệu là đối tượng được xử lý, nó biểu diễn các thông tin cần thiết cho bài toán: dữ liệu vào, dữ liệu ra.

A. Đúng. B. Sai.

Câu 017: Chọn phát biểu đúng

A. Một đối tượng được gọi là đệ quy, nếu nó không được mô tả thông qua định nghĩa của chính nó.

B. Một đối tượng được gọi là đệ quy, nếu nó được mô tả thông qua định nghĩa của đối tượng khác.

C. Một đối tượng được gọi là đệ quy, nếu nó được mô tả thông qua định nghĩa của chính nó.

Câu 018: Chọn phát biểu đúng

A. Đối tượng đệ quy được định nghĩa từ những khái niệm đơn giản nhất cùng dạng với nó.

B. Đối tượng đệ quy được định nghĩa bằng những khái niệm đơn giản nhất cùng dạng với nó.

C. Đối tượng đệ quy được định nghĩa bằng cách quy nạp từ những khái niệm đơn giản nhất cùng dạng với nó.

Câu 019: Chỉ ra khái niệm có tính chất đệ quy

A. khái niệm số tự nhiên.

B. khái niệm hình ảnh.

C. khái niệm âm thanh.

Câu 020: Chỉ ra khái niệm có tính chất đệ quy

A. khái niệm tính giai thừa của n ($n!$).

B. khái niệm hình ảnh.

C. khái niệm âm thanh.

Câu 021: Phát biểu sau đúng hay sai: Trong lập trình, giải thuật đệ quy được sử dụng để xây dựng hàm đệ quy.

A. Đúng. B. Sai.

Câu 022: Phát biểu sau đúng hay sai: Hàm đệ quy là hàm mà trong thân hàm có lời gọi hàm đến chính nó

A. Đúng. B. Sai.

Câu 023: Phát biểu sau đúng hay sai: Chương trình đệ quy là những chương trình máy tính có sử dụng giải thuật đệ quy

A. Đúng. B. Sai.

Câu 024: Phát biểu sau đúng hay sai: Khi có lời gọi đệ quy, trạng thái hiện thời của chương trình (giá trị hiện thời của các biến, điểm ngắt thực hiện của chương trình) được lưu vào vùng bộ nhớ ngăn xếp.

A. Đúng. B. Sai.

Câu 025: Phát biểu sau đúng hay sai: Khi hết lời gọi đệ quy, chương trình chưa kết thúc. Chương trình được tiếp tục thực hiện từ "điểm ngắt" với những giá trị của các biến ở thời điểm ngắt

A. Đúng. B. Sai.

Câu 026: Cho bài toán: cho trước n là một số tự nhiên, tính $n!$. Chỉ ra trường hợp suy biến

A. với $n = 0$

B. với $n = 1$.

C. với $n = 0$ hoặc $n = 1$.

Câu 027: Cho bài toán: cho trước n là một số tự nhiên, tính $n!$. Chỉ ra trường hợp gọi đệ quy

A. với $n = 0$

B. với $n = 1$.

C. với $n=0$ hoặc $n=1$. **D. với $n > 1$**

Câu 028: Cho hàm tính giai thừa như sau:

```
long gthua(int n)
{
    if(n==0) return 1;
    else ... ;
}
```

Chọn lệnh điền vào vị trí...

A. return $n * gthua(n)$. B. return $n * gthua(n+1)$.

C. return $n * gthua(n*1)$. **D. return $n * gthua(n-1)$.**

:: Câu 029: Cho hàm tính giai thừa như sau:

```
long gthua(int n)
{
    if(n==0) return ...;
    else return  $n * gthua(n-1)$ ;
}
```

Chọn lệnh điền vào vị trí...

A. gthua(n). B. 2. C. 0. **D. 1.**

Câu 030: Cho chương trình sau:

<pre>#include<stdio.h> #include<conio.h> long gthua(int n) { if(n==0) return 1; else return $n * gthua(n-1)$; }</pre>	<pre>int main() { int n; long kq; printf("Nhap n = "); scanf("%d",&n); ... printf("%d!=%ld",n,kq); getch(); }</pre>
--	---

Điền lệnh còn thiếu vào dấu ...

A. $kq=gthua(1)$. B. $kq=gthua(1);$. C. $kq=gthua(n)$. **D. $kq=gthua(n);$.**

:: Câu 031: Dãy số Fibonacci là dãy số tự nhiên: 1 1 2 3 5 8 13 ...

Sử dụng giải thuật đệ quy để tính giá trị của số fibonacci thứ n. Chỉ ra trường hợp suy biến.

A. $n=1$. B. $n=2$. C. $n < 2$. **D. $n < 3$.**

Câu 032: Dãy số Fibonacci là dãy số tự nhiên: 1 1 2 3 5 8 13 ...

Sử dụng giải thuật đệ quy để tính giá trị của số fibonacci thứ n. Chỉ ra trường hợp gọi đệ quy

A. $n \geq 1$. B. $n \geq 2$. **C. $n \geq 3$.** D. $n \geq 4$.

Câu 033: Cho hàm tính Fibonacci như sau:

```
int fibo(int n)
{
    if(n<3) return ...;
    else return fibo(n-1)+fibo(n-2);
}
```

Điền vào dấu ...

A. 0. **B. 1.** C. 2. D. 3.

Câu 034: Cho hàm tính Fibonacci như sau:

```
int fibo(int n)
{
    if(n<3) return 1;
    else ...
}
```

Điền vào dấu ...

A. return fibo(n-1)+fibo(n-2)

B. return fibo(n-1)+fibo(n-2);

C. return fibo(n-2)+fibo(n-3)

D. return fibo(n-2)+fibo(n-3);

Câu 035: Cho hàm tính Fibonacci như sau:

```
int fibo(int n)
{
    if( ... ) return 1;
    else return fibo(n-1)+fibo(n-2);
}
```

Điền vào dấu ...

A. $n < 0$

B. $n < 1$.

C. $n < 2$.

D. $n < 3$.

Câu 036: Cho hàm tính giai thừa như sau:

```
long gthua(int n)
{
    if( ... ) return 1;
    else return n*gthua(n-1);
}
```

Chọn lệnh điền vào vị trí ...

A. $n == 0$

B. $n == 1$

C. $n == 2$

D. $n == 3$

Câu 037: Giả sử có hàm tính số Fibonacci là fibo(int n), cho biết lệnh nào đúng khi tính giá trị Fibonacci của n và gán cho biến kq?

A. $kq == \text{fibo}(n)$;

B. $kq != \text{fibo}(n)$;

C. $kq = \text{fibo}(n)$;

D. $kq = \text{fibo}(n)$;

Câu 038: Giả sử có hàm tính số Fibonacci là fibo(int n), cho biết lệnh nào đúng khi tính giá trị Fibonacci của n và gán cho biến kq?

A. $kq == \text{fibo}(n)$;

B. $kq = \text{fibo } f(n)$;

C. $kq = \text{fibo } n(n)$;

D. $kq = \text{fibo}(n)$;

Câu 039: Bài toán tháp Hà Nội được phát biểu như sau:

Input: có 3 cái cọc và n cái đĩa xếp tại cọc 1

Output: Chỉ ra các bước thực hiện di chuyển n cái đĩa từ cọc 1 sang cọc

Nếu sử dụng giải thuật đệ quy, chỉ ra trường hợp suy biến

A. $n = 0$

B. $n = 1$

C. $n = 2$

D. $n = 3$

Câu 040: Bài toán tháp Hà Nội được phát biểu như sau:

Input: có 3 cái cọc và n cái đĩa xếp tại cọc 1

Output: Chỉ ra các bước thực hiện di chuyển n cái đĩa từ cọc 1 sang cọc

Nếu sử dụng giải thuật đệ quy, chỉ ra trường hợp gọi đệ quy

A. $n > 0$ **B. $n > 1$** C. $n > 2$ D. $n > 3$

Câu 041: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        ...
        dichuyen(n-1,T,N,D);
    }
}
```

Điền lệnh còn thiếu vào dấu ...

A. dichuyen(1,N,T,D); B. dichuyen(1,N,T,D)

C. dichuyen(1,N,D,T); D. dichuyen(1,N,T,T);

Câu 042: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        ...
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Điền lệnh còn thiếu vào dấu ...

A. dichuyen(n,N,D,T); B. dichuyen(1,N,D,T);

C. dichuyen(n-1,N,D,T); D. dichuyen(n+1,N,D,T);

Câu 043: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        ...
    }
}
```

Điền lệnh còn thiếu vào dấu ...

- A. dichuyen(n,T,N,D); B. dichuyen(1,T,N,D);
- C. dichuyen(n+1,T,N,D); **D. dichuyen(n-1,T,N,D);**

Câu 044: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if( ... ) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Điền lệnh còn thiếu vào dấu ...

- A. n==1** B. n==2
- C. n==3 D. n==n

Câu 045: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc A sang cọc B với cọc C là trung gian, lệnh gọi hàm nào là đúng?

- A. dichuyen(n,"A","C","B");**

B. `dichuyen(n,"A","B","C");`

C. `dichuyen(n,"A","B","C")`

D. `dichuyen(n,"B","A","C");`

Câu 046:: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc B sang cọc A với cọc C là trung gian, lệnh gọi hàm nào

là đúng?

A. `dichuyen(n,"A","C","B");`

B. `dichuyen(n,"B","C","A");`

C. `dichuyen(n,"A","B","C")`

D. `dichuyen(n,"B","A","C");`

Câu 047: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc A sang cọc C với cọc B là trung gian, lệnh gọi hàm nào là đúng?

A. `dichuyen(n,"A","C","B");`

B. `dichuyen(n,"A","B","C");`

C. `dichuyen(n,"A","B","C")`

D. `dichuyen(n,"B","A","C");`

Câu 048: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc C sang cọc A với cọc B là trung gian, lệnh gọi hàm nào là đúng?

- A. **dichuyen(n,"C","B","A");** B. dichuyen(n,"A","B","C");
- C. dichuyen(n,"A","B","C") D. dichuyen(n,"B","A","C");

Câu 049: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc B sang cọc C với cọc A là trung gian, lệnh gọi hàm nào là đúng?

- A. dichuyen(n,"A","C","B"); B. dichuyen(n,"A","B","C");
- C. dichuyen(n,"A","B","C") D. **dichuyen(n,"B","A","C");**

Câu 050: Cho hàm dịch chuyển giải bài toán tháp Hà Nội như sau:

```
void dichuyen(int n, char N, char T, char D)
{
    if(n==1) printf("%c%s%c",N,"-->",D);
    else
    {
        dichuyen(n-1,N,D,T);
        dichuyen(1,N,T,D);
        dichuyen(n-1,T,N,D);
    }
}
```

Để dịch chuyển các đĩa từ cọc C sang cọc B với cọc A là trung gian, lệnh gọi hàm nào là đúng?

- A. **dichuyen(n,"C","A","B");**
- B. dichuyen(n,"A","B","C");
- C. dichuyen(n,"A","B","C")
- D. dichuyen(n,"B","A","C");

Câu 051: Phát biểu sau đúng hay sai “Mô hình toán học của danh sách là một tập hữu hạn các phần tử có cùng kiểu dữ liệu”.

A. Đúng B. Sai

Câu 052: Phát biểu sau đúng hay sai “Danh sách được biểu diễn bằng một dãy các phần tử: a_1, a_2, \dots, a_n với $n \geq 0$, nếu $n=0$ ta nói danh sách rỗng”

A. Đúng B. Sai

Câu 053: Phát biểu sau đúng hay sai “Nếu $n > 0$ ta gọi a_1 là phần tử đầu tiên và a_n là phần tử cuối cùng của danh sách”.

A. Đúng B. Sai

Câu 054: Phát biểu sau đúng hay sai “Độ dài của danh sách là số phần tử của danh sách”.

A. Đúng B. Sai

Câu 055: Phát biểu sau đúng hay sai “Các phần tử của danh sách có thứ tự tuần tự (linear) theo vị trí xuất hiện của chúng”.

A. Đúng B. Sai

Câu 056: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Khởi tạo danh sách rỗng (không có phần tử): `init(list)`. Tức là danh sách có độ dài `len=0`.

B. Khởi tạo danh sách rỗng (không có phần tử): `init(list)`. Tức là danh sách có độ dài `len=1`.

C. Khởi tạo danh sách rỗng (không có phần tử): `init(list)`. Tức là danh sách có độ dài `len=n`.

Câu 057: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Kiểm tra danh sách có rỗng các phần tử hay không? `empty(list)`. Tức là độ dài thực sự `len=1`?

B. Kiểm tra danh sách có rỗng các phần tử hay không? `empty(list)`. Tức là độ dài thực sự `len=0`?

C. Kiểm tra danh sách có rỗng các phần tử hay không? `empty(list)`. Tức là độ dài thực sự `len=n`?

Câu 058: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Kiểm tra danh sách có đầy các phần tử hay không? `full(list)`. Tức là độ dài thực sự của danh sách `len=n`?

B. Kiểm tra danh sách có đầy các phần tử hay không? `full(list)`. Tức là độ dài thực sự của danh sách `len=0`?

C. Kiểm tra danh sách có đầy các phần tử hay không? `full(list)`. Tức là độ dài thực sự của danh sách `len=1`?

Câu 059: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Tạo danh sách: `make(list)`. Tức là nhập giá trị cho các phần tử trong danh sách

B. Tạo danh sách: `make(list)`. Tức là xuất giá trị cho các phần tử trong danh sách

C. Tạo danh sách: `make(list)`. Tức là in giá trị cho các phần tử trong danh sách

Câu 060: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Duyệt danh sách: `view(list)`. Tức là in giá trị của các phần tử trong danh sách ra màn hình

B. Duyệt danh sách: `view(list)`. Tức là đổi giá trị của các phần tử trong danh sách

C. Duyệt danh sách: view(list). Tức là nhập giá trị của các phần tử trong danh sách

Câu 061: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Bổ sung một phần tử x vào danh sách list tại vị trí thứ p (với $p=1..n$): insert(x,p,list).

B. Bổ sung một phần tử x vào danh sách list tại vị trí thứ p (với $p=1..i$): insert(x,p,list).

C. Bổ sung một phần tử x vào danh sách list tại vị trí thứ p (với $p=1..1$): insert(x,p,list).

D. Bổ sung một phần tử x vào danh sách list tại vị trí thứ p (với $p=1..len$): insert(x,p,list).

Câu 062: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Xóa một phần tử tại vị trí thứ p ($p=1..len$) ra khỏi danh sách list: delete(n,list).

B. Xóa một phần tử tại vị trí thứ p ($p=1..len$) ra khỏi danh sách list: delete(1,list).

C. Xóa một phần tử tại vị trí thứ p ($p=1..len$) ra khỏi danh sách list: delete(len,list).

D. Xóa một phần tử tại vị trí thứ p ($p=1..len$) ra khỏi danh sách list: delete(p,list).

Câu 063: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Lấy giá trị của phần tử thứ p ($p=1..len$) từ danh sách list: getv(1,list)

B. Lấy giá trị của phần tử thứ p ($p=1..len$) từ danh sách list: getv(n,list)

C. Lấy giá trị của phần tử thứ p ($p=1..len$) từ danh sách list: getv(len,list)

D. Lấy giá trị của phần tử thứ p ($p=1..len$) từ danh sách list: getv(p,list)

Câu 064: Kí hiệu danh sách là list, vị trí của phần tử trong danh sách là p (position), độ dài tối đa của danh sách n (số lượng phần tử lớn nhất của list), độ dài thực sự của danh sách là len (length). Chọn phát biểu đúng.

A. Lấy vị trí của phần tử có giá trị x trong danh sách list: getp(p,list)

B. Lấy vị trí của phần tử có giá trị x trong danh sách list: getp(x,list)

C. Lấy vị trí của phần tử có giá trị x trong danh sách list: getp(i,list)

D. Lấy vị trí của phần tử có giá trị x trong danh sách list: getp(n,list)

Câu 065: Khai báo kiểu danh sách chứa tối đa 100 số nguyên

```
#define n 100 typedef struct { ... } danhsach;
```

Điền lệnh vào dấu ...

A. int len; int v[10];

B. int len; int v[1];

C. int len; int v[n]

D. int len; int v[n];

Câu 066: Khởi tạo danh sách mảng rỗng

```
void init(danhsach *list) { ... }
```

Điền lệnh vào dấu ...

A. list->len=0;

B. list->len=1;

C. list->len=10;

D. list->len=100;

Câu 067: Kiểm tra danh sách mảng có rỗng hay không

```
int empty(danhsach *list) { ... }
```

Điền lệnh vào dấu ...

A. if(list->len==0) return 1; else return 0

B. if(list->len==0) return 1; else return n;

C. if(list->len==0) return 1; else return 1;

D. if(list->len==0) return 1; else return 0;

Câu 068: Kiểm tra danh sách mảng có đầy hay không

```
int full(danhsach *list) { ... }
```

Điền lệnh vào dấu ...

A. return (list->len==n);

B. return (list->*len==n);

C. return (list*->len==n);

D. return (list->len*==n);

Câu 069: Tạo danh sách mảng: Hàm này thực hiện nhập giá trị cho các phần tử trong danh sách từ bàn phím.

Điền lệnh vào dấu ...

A. list->v[list->len]==so;

B. list->v[list->len] <- so;

C. list->v[list->len]=so

D. list->v[list->len]=so;

Câu 070: Duyệt danh sách mảng:

```
void view(danhsach list) { int i; ... printf("%6d",list.v[i]); }
```

Điền lệnh vào dấu ...

A. for(i=0;i>list.len;i++)

B. for(i=0;i<list.len;i+)

C. for(i=0;i>list.len;i+)

D. for(i=0;i<list.len;i++)

Câu 071: Bổ sung thêm phần tử x vào vị trí thứ p của danh sách list Thuật toán:

B1 Nhập giá trị cho x, p từ bàn phím

B2 Kiểm tra giá trị của p nếu hợp lệ (p=1..len)

B3 Gán giá trị x vào vị trí thứ p-1 trên danh sách.

B4 Dịch chuyển các phần tử của danh sách từ vị trí thứ p..len sang phải 1 vị trí

B5 Tăng độ dài thực sự (biến len) của danh sách thêm 1

Chọn thứ tự đúng

A. B1 B2 B3 B4 B5

B. B1 B2 B5 B4 B3

C. B1 B4 B3 B2 B5

D. B1 B2 B4 B3 B5

Câu 072: Bổ sung thêm phần tử x vào vị trí thứ p của danh sách

```
list void insert(int x, int p, danhsach *list)
{
    int i; if(p<1 || p>list->len) Thông báo p
    không hợp lệ;
    else if(full(list)) Thông báo danh sách đã
    đầy;
    else
    {
        for(i=list->len;i>=p;i--) list->v[i]=list->v[i-1];
        ...
    }
}
```

Điền lệnh vào dấu ...

A. list->v[p-1]=x; list->len++;

B. list->v[p-1]=x; list->len;

C. list->v[p-1]=x; list->len+;

D. list->v[p-1]=x; list->len++

Câu 073: Bổ sung thêm phần tử x vào vị trí thứ p của danh sách

```
list void insert(int x, int p, danh sach *list)
{
    int i; if(p<1 || p>list->len) Thông báo p không
    hợp lệ;
    else if(full(list)) Thông báo danh sách đã đầy;
    else
    {
        for(i=list->len; i>=p; i--) ...
        list->v[p-1]=x; list->len++;
    }
}
```

Điền lệnh vào dấu ...

A. list->v[i]=list->v[i+1];

B. list->v[i]=list->v[i+2];

C. list->v[i]=list->v[i-1];

D. list->v[i]=list->v[i];

Câu 074: Bổ sung thêm phần tử x vào vị trí thứ p của danh sách

```
list void insert(int x, int p, danh sach *list)
{
    int i; ... Thông báo p không hợp lệ;
    else if(full(list)) Thông báo danh sách đã
    đầy;
    else
    {
        for(i=list->len; i>=p; i--) list->v[i]=list->v[i-1];
        list->v[p-1]=x; list->len++;
    } }
```

Điền lệnh vào dấu ...

A. if(p>1 || p>list->len)

B. if(p<len || p>list->len)

C. if(p<1 || p>list->len)

D. if(len<1 || p>list->len)

Câu 075: Bổ sung thêm phần tử x vào vị trí thứ p của danh sách

```
list void insert(int x, int p, danh sach *list)
{
    int i; if(p<1 || p>list->len) Thông báo p không
    hợp lệ;
    else ... Thông báo danh sách đã đầy;
    else
    {
        for(i=list->len; i>=p; i--) list->v[i]=list->v[i-1];
        list->v[p-1]=x; list->len++;
    } }
```

Điền lệnh vào dấu ...

A. if(full(len))

B. if(full(list))

C. if(full(p))

D. if(full(x))

Câu 076: Xóa 1 phần tử thứ p của danh sách list Thuật toán:

- 1 Nhập giá trị cho p từ bàn phím
- 2 Giảm độ dài của danh sách đi 1
- 3 Kiểm tra nếu danh sách không rỗng
- 4 Dịch chuyển các phần tử của danh sách từ vị trí thứ p-1..len-1 sang trái 1 vị trí
- 5 Giá trị phần tử tại vị trí p trên danh sách đã bị ghi đè giá trị (bị xóa)
- 6 Kiểm tra giá trị của p nếu hợp lệ (p=1..len)

Chọn thứ tự đúng

- A. 123456 **B. 163452**
C. 123546 D. 134652

Câu 077: Xóa phần tử thứ p của danh sách list

```
void delete(int p, danhsach *list) {  
    int i; if(p<1 || p>list->len) Thông báo p không hợp  
    lệ;  
    else if(empty(list)) Thông báo danh sách rỗng;  
    else  
    {  
        for(i=p-1; i<list->len; i++) list->v[i]=list->v[i+1];  
        ...  
    } }  

```

Điền lệnh vào dấu ...

- A. list->len-- **B. list->len--;**
C. list->len++; D. list->len++

Câu 078: Xóa phần tử thứ p của danh sách list

```
void delete(int p, danhsach *list)  
{  
    int i; ... Thông báo p không hợp lệ;  
    else if(empty(list)) Thông báo danh sách rỗng;  
    else  
    {  
        for(i=p-1; i<list->len; i++) list->v[i]=list->v[i+1];  
        list->len--;  
    } }  

```

Điền lệnh vào dấu ...

- A. if(p>1 || p>list->len)
B. if(p<1 || p<list->len)
C. if(p>1 || p<list->len)

D. if(p<1 || p>list->len)

Câu 079: Xóa phần tử thứ p của danh sách list

```
void delete(int p, danhsach *list) {  
    int i; if(p<1 || p>list->len) Thông báo p  
    không hợp lệ;  
    else ... Thông báo danh sách rỗng;  
    else  
    {  
        for(i=p-1; i<list->len; i++) list->v[i]=list->  
        v[i+1];  
        list->len--;  
    } }  

```

Điền lệnh vào dấu ...

- A. if(empty(p))
B. if(empty(len))
C. if(empty(v[len]))
D. if(empty(list))

Câu 080: Xóa phần tử thứ p của danh sách list

```
void delete(int p, danh sach *list) {  
    int i; if(p<1 || p>list->len) Thông báo p không  
    hợp lệ;  
    else if(empty(list)) Thông báo danh sách  
    rỗng;  
    else  
    {  
        for(i=p-1; i<list->len; i++) ...  
        list->len--;  
    } }
```

Điền lệnh vào dấu ...

A. list->v[i]=list->v[i-1]

B. list->v[i]=list->v[i]

C. list->v[i]=list->v[i+1]

D. list->v[i]=list->v[i+1];

Câu 081: Lấy giá trị của phần tử thứ p trong danh sách list. Thuật toán:

1 Nhập giá trị cho p từ bàn phím

2 Kiểm tra giá trị của p nếu hợp lệ (p=1..len)

3 Kiểm tra nếu danh sách không rỗng

4 return v[p-1]

Chọn thứ tự đúng

A.1234

B. 1243

C. 4321

D. 1324

Câu 082: Lấy giá trị của phần tử thứ p trong danh sách

```
list. int getv(int p, danh sach list)  
{  
    ... {  
        Thông báo p không hợp lệ; return -1;}  
    else if(empty(&list))  
    {Thông báo danh sách rỗng; return -2;}  
    else return list.v[p-1];  
}
```

Điền lệnh vào dấu ...

A. if(p<1 || p>list->len)

B. if(p>1 || p>list->len)

C. if(p<1 || p<list->len)

D. if(p<1 || p>list->len)

Câu 083: Lấy giá trị của phần tử thứ p trong danh sách

```
list. int getv(int p, danh sach list)  
{  
    if(p<1 || p>list->len)  
    { ... }  
    else if(empty(&list))  
    {Thông báo danh sách rỗng; return -2;}  
    else return list.v[p-1];  
}
```

Điền lệnh vào dấu ...

A. Thông báo p không hợp lệ; return -1;

B. Thông báo p không hợp lệ; return len;

C. Thông báo p không hợp lệ; return list;

D. Thông báo p hợp lệ; return -1;

Câu 084: Lấy giá trị của phần tử thứ p trong danh sách

```
list. int getv(int p, danh sach list)
{
    if(p<1 || p>list->len)
    {Thông báo p không hợp lệ; return -1;}
    else ...
    {Thông báo danh sách rỗng; return -2;}
    else return list.v[p-1];
}
```

Điền lệnh vào dấu ...

A. if(empty(list)) **B. if(empty(&list))**

C. if(empty(^list)) D. if(empty(-list))

Câu 085: Lấy giá trị của phần tử thứ p trong danh sách

```
list. int getv(int p, danh sach list)
{
    if(p<1 || p>list->len)
    {Thông báo p không hợp lệ; return -1;}
    else if(empty(&list))
    { ... }
    else return list.v[p-1]; }
```

Điền lệnh vào dấu ...

A. Thông báo danh sách rỗng; return -p;

B. Thông báo danh sách rỗng; return -list;

C. Thông báo danh sách rỗng; return -2

D. Thông báo danh sách rỗng; return -2;

Câu 086: Lấy giá trị của phần tử thứ p trong danh sách

```
list. int getv(int p, danh sach list)
{
    if(p<1 || p>list->len)
    {Thông báo p không hợp lệ; return -1;}
    else if(empty(&list))
    {Thông báo danh sách rỗng; return -2;}
    else ... }
```

A. return list.v[p]; B. return list.v[p+1];

C. return list.v[p-1] **D. return list.v[p-1];**

Câu 087: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list.

Thuật toán:

1 Nhập giá trị cho x từ bàn phím

2 Nếu danh sách rỗng - báo lỗi

3 Ngược lại: So sánh giá trị của x với giá trị của các phần tử vi trong danh sách.

4 Nếu bằng nhau thì return i+1, ngược lại thông báo "không tìm thấy"

Chọn thứ tự đúng

A. 4321

B. 3214

C. 2143

D. 1234

Câu 088: Yếu tố nào tác động trực tiếp nhất đến sự phát triển TMĐT (thương mại điện tử) Nhận thức của người dân

A. Cơ sở pháp lý

B. Chính sách phát triển TMĐT

C. Các chương trình đào tạo về TMĐT

Câu 089: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

```
getp(int x, danh sach list)
{
    int i; ...
    if(empty(&list)) { Thông báo danh sách
    rỗng; return -
    2;} else
    {
        for(i=0;i<list.len;i++)
        if(list.v[i]==x)
        {
            p=i; break;
        }
        if(p==-1) printf("Khong tim thay x");
        else return p+1;    } }
```

Điền lệnh vào dấu ...

a. int p=1; b. int p=-1

c. int p=0; **d. int p=-1;**

Câu 090: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

```
getp(int x, danh sach list)
{
    int i; int p=-1;
    if ... { Thông báo danh sách rỗng;
    return -2;} else
    {
        for(i=0;i<list.len;i++)
        if(list.v[i]==x)
        {
            p=i; break;
        }
        if(p==-1) printf("Khong tim thay
        x");
        else return p+1;    } }
```

Điền lệnh vào dấu ...

A. (empty(^list))

B. (empty(-list))

C. (empty(+list))

D. (empty(&list))

Câu 091: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

<pre>getp(int x, danh sach list) { int i; int p=-1; if(empty(&list)) { ... } else { for(i=0;i<list.len;i++) if(list.v[i]==x) { p=i; break; } if(p==-1) printf("Khong tim thay x"); else return p+1; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. Thông báo danh sách đầy; return -2; B. Thông báo danh sách đầy; return -p; C. Thông báo danh sách rỗng; return -2 D. Thông báo danh sách rỗng; return -2;</p>
--	---

Câu 092: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

<pre>getp(int x, danh sach list) { int i; int p=-1; if(empty(&list)) { Thông báo danh sách rỗng; return - 2;} else { for(i=0;i<list.len;i++) ... { p=i; break; } if(p==-1) printf("Khong tim thay x"); else return p+1; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. if(list.v[i]>=x) B. if(list.v[i]=x) C. if(list.v[i]<=x) D. if(list.v[i]==x)</p>
--	---

Câu 093: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

<pre>getp(int x, danh sach list){ int i; int p=-1; if(empty(&list)) { Thông báo danh sách rỗng; return - 2;} else { for(i=0;i<list.len;i++) if(list.v[i]==x) { ... } if(p==-1) printf("Khong tim thay x"); else return p+1; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. p=i; continue; B. p=i; break C. p=i; continue D. p=i; break;</p>
--	--

Câu 094: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

<pre>getp(int x, danh sach list) { int i; int p=-1; if(empty(&list)) { Thông báo danh sách rỗng; return - 2;} else { for(i=0;i<list.len;i++) if(list.v[i]==x) { p=i; break; } ... printf("Khong tim thay x"); else return p+1; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. if(p==i) B. if(p==i) C. if(p==0) D. if(p==-1)</p>
---	---

Câu 095: Lấy vị trí của phần tử có giá trị bằng x trong danh sách list. int

<pre>getp(int x, danh sach list) { int i; int p=-1; if(empty(&list)) { Thông báo danh sách rỗng; return - 2;} else { for(i=0;i<list.len;i++) if(list.v[i]==x) { p=i; break; } if(p==-1) printf("Khong tim thay x"); else ... } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. return p+1 B. return p; C. return p-1; D. return p+1;</p>
---	---

Câu 096: Phát biểu sau đúng hay sai: Ngăn xếp là một kiểu danh sách mà việc thêm 1 phần tử vào danh sách hay lấy 1 phần tử ra khỏi danh sách chỉ được thực hiện tại 1 đầu của danh sách

A. Sai **B. Đúng**

Câu 097: Phát biểu sau đúng hay sai: Vị trí thực hiện việc thêm vào hoặc lấy ra 1 phần tử của danh sách gọi là ĐỈNH của ngăn xếp.

A. Sai **B. Đúng**

Câu 098: Phát biểu sau đúng hay sai: Do các phần tử trong ngăn xếp chỉ được đưa vào hoặc lấy ra ở 1 đầu nên các phần tử đưa vào trước sẽ được lấy ra sau và ngược lại.

A. Sai **B. Đúng**

Câu 099: Phát biểu sau đúng hay sai: Kiểu dữ liệu thích hợp để khai báo ngăn xếp là struct trong C/C++

A. Sai **B. Đúng**

Câu 100: Khai báo kiểu ngăn xếp chứa tối đa 100 số nguyên

#define n 100 typedef struct { ... } stack;

Điền lệnh vào dấu ...

A. int p; int v[1]; B. int p; int v[10];

C. int p; int v[0]; **D. int p; int v[n];**

Câu 101: Phát biểu sau đúng hay sai: Ngăn xếp được coi là rỗng nếu nó không chứa 1 phần tử nào hay vị trí đỉnh ngăn xếp p= -1

A. Sai **B. Đúng**

Câu 102: Khởi tạo ngăn xếp rỗng: Ngăn xếp được coi là rỗng nếu nó không chứa 1 phần tử nào hay vị trí đỉnh ngăn xếp $p = -1$.

```
void init(stack *s) { ... }
```

Điền lệnh vào dấu ...

A. $s \rightarrow p = -0$; B. $s \rightarrow p = -n$;

C. $s \rightarrow p = -p$; **D. $s \rightarrow p = -1$;**

Câu 103: Phát biểu sau đúng hay sai: Ngăn xếp được coi là đầy khi nó đã chứa đầy các giá trị hay giá trị đỉnh ngăn xếp $p = n$.

A. Sai **B. Đúng**

Câu 104: Phát biểu sau đúng hay sai: Hàm `full(...)` so sánh giá trị của đỉnh ngăn xếp p với số lượng phần tử tối đa của ngăn xếp n . Nó trả về 1 nếu ngăn xếp đầy và ngược lại.

A. Sai **B. Đúng**

Câu 105: Kiểm tra ngăn xếp đầy hay không

```
int full(stack s) { }
```

Điền lệnh vào dấu ...

A. `return(s.p==1);` B. `return(s.p==n);`

C. `return(s.p==n-1)` **D. `return(s.p==n-1);`**

Câu 106: Phát biểu sau đúng hay sai: Một ngăn xếp được coi là rỗng khi nó không chứa phần tử nào hay giá trị vị trí đỉnh ngăn xếp $p = -1$

A. Sai **B. Đúng**

Câu 107: Phát biểu sau đúng hay sai: Hàm `empty(...)` so sánh giá trị của đỉnh ngăn xếp p với -1 . Nó trả về 1 nếu ngăn xếp rỗng và ngược lại.

A. Sai **B. Đúng**

Câu 108: Kiểm tra ngăn xếp rỗng hay không

```
int empty(stack s) { ... }
```

Điền lệnh vào dấu ...

A. `return (s.p==1)` B. `return (s.p==1);`

C. `return (s.p==1);` D. `return (s.p==1);`

Câu 109: Phát biểu sau đúng hay sai:

Giải thuật đẩy 1 phần tử vào ngăn xếp:

1. Nhập giá trị cho phần tử x cần đưa vào ngăn xếp.

2. Kiểm tra nếu ngăn xếp chưa đầy, tăng p lên 1, gán phần tử thứ p của ngăn xếp bằng x .

A. Sai **B. Đúng**

Câu 110: Đẩy 1 phần tử vào ngăn xếp

<pre>void push(int x, stack *s){ if(...) { s->p=s->p+1; s->v[s->p]=x; } else Thông báo ngăn xếp đầy; }</pre>	Điền lệnh vào dấu ... A. $s \rightarrow \text{full}(p)$ B. $s \rightarrow \text{full}(p)$ C. $!s \rightarrow \text{full}(p)$ D. $!s \rightarrow \text{full}()$
--	--

Câu 111: Đẩy 1 phần tử vào ngăn xếp

<pre>void push(int x, stack *s) { if(!s->full()) { ... s->v[s->p]=x; }</pre>	Điền lệnh vào dấu ... A. $s \rightarrow p = s \rightarrow p - 1$; B. $s \rightarrow p = s \rightarrow p + 1$ C. $s \rightarrow p = s \rightarrow p - 1$ D. $s \rightarrow p = s \rightarrow p + 1$;
---	--

else Thông báo ngăn xếp đầy; }	
-----------------------------------	--

Câu 112: Đẩy 1 phần tử vào ngăn xếp

<pre>void push(int x, stack *s) { if(!s->full()) { s->p=s->p+1; ... } else Thông báo ngăn xếp đầy; }</pre>	Điền lệnh vào dấu ... A. s->v[s->p]=1; B. s->v[s->p]=100; C. s->v[s->p]=x D. s->v[s->p]=x;
---	---

Câu 113: Phát biểu sau đúng hay sai “Giải thuật lấy 1 phần tử ra khỏi ngăn xếp”

- 1 Kiểm tra nếu ngăn xếp khác rỗng
- 2 Gán giá trị phần tử ở đỉnh ngăn xếp cho x
- 3 Giảm giá trị của p đi 1.

A.Sai **B. Đúng**

Câu 114: Lấy 1 phần tử ra khỏi ngăn xếp

<pre>int pop(stack *s) { int x; if(...) { x =s->v[s->p]; s->p--; return x; } else Thông báo ngăn xếp rỗng; }</pre>	Điền lệnh vào dấu ... A. !empty(p) B. !empty(p); C. !empty(s); D. !empty(s)
---	--

Câu 115: Lấy 1 phần tử ra khỏi ngăn xếp

<pre>int pop(stack *s) { int x; if(!empty(s)) { ... s->p--; return x; } else Thông báo ngăn xếp rỗng; }</pre>	Điền lệnh vào dấu ... A. x=s->v[s->s]; B. x=s->v[s->s] C. x=s->v[s->p] D. x=s->v[s->p];
--	--

Câu 116: Lấy 1 phần tử ra khỏi ngăn xếp

<pre>int pop(stack *s) { int x; if(!empty(s)) {</pre>	Điền lệnh vào dấu ... A. s->p--; return x B. s->p--; return x; C. s->p++; return x D. s->p++; return x;
---	---

<pre>x=s->v[s->p]; ... } else Thông báo ngăn xếp rỗng; }</pre>	
--	--

Câu 117: Phát biểu sau đúng hay sai: Hàng đợi là một kiểu danh sách mà việc đẩy 1 phần tử vào danh sách được thực hiện ở cuối (rear) danh sách và việc lấy 1 phần tử ra khỏi danh sách được thực hiện ở đầu (front) danh sách.

A. Sai **B. Đúng**

Câu 118: Phát biểu sau đúng hay sai: Phần tử nào được đưa vào hàng đợi trước thì được lấy ra trước và ngược lại.

A. Sai **B. Đúng**

Câu 119: Phát biểu sau đúng hay sai: Kiểu dữ liệu thích hợp để cài đặt hàng đợi là struct trong C/C++.

A. Sai **B. Đúng**

Câu 120: Khai báo kiểu hàng đợi tối đa 100 số nguyên

```
#define n 100 typedef struct { ... } queue;
```

Điền lệnh vào dấu ...

A. int rear, front, v[1];

B. int rear, front, v[n];

C. int rear, front, v[10];

D. int rear, front, v[1000];

Câu 121: Phát biểu sau đúng hay sai: Hàng đợi được coi là rỗng nếu nó không chứa phần tử nào hay giá trị của biến rear=front=-1.

A. Sai **B. Đúng**

Câu 122: Khởi tạo hàng đợi rỗng: Hàng đợi được coi là rỗng nếu nó không chứa phần tử nào hay giá trị của biến rear=front=-1

```
void init(queue *q) { }
```

Điền lệnh vào dấu ...

A. q->rear=q->front=-0;

B. q->rear=q->front=-1;

C. q->rear=q->front=-10;

D. q->rear=q->front=-100;

Câu 123: Phát biểu sau đúng hay sai: Hàng đợi được coi là đầy khi giá trị của biến rear=n.

A. Sai **B. Đúng**

Câu 124: Kiểm tra hàng đợi đầy: Hàng đợi được coi là đầy khi giá trị của biến rear=n int full(queue *q) { return ... }

Điền lệnh vào dấu ...

A. (q->rear==1);

B. (q->rear==n);

C. (q->rear==100);

D. (q->rear==n)

Câu 125: Phát biểu sau đúng hay sai: Hàng đợi được coi là rỗng khi giá trị của biến front=rear

A. Sai **B. Đúng**

Câu 126: Kiểm tra hàng đợi rỗng: empty(...) • Hàng đợi được coi là rỗng khi giá trị của biến front=rear int empty(queue *q) { return (q->front==q->rear); }

A. Sai **B. Đúng**

Câu 127: Kiểm tra hàng đợi rỗng: Hàng đợi được coi là rỗng khi giá trị của biến front=rear

```
int empty(queue *q) { return ... }
```

Điền lệnh vào dấu ...

A. (q->front==q->rear)

B. (q->front==q->rear);

C. (q->front==q->0);

D. (q->front==q->0)

Câu 128: Phát biểu sau đúng hay sai: Giải thuật đẩy 1 phần tử vào hàng đợi:

1 Kiểm tra nếu hàng đợi chưa đầy

2 Tăng giá trị của rear lên 1, gán giá trị của phần tử thứ rear bằng x

A. Sai **B. Đúng**

Câu 129: Đẩy 1 phần tử vào hàng đợi

<pre>void insert(int x, queue *q) { if(...) { q ->rear++; q->v[q->rear]=x; } else Thông báo hàng đợi đầy. }</pre>	Điền lệnh vào dấu ... A. !full(x) B. !full(q) C. !full(rear) D. !full(v)
--	---

Câu 130: Đẩy 1 phần tử vào hàng đợi

<pre>void insert(int x, queue *q) { if(!full(q)) { ... q- >v[q->rear]=x; } else Thông báo hàng đợi đầy. }</pre>	Điền lệnh vào dấu ... A. q->rear++ B. q->rear++; C. q->rear--; D. q->rear--
---	---

Câu 131: Phát biểu sau đúng hay sai: Giải thuật lấy 1 phần tử ra khỏi hàng đợi:

1 Kiểm tra nếu hàng đợi chưa rỗng

2 Tăng giá trị của front lên 1, gán x bằng phần tử thứ front trong hàng đợi.

A. Đúng B. Sai

Câu 132: Lấy 1 phần tử ra khỏi hàng đợi:

<pre>int remove(queue *q){ int x; if(...) { q->front++; x=q->v[q->front]; return x; } else Thông báo hàng đợi rỗng; }</pre>	Điền lệnh vào dấu ... A. !empty((q)) B. !empty(q) C. !empty(q)) D. empty(q))
--	---

Câu 133: Lấy 1 phần tử ra khỏi hàng đợi:

<pre>int remove(queue *q) { int x; if(!empty(q)) { ... x= q->v[q->front]; return x; }</pre>	Điền lệnh vào dấu ... A. q->front++ B. q->front++; C. q->front--; D. q->front--
---	---

<pre> } else Thông báo hàng đợi rỗng; } </pre>	
--	--

Câu 134: Lấy 1 phần tử ra khỏi hàng đợi:

<pre> int remove(queue *q) { int x; if(!empty(q)) { q ->front++; ... return x; } else Thông báo hàng đợi rỗng; } </pre>	Điền lệnh vào dấu ... A. $x=q->v[q->front]$ B. $x=q->v[q->front];$ C. $x=q->v[q->rear]$ D. $x=q->v[q->rear];$
--	---

Câu 135: Lấy 1 phần tử ra khỏi hàng đợi:

<pre> int remove(queue *q) { int x; if(!empty(q)) { q ->front++; $x=q->v[q->front];$... } else Thông báo hàng đợi rỗng; } </pre>	Điền lệnh vào dấu ... A. return x B. return x; C. return rear D. return rear;
---	--

Câu 13: Phát biểu sau đúng hay sai: Sắp xếp là quá trình bố trí lại các phần tử của một tập theo thứ tự tăng dần hoặc giảm dần của giá trị các phần tử.

A. Sai **B. Đúng**

Câu 137: Phát biểu sau đúng hay sai: Mục đích chính của sắp xếp là giúp cho việc tìm kiếm trên tập phần tử đó dễ dàng.

A., Sai **B. Đúng**

Câu 138: Phát biểu sau đúng hay sai: Các đối tượng được sắp xếp thường có nhiều thuộc tính.

A. Sai **B. Đúng**

Câu 139: Phát biểu sau đúng hay sai: Khi sắp xếp chúng ta chỉ sắp xếp chúng theo giá trị của thuộc tính khóa.

A. Sai **B. Đúng**

Câu 140: Phát biểu sau đúng hay sai: Phát biểu bài toán sắp xếp

- Input: Dãy số nguyên có n phần tử, các phần tử chưa có thứ tự xác định.
- Output: Dãy số nguyên có các phần tử được sắp xếp theo thứ tự xác định (tăng hoặc giảm dần).

A. Sai **B. Đúng**

Câu 141: Phát biểu sau đúng hay sai: Ý tưởng sắp xếp tăng dần theo phương pháp sắp xếp đối chỗ trực tiếp: Nếu các phần tử của mảng số nguyên a có thứ tự tăng dần theo giá trị thì phần tử a_i luôn là $\min(a_i, a_{i+1}, a_{i+2}, \dots, a_{n-1})$

A. Sai **B. Đúng**

Câu 142: Phát biểu sau đúng hay sai: Thao tác sắp xếp tăng dần theo phương pháp sắp xếp đối chỗ trực tiếp: Dem phần tử đầu tiên trong mảng, so sánh với tất cả các phần tử còn lại, tìm được phần tử nhỏ nhất của mảng, đưa về vị trí đầu tiên trong mảng. Dem phần tử thứ hai trong mảng, so sánh với tất cả các phần tử còn lại, tìm được phần tử nhỏ thứ hai của mảng, đưa về vị trí thứ hai trong mảng. ... Lặp lại công việc trên cho đến khi mảng chỉ còn lại phần tử cuối cùng cần sắp xếp

A.Sai **B. Đúng**

Câu 143: Phát biểu sau đúng hay sai: Ý tưởng sắp xếp giảm dần theo phương pháp sắp xếp đổi chỗ trực tiếp: Nếu các phần tử của mảng số nguyên a có thứ tự tăng dần theo giá trị thì phần tử a_i luôn là $\min(a_i, a_{i+1}, a_{i+2}, \dots, a_{n-1})$

A. Sai B. Đúng

Câu 14: Thuật toán sắp xếp đổi chỗ trực tiếp: Dãy số cần sắp xếp tăng dần là mảng số nguyên a có n phần tử.

B1: Gán $i=0$;

B2: Gán $j=i+1$;

- so sánh a_i với a_j , nếu $a_i > a_j$ thì hoán đổi giá trị của a_j cho a_i .

- Tăng j thêm 1, lặp lại việc so sánh a_i với a_j , dừng lại khi giá trị của $j > n-1$.

B3: -Tăng giá trị của i thêm 1 - Nếu $i < n-1$, lặp lại bước 2, ngược lại kết thúc.

Lựa chọn thứ tự thực hiện đúng

A. B3 B2 B1

B. B2 B3 B1

C. B1 B2 B3

Câu 145: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau:

<pre>void interchange_sort(int a[], int n) { int i,j,tg; for(i=0;...;i++) for(j=i+1;j<n;j++) if(a[i]>a[j]) { tg=a[i]; a[i]=a[j]; a[j]=tg; } }</pre>	Điền lệnh vào dấu ... A. $i < n$ B. $i < n-1$ C. $i < n+1$
---	--

Câu 146: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau:

<pre>void interchange_sort(int a[], int n) { int i,j,tg; for(i=0; i<n-1;i++) for(j=i+1;...;j++) if(a[i]>a[j]) { tg=a[i]; a[i]=a[j]; a[j]=tg; } }</pre>	Điền lệnh vào dấu ... A. $j > n$ B. $j < n$ C. $j < n+1$
--	--

Câu 147: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau:

<pre>void interchange_sort(int a[], int n) { int i,j,tg; for(i=0; i<n-1;i++) for(j=i+1;j<n;j++) if(...) { tg=a[i]; a[i]=a[j];</pre>	Điền lệnh vào dấu ... A. $a[i] < a[j]$ B. $a[i] > a[j]$ C. $a[i] = a[j]$ Đề chỉ nói là sắp xếp, không nói sắp xếp giảm hay tăng nên cả 2 đều đúng!
---	--

<pre>a[j]=tg; } }</pre>	
-------------------------	--

Câu 148: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau:

<pre>void interchange_sort(int a[], int n) { int i,j,tg; for(i=0; i<n-1;i++) for(j=i+1;j<n;j++) if(a[i]>a[j]) { ... a[i]=a[j]; a[j]=tg; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. tg=a[j]; B. tg=a[i]; C. tg=a[i]</p>
--	---

Câu 149: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau:

<pre>void interchange_sort(int a[], int n) { int i,j,tg; for(i=0; i<n-1;i++) for(j=i+1;j<n;j++) if(a[i]>a[j]) { tg=a[i]; ... a[j]=tg; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[i]=a[j] B. a[i]=a[j]; C. a[j]=a[j];</p>
--	---

Câu 150: Cài đặt thuật toán sắp xếp đổi chỗ trực tiếp thành hàm interchange_sort như sau: void

<pre>interchange_sort(int a[], int n) { int i,j,tg; for(i=0; i<n-1;i++) for(j=i+1;j<n;j++) if(a[i]>a[j]) { tg=a[i]; a[i]=a[j]; ... } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[i]=tg; B. a[j]=tg C. a[j]=tg;</p>
---	---

Câu 151: Phát biểu sau đúng hay sai: Thao tác sắp xếp tăng dần theo phương pháp sắp xếp chọn trực tiếp: So sánh giá trị của 1 phần tử trong mảng với các phần tử còn lại, nếu nó lớn hơn thì ghi nhận vị trí của phần tử bé hơn nó, tìm được vị trí của phần tử bé nhất, kí hiệu là min. Hoán đổi giá trị của phần tử này với phần tử có vị trí min. Lặp lại các bước trên cho đến khi phần tử được so sánh là a_{n-1}.

A. Sai **B. Đúng**

Câu 152: Thuật toán: Sắp xếp tăng dần theo phương pháp chọn trực tiếp.

Giả thiết: Dãy số cần sắp xếp tăng dần là mảng số nguyên a có n phần tử.

B1: Gán i=0;

B2: Hoán đổi giá trị của a_{min} và a_i

B3: Gán min=i,

- so sánh phần tử a_{min} với các phần tử a_j với j=i+1..n-1. Nếu a_{min}>a_j thì min=j. B4:

Lặp lại B3 cho đến khi dãy số được sắp xong.

Sắp xếp các bước thực hiện đúng

A. B1 B2 B3 B4

B. B1 B3 B2 B4

C. B1 B4 B2 B3

D. B1 B4 B3 B2

Câu 153: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre>void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;...;i++) { vtmin=i; for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) vtmin=j; tg=a[i]; a[i]=a[vtmin]; a[vtmin]=tg; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. i>n-1 B. i<n-1 C. i<n+1</p>
---	--

Câu 154: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre>void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;i<n-1;i++) { ... for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) vtmin=j; tg=a[i]; a[i]=a[vtmin]; a[vtmin]=tg; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. vtmin=i B. vtmin=i; C. vtmin=i+1; D. vtmin=i+1</p>
---	--

Câu 155: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre>void selection_sort(int a[], int n){ int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; ... if(a[vtmin]>a[j]) vtmin=j; tg=a[i]; a[i]=a[vtmin]; a[vtmin]=tg; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. for(j=i+1;j>n;j++) B. for(j=i+1;j>n;j--) C. for(j=i+1;j<n;j--) D. for(j=i+1;j<n;i++)</p>
---	--

Câu 156: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre>void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; for(j=i+1;j<n;j++) if(...) vtmin=j;</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[vtmin]>a[i] B. a[vtmin]>a[j] C. a[vtmin]<a[j] D. a[vtmin]<a[i] P/s: Đề chỉ nói là sắp xếp, không nói sắp xếp giảm</p>
--	--

<pre> tg=a[i]; a[i]=a[vtmin]; a[vtmin]=tg; } } </pre>	hay tăng nên cả 2 đều đúng!
---	-----------------------------

Câu 157: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre> void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) ... tg=a[i]; a[i]=a[vtmin]; a[vtmin]=tg; } } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. vtmin=i; B. vtmin=j; C. vtmin=j D. vtmin=i</p>
---	---

Câu 158: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre> void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) vtmin=j; ... a[i]=a[vtmin]; a[vtmin]=tg; } } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. tg=a[j]; B. tg=a[i]; C. tg=a[i] D. tg=a[j]</p>
---	---

Câu 159: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre> void selection_sort(int a[], int n){ int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) vtmin=j; tg=a[i]; ... a[vtmin]=tg; } } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[i]=a[vtmin] B. a[i]=a[vtmin]; C. a[i]=a[j]; D. a[i]=a[j]</p>
--	---

Câu 160: Cài đặt thuật toán sắp xếp theo phương pháp chọn trực tiếp thành hàm selection_sort

<pre> void selection_sort(int a[], int n) { int i,j,vtmin,tg; for(i=0;i<n-1;i++) { vtmin=i; for(j=i+1;j<n;j++) if(a[vtmin]>a[j]) </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[vtmin]=tg B. a[vtmin]=tg; C. a[j]=tg; D. a[i]=tg;</p>
--	--

<pre>vtmin=j; tg=a[i]; a[i]=a[vtmin]; ... } }</pre>	
---	--

Câu 161: Phát biểu sau đúng hay sai: Ý tưởng sắp xếp tăng dần theo phương pháp sắp xếp chèn trực tiếp: Giả sử có 1 dãy số gồm n phần tử a_1, a_2, \dots, a_n . Trong đoạn i phần tử đầu tiên của dãy a_1, a_2, \dots, a_i thì các phần tử a_1, a_2, \dots, a_{i-1} đã có thứ tự tăng dần. Cần chèn phần tử a_i vào vị trí thích hợp của đoạn đã có thứ tự ta được đoạn mới a_1, a_2, \dots, a_i vẫn có thứ tự tăng dần

A. Sai **B. Đúng**

Câu 162: Thuật toán: Sắp xếp tăng dần theo phương pháp chèn trực tiếp

B1: Gán $i=1$; // Vì đoạn a_0 đã tăng dần, nên xét từ phần tử a_1

B2: Gán $x=a_i$; $pos=i-1$;

B3- Tìm vị trí chèn phù hợp: Nếu ($pos \geq 0$ và $a_{pos} > x$) thì di chuyển các phần tử từ a_{pos} đến a_{i-1} sang phải 1 vị trí để dành chỗ cho a_i

B4: Gán $a_{pos}=x$, được đoạn a_0, a_1, \dots, a_i tăng dần.

B5: Nếu $i < n$ thì lặp lại bước 2, ngược lại kết thúc.

Chọn thứ tự thực hiện đúng

A. 1-2-3-4-5

B. 1-3-2-4-5

C. 1-2-4-3-5

D. 1-3-4-2-5

Câu 163: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; ... { x =a[i]; pos=i-1; while(pos>=0 && a[pos]>x) { a[pos+1]=a[pos]; pos--; } a[pos+1]=x; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. for($i=1; i < n-1; i++$)</p> <p>B. for($i=1; i < n; i++$)</p> <p>C. for($i=1; i < n+1; i++$)</p>
--	--

Câu 164: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; for($i=1; i < n; i++$) { ... pos=i-1; while(pos>=0 && a[pos]>x) { a[pos+1]=a[pos]; pos--; } a[pos+1]=x; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. $x=a[i-1]$;</p> <p>B. $x=a[i]$;</p> <p>C. $x=a[i+1]$;</p>
--	--

Câu 165: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; for(i=1;i<n;i++) { x =a[i]; ... while(pos>=0 && a[pos]>x) { a[pos+1]=a[pos]; pos--; } a[pos+1]=x; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. pos=i+1; B. pos=i-1; C. pos=i+1 D. pos=i;</p>
---	---

Câu 166: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; for(i=1;i<n;i++) { x =a[i]; pos=i-1; ... { a[pos+1]=a[pos]; pos--; } a[pos+1]=x; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. while(pos>=0 && a[pos]>x) B.while(pos>=0 a[pos]>x) C. while(pos>=0 && a[pos]<x) D. while(pos>=0 a[pos]<x)</p> <p>P/s: Đề chỉ nói là sắp xếp, không nói sắp xếp giảm hay tăng nên cả 2 đều đúng!</p>
--	--

Câu 167: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; for(i=1;i<n;i++) { x =a[i]; pos=i-1; while(pos>=0 && a[pos]>x) { ... pos--; } a[pos+1]=x; } }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[pos-1]=a[pos]; B. a[pos+1]=a[pos]; C. a[pos-1]=a[pos] D. a[pos+1]=a[pos]</p>
---	--

Câu 168: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre>insertion_sort void insertion_sort(int a[], int n) { int x,i,pos;</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. pos-- B. pos--;</p>
--	---

<pre> for(i=1;i<n;i++) { x =a[i]; pos=i-1; while(pos>=0 && a[pos]>x) { a[pos+1]=a[pos]; ... } a[pos+1]=x; } } </pre>	<p>C. pos++ D. pos++;</p>
---	-------------------------------

Câu 169: Cài đặt thuật toán thành hàm sắp xếp theo phương pháp chèn trực tiếp thành hàm

<pre> insertion_sort void insertion_sort(int a[], int n) { int x,i,pos; for(i=1;i<n;i++) { x =a[i]; pos=i-1; while(pos>=0 && a[pos]>x) { a[pos+1]=a[pos]; pos--; } ... } } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[pos-1]=x; B. a[pos+1]=x; C. a[pos-1]=x D. a[pos+1]=x</p>
--	--

Câu 170: Phát biểu sau đúng hay sai: Ý tưởng phương pháp sắp xếp Nổi bọt

- Xuất phát từ cuối dãy số
- So sánh giá trị của hai phần tử liền kề. Nếu thỏa mãn điều kiện:
- Hoán đổi giá trị hai phần tử này cho nhau để đưa phần tử có giá trị nhỏ hơn về đầu dãy hiện thời.
- Không xét phần tử đầu dãy hiện thời trong lần sắp xếp tiếp theo (Lần sắp xếp thứ i thì vị trí đầu dãy hiện thời là i)
- Lặp lại việc sắp xếp cho đến khi không còn cặp phần tử nào cần so sánh

A. Sai **B. Đúng**

Câu 171: Thuật toán: Sắp xếp tăng dần theo phương pháp nổi bọt

B1: Gán i=0;

B2: Gán j=n-1; Nếu

(j<i) thực hiện:

Nếu (a_j<a_{j-1}) thì Hoán đổi giá trị a_j và a_{j-1}

j=j-1;

B3: i=i+1; kiểm tra giá trị của i Nếu i>n-1 thì kết thúc Ngược lại thì lặp lại B2

Chọn thứ tự thực hiện đúng

A. B1 B3 B2 **B. B1 B2 B3**

C. B3 B2 B1 D. B3 B1 B2

Câu 172: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre> void bubble_sort(int a[], int n) { int i,j,tg; ... for(j=n-1;j>i;j--) if(a[j]<a[j-1]) </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. for(i=0;i<n+1;i++) B. for(i=0;i<n-1;i++) C. for(i=0;i<n-1;i--) D. for(i=0;i<n;i++)</p>
--	--

<pre> { tg=a[j]; a[j]=a[j-1]; a[j-1]=tg; } </pre>	
---	--

Câu 173: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre> void bubble_sort(int a[], int n) { int i,j,tg; for(i=0;i<n-1;i++) ... if(a[j]<a[j-1]) { tg=a[j]; a[j]=a[j-1]; a[j-1]=tg; } } </pre>	Điền lệnh vào dấu ... A. for(j=n;j>i;j--) B. for(j=n+1;j>i;j--) C. for(j=n+1;j<i;j--) D. for(j=n-1;j>i;j--)
---	---

Câu 174: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre> void bubble_sort(int a[], int n) { int i,j,tg; for(i=0;i<n-1;i++) for(j=n-1;j>i;j--) ... { tg=a[j]; a[j]=a[j-1]; a[j-1]=tg; } } </pre>	Điền lệnh vào dấu ... A. if(a[j]<a[i-1]) B. if(a[j]<a[j+1]) C. if(a[j]<a[j-1]) D. if(a[j]<a[j])
--	---

Câu 175: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre> void bubble_sort(int a[], int n) { int i,j,tg; for(i=0;i<n-1;i++) for(j=n-1;j>i;j--) if(a[j]<a[j-1]) { ... a[j]=a[j-1]; a[j-1]=tg; } } </pre>	Điền lệnh vào dấu ... A. tg=a[i]; B. tg=a[j+1]; C. tg=a[j]; D. Tg=a[j];
--	--

Câu 176: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre> void bubble_sort(int a[], int n) { int i,j,tg; for(i=0;i<n-1;i++) for(j=n-1;j>i;j--) if(a[j]<a[j-1]) { tg=a[j]; </pre>	Điền lệnh vào dấu ... A. a[j]=a[j+1]; B. a[j]=a[j-1]; C. a[j]=a[j]; D. a[j]=a[j]
---	---

<pre>... a[j-1]=tg; } }</pre>	
-------------------------------	--

Câu 177: Cài đặt thuật toán sắp xếp theo phương pháp nổi bọt thành hàm bubble_sort

<pre>void bubble_sort(int a[], int n) { int i,j,tg; for(i=0;i<n-1;i++) for(j=n-1;j>i;j--) if(a[j]<a[j-1]) { tg=a[j]; a[j]=a[j-1]; } ... }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. a[j-1]=tg; B. a[j-1]=tg C. a[j]=tg; D. a[j]=tg;</p>
--	--

Câu 178: Phát biểu sau đúng hay sai: Bài toán tìm kiếm

- Input: Cho một tập các phần tử hữu hạn.
- Output: Cần chỉ ra những phần tử của tập hợp thỏa mãn điều kiện cho trước.

A. Sai **B. Đúng**

Câu 179: Phát biểu sau đúng hay sai: Ý tưởng phương pháp tìm kiếm Tuần tự

- Duyệt lần lượt các phần tử của dãy số.
- So sánh giá trị của phần tử đang duyệt với giá trị cần tìm. Nếu giá trị bằng nhau là "tìm thấy", ngược lại là "không tìm thấy".

A. Sai **B. Đúng**

Câu 180: Thuật toán tìm kiếm tuần tự:

Tìm phần tử dãy số a có giá trị bằng x nhập từ bàn phím?

B1: Gán kt=0; // biến kiểm tra

B2: Với i=0..n-1, so sánh a[i] với x. Nếu a[i]=x thì kt=1 và kết thúc kiểm tra.

B3: Kiểm tra giá trị của kt. Nếu kt=1 thì thông báo "tìm thấy". Ngược lại, thông báo "không tìm thấy".

Chọn thứ tự thực hiện đúng

A. B3 B2 B1

B. B2 B1 B3

C. B3 B1 B2

D. B1 B2 B3

Câu 181: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, ... for(i=0;i<n;i++) if(a[i]==x) { kt=1; break; } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. kt=0 B. kt=0; C. kt=1 D. kt=1;</p>
---	---

Câu 182: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; ... if(a[i]==x) { kt=1; break; } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	Điền lệnh vào dấu ... A. for(i=0;i>n;i++) B. for(i=0;i<n;i++) C. for(i=0;i<n;i--) D. for(i=0;i>n;i--)
---	---

Câu 183: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0;i<n;i++) ... { kt=1; break; } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	Điền lệnh vào dấu ... A. if(a[i]=x) B. if(a[i]==x) C. if(a[i]!=x) D. if(a[i]!=x);
---	--

Câu 184: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0;i<n;i++) if(a[i]==x) { ... break; } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	Điền lệnh vào dấu ... A. kt=1; B. kt=1 C. kt=0; D. kt=0
---	--

Câu 185: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0;i<n;i++) if(a[i]==x) { kt=1; ... } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	Điền lệnh vào dấu ... A. break B. break; C. continue; D. continue
--	--

Câu 186: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0; i<n; i++) if(a[i]==x) { kt=1; break; } ... printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. if(kt!=1) B. if(kt==1) C. if(kt!=0) D. if(kt==0)</p> <p>P/s: Theo thuật toán thì k=1 thì tìm thấy, ngược lại thì k không tìm thấy. Thì cũng tương đương với k≠0 là ngược với không tìm thấy => if(kt!=0) cũng đúng!</p>
---	--

Câu 187: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0; i<n; i++) if(a[i]==x) { kt=1; break; } if(kt==1) ... else printf("Khong tim thay x trong day"); }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. printf("Tim thay x trong day") B. printf("Tim thay x trong day"); C. printf("Tim thay x trong day" D. printf("Tim thay x trong day";</p>
---	---

Câu 188: Cài đặt thuật toán tìm kiếm tuần tự thành hàm searching

<pre>void searching(int a[], int n, int x) { int i, kt=0; for(i=0; i<n; i++) if(a[i]==x) { kt=1; break; } if(kt==1) printf("Tim thay x trong day"); else ... }</pre>	<p>Điền lệnh vào dấu ...</p> <p>A. printf("Khong tim thay x trong day") B. printf("Khong tim thay x trong day"); C. printf("Khong tim thay x trong day" D. printf("Khong tim thay x trong day";</p>
---	---

Câu 189: Phát biểu sau đúng hay sai: Ý tưởng phương pháp tìm kiếm Nhị Phân

- Cho dãy số a có n phần tử đã được sắp xếp tăng dần
- Chọn phần tử nằm giữa dãy số a, gọi là a[mid]
- So sánh giá trị của phần tử a[mid] với x
- Nếu a[mid]<x. Tìm x trong đoạn từ a[mid+1] đến a[n-1]
- Nếu a[mid]==x, kết thúc và thông báo "tìm thấy"
- Nếu a[mid]>x. Tìm x trong đoạn từ a[0] đến a[mid-1]
- Kết thúc khi đã duyệt hết các đoạn

A. Sai **B. Đúng**

Câu 190: Phát biểu sau đúng hay sai: Ý tưởng phương pháp tìm kiếm Nhị Phân

- Cho dãy số a có n phần tử đã được sắp xếp tăng dần
- Chọn phần tử nằm chính giữa dãy số a, gọi là a[mid]
- So sánh giá trị của phần tử a[mid] với x
- Nếu a[mid]<x. Tìm x trong đoạn từ a[mid+1] đến a[n-1]
- Nếu a[mid]==x, kết thúc và thông báo "tìm thấy"
- Nếu a[mid]>x. Tìm x trong đoạn từ a[0] đến a[mid-1]

- Kết thúc khi đã duyệt hết các đoạn

A. Sai **B. Đúng**

Câu 191: Thuật toán tìm kiếm nhị phân: Tìm phần tử dãy số a có giá trị bằng x nhập từ bàn phím?

B1: Gán $L=i$; $r=n-1$; $kt=0$;

B2: Trong khi ($L \leq r$); $mid=(L+r)/2$. So sánh $a[mid]$ với x. Nếu:

- $a[mid] < x$ thì $L=mid+1$;
- $a[mid] > x$ thì $r=mid-1$;
- $a[mid] == x$ thì $kt=1$, dừng kiểm tra.

B3: Kiểm tra giá trị của kt. Nếu $kt=1$ thì thông báo "tìm thấy". Ngược lại, thông báo "không tìm thấy".

Chọn thứ tự thực hiện đúng

A. B1 B2 B3 B. B1 B3 B2

C. B2 B1 B3 D. B3 B2 B1

Câu 192: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid, ... while(l<=r) { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else if(a[mid] > x) r=mid-1; else } } { Kt=1; break; } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. kt=0; B. kt=1; C. kt=2; D. kt=3;</p>
---	--

Câu 193: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; ... { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else if(a[mid] > x) r=mid-1; else { Kt=1; break; } } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. while(l <= r) B. while(l <= r); C. while(l >= r) D. while(l >= r);</p>
---	--

Câu 194: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { ... </pre>	<p>Điền lệnh vào dấu ...</p> <p>A. $mid=(l+r)/2$ B. $mid=(l-r)/2$; C. $mid=(l-r)/2$ D. $mid=(l+r)/2$;</p>
---	--

<pre> if(a[mid] < x) l=mid+1; else if(a[mid] > x) r=mid-1; else { Kt=1; break; } } if(k==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	
---	--

Câu 195: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; ... l=mid+1; else if(a[mid] > x) r=mid-1; else { Kt=1; break; } } if(k==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	Điền lệnh vào dấu ... A. if (a[mid] < x) B. if (a[mid] > x) C. if (a[mid] = x) D. if (a[mid] != x)
---	--

Câu 196: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; if(a[mid] < x) ... else if(a[mid] > x) r=mid-1; else { Kt=1; break; } } if(k==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	Điền lệnh vào dấu ... A. l=mid-1; B. l=mid++1; C. l=mid++2; D. l=mid+1;
--	--

Câu 197: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else ... r=mid-1; else { </pre>	Điền lệnh vào dấu ... A. if(a[mid] < x) B. if(a[mid] > x); C. if(a[mid] > x) D. if(a[mid] < x);
--	---

<pre> Kt=1; break; } } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	
--	--

Câu 198: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else if(a[mid] > x) ... else { kt=1; break; } } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	Điền lệnh vào dấu ... A. r=mid-1; B. r=mid+1; C. r=mid--1; D. r=mid++1;
---	--

Câu 199: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else if(a[mid] > x) r=mid-1; else { ... break; } } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	Điền lệnh vào dấu ... A. kt=1; B. kt=0; C. kt==0; D. kt==1;
--	--

Câu 200: Cài đặt thuật toán tìm kiếm nhị phân thành hàm

<pre> BinarySearching void BinarySearching(int a[], int n, int x) { int l,r,mid,kt=0; while(l <= r) { mid=(l+r)/2; if(a[mid] < x) l=mid+1; else if(a[mid] > x) r=mid-1; else { kt=1; ... } } if(kt==1) printf("Tim thay x trong day"); else printf("Khong tim thay x trong day"); } </pre>	Điền lệnh vào dấu ... A. break B. break; C. continue D. continue;
---	--

