

TT	Nội dung	Trả lời
1	Ngôn ngữ lập trình C được phát triển dựa trên các ngôn ngữ lập trình nào?	
A	Ngôn ngữ Assembler.	
B	Ngôn ngữ COBOL.	
C	Ngôn ngữ DEC PDP.	
D	Ngôn ngữ B và BCPL.	
E		
F		
2	Ngôn ngữ lập trình C là ngôn ngữ lập trình:	
A	Cấp cao, có cấu trúc.	
B	Cấp cao, không có cấu trúc.	
C	Cấp thấp, có cấu trúc.	
D	Cấp thấp, không có cấu trúc.	
E		
F		
3	Trong lập trình C, cách viết nào là đúng khi cần khai báo dòng ghi chú:	
A	/* dòng ghi chú*/ hoặc //dòng ghi chú	
B	/dòng ghi chú/	
C	“dòng ghi chú”	
D	dòng ghi chú	
E		
F		
4	Để dừng màn hình xem kết quả chương trình viết bằng ngôn ngữ C (trên phần mềm Visual Studio), ta sử dụng lệnh:	
A	printf()	
B	getch()	
C	scanf()	
D	exit()	
E		
F		
5	Trong ngôn ngữ lập trình C, từ khóa bắt buộc phải được khai báo dưới dạng:	
A	Ký tự chữ in.	
B	Ký tự chữ thường.	
C	Ký tự chữ in hoặc chữ thường.	
D	Các ký tự dạng số.	
E		
F		

6	Biến là một đại lượng:	
A	Thay đổi giá trị khi thực thi chương trình.	
B	Không thay giá trị đôi khi thực thi chương trình.	
C	Không thể chứa giá trị.	
D	Có giá trị tùy ý.	
E		
F		
7	Hằng là một đại lượng:	
A	Thay đổi giá trị khi thực thi chương trình.	
B	Không đổi giá trị khi thực thi chương trình.	
C	Không thể chứa giá trị.	
D	Không cần khai báo trong chương trình.	
E		
F		
8	Cho biểu thức sau: $tong = a + b - c$ ; Phát biểu nào sau đây là đúng:	
A	Các toán hạng là "=", "+", "-"	
B	Các toán tử là "tong, a, b, c"	
C	Biểu thức trên chỉ bao gồm các toán hạng.	
D	Các toán hạng là "tong, a, b, c", toán tử là "=", "+", "-"	
E		
F		
9	Trong ngôn ngữ lập trình C, tên biến nào được khai báo dưới đây là đúng:	
A	diem toan	
B	3diemtoan	
C	diemtoan	
D	-diemtoan	
E		
F		
10	Miền giá trị xử lý dữ liệu kiểu <b>short int</b> là:	
A	0...255	
B	-32768 ...32767	
C	-128...127	
D	0...65535	
E		
F		
11	Cho biểu thức: $chu\_vi = 2 * \pi * r$ ; Với $\pi = 3.14$ , r là bán kính đường tròn. Phát biểu nào dưới đây là đúng:	
A	$\pi$ là hằng số, r biến số.	

B	pi, r là các biến số.	
C	pi, r là các hằng số.	
D	pi là biến số, r là hằng số.	
E		
F		
12	Cho biểu thức: $y = ++x$ ; Với $x = 10$ , y có giá trị bằng bao nhiêu?	
A	9	
B	10	
C	11	
D	12	
E		
F		
13	Chương trình sau có kết quả bằng: <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i=10;     printf("%o",i); }</pre>	
A	12	
B	10	
C	8	
D	5	
E		
F		
14	Cho đoạn chương trình sau: <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i=5, j=6, tong;     tong=i+j;     printf("%d",tong); }</pre> Hãy cho biết biến tong có giá trị bằng bao nhiêu?	
A	6	
B	5	
C	2	
D	11	
E		
F		

15	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int a, tích;     printf("nhập vào số a:");     scanf("%d",&amp;a);     tích = a*a*a;     printf("%d\n",tích); }</pre> <p>Với a=3, hãy cho biết biến tích có giá trị bằng bao nhiêu?</p>	
A	9	
B	27	
C	36	
D	12	
E		
F		
16	Cách viết nào sau đây là đúng với cú pháp khai báo của câu lệnh if?	
A	if bieu_thuc	
B	if < bieu_thuc >	
C	if { bieu_thuc }	
D	if (bieu_thuc)	
E		
F		
17	Vòng lặp for() đầy đủ là một dạng vòng lặp:	
A	Biết trước được số lần lặp.	
B	Không xác định được số lần lặp.	
C	Có số lần lặp tùy ý.	
D	Có số lần lặp cố định.	
E		
F		
18	Vòng lặp while() đầy đủ là một dạng vòng lặp:	
A	Biết trước số lần lặp.	
B	Không biết trước số lần lặp.	
C	Lặp không cần kiểm tra biểu thức điều kiện.	
D	Có số lần lặp cố định.	
E		
F		
19	Lệnh nào trong các lệnh sau cho phép chuyển sang lần lặp tiếp theo mà không cần phải thực hiện phần còn lại của	

	vòng lặp?	
A	break	
B	goto	
C	continue	
D	return	
E		
F		
20	Lệnh nào trong các lệnh sau cho phép chuyển tới một nơi nào đó đã được gán nhãn?	
A	break	
B	goto	
C	continue	
D	exit	
E		
F		
21	Cấu trúc switch...case là một dạng cấu trúc:	
A	Lựa chọn 1 trong nhiều giá trị.	
B	Lựa chọn 1 giá trị.	
C	Không lựa chọn giá trị nào cả.	
D	Chức năng khác.	
E		
F		
22	Cho cấu trúc sau: if(bieu_thuc_dieu_kien) { khoi_lenh; } Cấu trúc if() là một dạng cấu trúc:	
A	Kiểm tra biểu thức điều kiện trước, nếu biểu thức điều kiện có giá trị đúng mới thực hiện khối lệnh.	
B	Thực hiện khối lệnh trước, sau đó mới kiểm tra biểu thức điều kiện.	
C	Thực hiện khối lệnh mà không cần kiểm tra biểu thức điều kiện	
D	Chức năng khác.	
E		
F		
23	Cho cấu trúc sau: for (bieu_thuc_1; bieu_thuc_2; bieu_thuc_3) { khoi_lenh; }	

	bieu_thuc_1 có ý nghĩa:	
A	Khởi tạo giá trị ban đầu cho biến điều khiển.	
B	Điều kiện để thực hiện tiếp tục vòng lặp.	
C	Thay đổi giá trị biến điều khiển cho lần lặp tiếp theo.	
D	Không có nghĩa.	
E		
F		
24	Cho cấu trúc sau: for (bieu_thuc_1; bieu_thuc_2; bieu_thuc_3) { khai_lenh; } bieu_thuc_2 có ý nghĩa:	
A	Khởi tạo giá trị ban đầu cho biến điều khiển.	
B	Kiểm tra biểu thức điều kiện đúng để thực hiện vòng lặp.	
C	Thay đổi giá trị biến điều khiển cho lần lặp tiếp theo.	
D	Không có nghĩa.	
E		
F		
25	Cho cấu trúc vòng lặp sau: while (bieu_thuc_dieu_kien) { khai_lenh; } Cấu trúc vòng lặp while là cấu trúc:	
A	Kiểm tra biểu thức điều kiện đúng, sau đó thực hiện khối lệnh.	
B	Thực hiện khối lệnh trước, sau đó mới kiểm tra biểu thức điều kiện.	
C	Không cần kiểm tra biểu thức điều kiện vẫn thực hiện khối lệnh.	
D	Chức năng khác.	
E		
F		
26	Cho đoạn chương trình sau: #include<stdio.h> #include<conio.h> int main() { int n; printf("nhap vao so n:"); scanf("%d",&n); switch (n%2)	

	<pre> {     case 0:         printf("so chan");         break;     case 1:         printf("so le");         break; } getch(); } </pre> <p>Đoạn chương trình trên thực hiện:</p>	
A	Nhập vào 1 số n, cho biết số n là số chẵn hay lẻ.	
B	Nhập vào 1 số n, cho biết số n là số lẻ.	
C	Nhập vào 1 số n, cho biết số n là số chẵn.	
D	Kết quả khác.	
E		
F		
27	<p>Cho đoạn chương trình sau:</p> <pre> #include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i=0, n,t=0;     while (i&lt;=n)     {         t=t+i;         i++;     }     printf("%d\n",t); } </pre> <p>Với n=3, biến t có giá trị bằng bao nhiêu?</p>	
A	0	
B	1	
C	3	
D	6	
E		
F		
28	<p>Cho đoạn chương trình sau:</p> <pre> #include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i, n, tong; </pre>	

	<pre>printf("nhap vao so n:"); scanf("%d",&amp;n); for (i=2; ; i++)     printf("%d",tong); }</pre> <p>Kết quả đoạn chương trình trên:</p>	
A	Lập vô tận vì thiếu biểu thức điều kiện lặp.	
B	Lỗi vì thiếu biểu thức khởi tạo giá trị ban đầu.	
C	Lỗi vì thiếu biểu thức thay đổi giá trị biến điều khiển.	
D	Lỗi vì thiếu khai báo biến.	
E		
F		
29	<p>Cho chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i, n, solan = 0;     printf("nhap vao so n:");     scanf("%d",&amp;n);     for (i=1; i&lt;n; i++)         solan = solan + 1;     printf("%d",solan); }</pre> <p>với n = 4, biến solan có kết quả bằng bao nhiêu?</p>	
A	1	
B	2	
C	3	
D	4	
E		
F		
30	<p>Cho chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     tong= 453+343;     printf("\Ket qua la %d ", tong) ; }</pre> <p>Đoạn chương trình trên báo lỗi do:</p>	
A	Thiếu khai báo dấu chấm phẩy (;)	
B	Thiếu khai báo dấu phẩy (,)	
C	Thiếu khai báo biến "tong"	



D	Thiếu hàm xuất giá trị.	
E		
F		
31	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i, n,t =1;     printf("nhap vao so n:");     scanf("%d",&amp;n);     for(i=1;i&lt;=n; i=i+2)         t = t*i;     printf("%d",t);     getch(); }</pre> <p>Với n =5, biến t có giá trị bằng bao nhiêu?</p>	
A	0	
B	5	
C	15	
D	16	
E		
F		
32	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i, x=6, y=5, tam;     if(x&gt;y)     {         tam = x;         x = y;         y = tam;     }     printf("Giá trị của x là %d",x); }</pre> <p>Biến x có giá trị bằng bao nhiêu?</p>	
A	5	
B	6	
C	1	
D	0	
E		

F		
33	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int i, n;     printf("nhap vao so n:");     scanf("%d",&amp;n);     for (i=1;i&lt;=n; i++)         if(n%i==0)             printf("%3d",i);     getch(); }</pre> <p>Với n = 4, biến i có giá trị bằng bao nhiêu?</p>	
A	1 2 4	
B	1 4	
C	2 4	
D	Kết quả khác	
E		
F		
34	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int a=40,b=4;     while(a!=b)         if (a&gt;b)             a=a-b;         else             b=b-a;     printf("%d",a); }</pre> <p>Biến a có giá trị cuối cùng bằng bao nhiêu?</p>	
A	2	
B	16	
C	4	
D	0	
E		
F		
35	<p>Cho đoạn chương trình sau:</p> <pre>#include &lt;stdio.h&gt;</pre>	

	<pre>#include &lt;conio.h&gt; int main() {     int sole=0, i, n;     printf("nhap vao so n:");     scanf("%d",&amp;n);     for(int i=1;i&lt;=n;i++)         if(i%2!=0)             sole = sole+1;     printf("%d",sole);     getch(); }</pre> <p>Đoạn chương trình trên cho thực hiện:</p>	
A	Nhập vào số n, đếm các số lẻ từ 1 đến n	
B	Nhập vào số n, in ra số các số chẵn từ 1 đến n	
C	Nhập vào số n, in ra tổng các số chẵn từ 1 đến n	
D	Nhập vào số n, in ra tích các số lẻ từ 1 đến n	
E		
F		
36	Chương trình con là:	
A	Một chương trình để thực hiện một công việc được lặp đi lặp lại nhiều lần.	
B	Một chương trình để thực hiện một công việc cố định.	
C	Một chương trình để thực hiện một khối công việc cố định.	
D	Một chương trình để thực hiện một công việc một lần.	
E		
F		
37	Hàm printf() dùng để:	
A	Xuất dữ liệu.	
B	Nhập dữ liệu.	
C	Nhập xuất dữ liệu.	
D	Xóa dữ liệu.	
E		
F		
38	Hàm scanf() dùng để:	
A	Nhập xuất dữ liệu.	
B	Nhập dữ liệu.	
C	Xuất dữ liệu.	
D	Xóa dữ liệu.	
E		
F		
39	Cách truyền nào trong hàm sẽ không làm thay đổi giá trị	

	của biến trong chương trình chính?	
A	Truyền bằng tham trị.	
B	Truyền bằng tham biến.	
C	Truyền bằng địa chỉ của biến.	
D	Truyền bằng tham biến hoặc tham trị	
E		
F		
40	Hàm đệ quy là hàm:	
A	Bên trong có lệnh gọi đến chính nó.	
B	Bên trong không có lệnh gọi đến chính nó.	
C	Xuất giá trị.	
D	Nhập giá trị.	
E		
F		
41	Hàm return() dùng để:	
A	Gọi và trả về kết quả từ chương trình.	
B	Trả về kết quả mà không cần gọi hàm từ chương trình.	
C	Gọi và trả về kết quả tùy ý.	
D	Không trả về kết quả khi gọi hàm.	
E		
F		
42	Đoạn chương trình sau cho thực hiện: <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int a=2,b=3;     printf("tích của 2 số a, b là: %d",a*b);     getch(); }</pre>	
A	Cho nhập vào 2 số a, b.	
B	Xuất ra kết quả là tích của 2 số a, b.	
C	Xuất ra kết quả tổng của 2 số a, b.	
D	Xuất ra kết quả của 2 số a, b.	
E		
F		
43	Đoạn chương trình sau cho thực hiện: <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; int main() {     int n;</pre>	

	scanf("%d",&n); }	
A	Nhập vào số n có kiểu số nguyên.	
B	Xuất ra số n kiểu số nguyên.	
C	Nhập vào n có kiểu tùy ý.	
D	Xuất ra số n có kiểu tùy ý.	
E		
F		
44	Hàm cho nhập một kí tự từ bàn phím ngay sau khi gõ, không chờ nhấn enter và không hiện ra màn hình là hàm:	
A	scanf();	
B	getchar();	
C	getch();	
D	getche();	
E		
F		
45	Cho đoạn chương trình sau: <pre>int ketqua(int n) {     int s=0;     for (int i=1;i&lt;=n;i++)         s=s+i;     return (s1); }</pre> Với n =3, s có giá trị bằng:	
A	Lỗi chương trình.	
B	1	
C	2	
D	3	
E		
F		
46	Cho đoạn chương trình sau: <pre>int ketqua(int n) {     int s=0;     for (int i=1;i&lt;=n;i++)         s=s+i;     return (); }</pre> Với n =3, s có giá trị bằng:	
A	0	
B	1	
C	2	

D	Lỗi chương trình.	
E		
F		
47	<p>Cho đoạn chương trình sau:</p> <pre>int ketqua(int n) {     int s=0;     for (int i=1;i&lt;=n;i++)         s=s+i;     return (s); }</pre> <p>Với n =3, s có giá trị bằng:</p>	
A	3	
B	4	
C	5	
D	6	
E		
F		
48	<p>Cho đoạn chương trình sau:</p> <pre>int ketqua(int n) {     int s=1;     for(int i=1;i&lt;=n;i++)         s=s*i;     return (s); }</pre> <p>Với n =2, s có giá trị bằng:</p>	
A	0	
B	1	
C	2	
D	6	
E		
F		
49	<p>Cho đoạn chương trình sau:</p> <pre>int ketqua(int n) {     int s=0;     for (int i=1;i&lt;=n;i++)         s=s*i;     return (s); }</pre> <p>Với n =3, s có giá trị bằng:</p>	
A	0	

B	1	
C	2	
D	3	
E		
F		
50	<p>Cho đoạn chương trình sau:</p> <pre>int ketqua(int n) {     int s=1;     for (int i=1;i&lt;=n;i++)         s=s+i;     return (s); }</pre> <p>Với <math>n = 3</math>, <math>s</math> có giá trị bằng:</p>	
A	0	
B	3	
C	5	
D	7	
E		
F		
51	Mảng là:	
A	Một tập hợp các phần tử có kiểu dữ liệu riêng và chung tên gọi.	
B	Một tập hợp các phần tử có kiểu dữ liệu riêng và có tên gọi riêng cho mỗi phần tử.	
C	Một tập hợp các phần tử không cùng kiểu dữ liệu và tên gọi	
D	Một tập hợp các phần tử có cùng kiểu dữ liệu và chung tên gọi.	
E		
F		
52	Khai báo <code>int m[10]</code> ; cho biết?	
A	Mảng có tên $m$ và có 10 phần tử.	
B	Mảng có tên $m$ và có 10 phần tử kiểu số nguyên.	
C	Mảng có tên $m$ và có số phần tử tùy ý kiểu số nguyên.	
D	Mảng có tên $m$ và có 9 phần tử.	
E		
F		
53	Cho khai báo mảng: <code>int m[3];</code>	

	Giá trị phần tử thứ 2 của mảng là:	
A	m[0]	
B	<b>m[1]</b>	
C	m[2]	
D	m[3]	
E		
F		
54	Cho khai báo mảng: int m[]={4,2,6,1,7}; Phần tử m[3] có giá trị bằng:	
A	<b>1</b>	
B	6	
C	7	
D	4	
E		
F		
55	Cho khai báo mảng: int x[]={4,2,6}; Phát biểu nào dưới đây là đúng:	
A	x[1]=4, x[2]=2, x[3]=6	
B	<b>x[0]=4, x[1]=2, x[2]=6</b>	
C	x[0]=2, x[1]=4, x[2]=6	
D	x[0]=6, x[1]=2, x[2]=4	
E		
F		
56	Cho đoạn chương trình sau: void mang1chieu(int a[], int n) { for(int i=0;i<n;i++) { printf("nhap vao phan tu thu%d:",i+1); scanf("%d",&a[i]); } } Với n = 5, chương trình trên cho thực hiện:	
A	Nhập vào 1 mảng 5 phần tử.	
B	<b>Nhập vào 1 mảng 5 phần tử kiểu số nguyên.</b>	
C	Nhập vào một mảng với số phần tử tùy ý.	
D	Nhập vào 1 mảng 10 phần tử.	
E		
F		
57	Cho đoạn chương trình sau:	



	<pre>void mang1chieu(int a[], int n) {     int t=0;     for(int i=0;i&lt;n;i++)         t=t+a[i];     printf("%d\n",t); }</pre> <p>Với n = 5, chương trình trên thực hiện:</p>	
A	Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.	
B	Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.	
C	Tính tổng giá trị 4 phần tử của mảng kiểu số nguyên.	
D	Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.	
E		
F		
58	<p>Cho đoạn chương trình sau:</p> <pre>void mang1chieu(int a[], int n) {     int t=1;     for(int i=0;i&lt;n;i++)         t=t*a[i];     printf("%d\n",t); }</pre> <p>Với n =6, chương trình trên thực hiện:</p>	
A	Tính tích giá trị 6 phần tử của mảng kiểu số nguyên.	
B	Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.	
C	Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.	
D	Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.	
E		
F		
59	<p>Cho đoạn chương trình sau:</p> <pre>void mang(int a[], int n) {     int m=a[0];     for(int i=1;i&lt;n;i++)         if (m &gt;=a[i])             m=a[i];     printf("%d\n",m); }</pre> <p>Với n = 3, chương trình trên thực hiện:</p>	
A	Tìm giá trị phần tử bé nhất trong 3 phần tử mảng.	
B	Tìm giá trị phần tử lớn nhất trong 3 phần tử mảng.	
C	Tính tổng 3 phần tử của mảng.	
D	Tính tích 3 phần tử của mảng.	

E		
F		
60	<p>Cho đoạn chương trình sau:</p> <pre>void mang1chieu(int a[], int n) {     int tich=1,tong=0;     for(int i=0;i&lt;n;i++)         if (a[i]%2==0)             tich=tich*a[i];         else             tong=tong +a[i];     printf("tich cac phan tu chan:%d\n",tich);     printf("tong cac phan tu le:%d\n",tong); }</pre> <p>Với n = 4, chương trình trên cho thực hiện:</p>	
A	Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số lẻ, tích các phần tử số chẵn.	
B	Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số chẵn, tích các phần tử số lẻ.	
C	Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng 4 phần tử trên.	
D	Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tích 4 phần tử trên.	
E		
F		

**Câu 150**

Cho chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    tong= 453+343;
```

```
    printf("\\Ket qua la %d ", tong) ;
```

```
}
```

Đoạn chương trình trên báo lỗi do:

- A. Thiếu khai báo dấu chấm phẩy (;)
- B. Thiếu khai báo dấu phẩy (,)
- C. Thiếu khai báo biến tong. **(Đúng)**
- D. Thiếu hàm xuất giá trị.

**Câu 151**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int i, n, t = 1;
    printf("nhap vao so n:");
    scanf("%d",&n);
```

```
for(i=1;i<=n; i=i+2)

    t = t*i;

    printf("%d",t);

    getch();

}
```

với n =5, biến t có giá trị bằng bao nhiêu?

A. 0

B. 5

C. 15

(Đúng)

D. 16

**Câu 152**

Cho đoạn chương trình sau:

```
#include <stdio.h>

#include <conio.h>
```

```
int main()
{
    int i, x=6, y=5, tam;
    printf("nhap vao so x:");
    scanf("%d",&x);
    printf("nhap vao so y:");
    scanf("%d",&y);

    if(x>y)
    {
        tam = x;
        x = y;
        y = tam;
    }

    printf("giá trị của x là %d",x);
}
```

Biến x có giá trị bằng bao nhiêu?

A. 5

(Đúng)

B. 6

C. 1

D. 0

**Câu 153**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, n;
    printf("nhap vao so n:");
    scanf("%d",&n);
    for (i=1;i<=n; i++)
        if(n%i==0)
            printf("%3d",i);
    getch();
}
```

Với  $n = 4$ , biến  $i$  có giá trị bằng bao nhiêu?

A. 1, 2, 4

(Đúng)

B. 1, 4

C. 2, 4

D. Kết quả khác

**Câu 154**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a=40, b=4;
    printf("nhap vao so a:");
    scanf("%d",&a);
    printf("nhap vao so b:");
    scanf("%d",&b);
    while(a!=b)
        if (a>b)
```

```
    a=a-b;  
    else b=b-a;  
    printf("%d",a);  
}
```

Biến a có giá trị bằng bao nhiêu?

A. 2

B. 16

C. 4

(Đúng)

D. 0

**Câu 155**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```



```
{  
int sole=0, i, n;  
    printf("nhap vap so n:");  
    scanf("%d",&n);  
    for(i=1;i<n;i++)  
        if(i % 2!=0)  
            sole = sole+1;  
    printf("%d",sole);  
    getch();  
}
```

Đoạn chương trình trên cho thực hiện:

- A.     Nhập vào số n, in ra số các số lẻ từ 1 đến n (Đúng)
  
- B.     Nhập vào số n, in ra số các số chẵn từ 1 đến n
  
- C.     Nhập vào số n, in ra tổng các số chẵn từ 1 đến n
  
- D.     Nhập vào số n, in ra tích các số lẻ từ 1 đến n

**Câu 156**

Hàm là:

- A. Một chương trình con để thực hiện nhiều công việc được lặp đi lặp lại nhiều lần. **(Đúng)**
- B. Một chương trình con để thực hiện một công việc cố định.
- C. Một chương trình con để thực hiện một khối công việc cố định.
- D. Một chương trình con để thực hiện một công việc lặp đi lặp lại nhiều lần.

**Câu 157**

Hàm printf() là hàm:

- A. Xuất dữ liệu. **(Đúng)**
- B. Nhập dữ liệu.

C. Nhập xuất dữ liệu.

D. Xóa dữ liệu.

**Câu 158**

Hàm scanf() là hàm:

A. Nhập xuất dữ liệu.

B. Nhập dữ liệu.

(Đúng)

C. Xuất dữ liệu.

D. Xóa dữ liệu.

**Câu 159**

Cách truyền nào trong hàm sẽ không làm thay đổi giá trị của biến trong chương trình chính?

- A. Truyền bằng tham trị. (Đúng)
- B. Truyền bằng tham biến.
- C. Truyền bằng địa chỉ của biến.
- D. Truyền bằng tham biến hoặc tham trị

**Câu 160**

Hàm đệ quy là hàm:

- A. Bên trong có lệnh gọi đến chính nó. (Đúng)
- B. Bên trong không có lệnh gọi đến chính nó.

- C. Xuất giá trị.
- D. Nhập giá trị.

**Câu 161**

Hàm return() là hàm:

- A. Gọi và trả về kết quả từ chương trình. (Đúng)
- B. Trả về kết quả mà không cần gọi hàm từ chương trình.
- C. Gọi và trả về kết quả tùy ý.
- D. Không trả về kết quả khi gọi hàm.

**Câu 162**

Đoạn chương trình sau cho thực hiện:

```
#include <stdio.h>

#include <conio.h>

int main()
{
    int a=2, b=3;

    printf("tich cua 2 so a, b la: %d",a*b);

    getch();

}
```

- A. Cho nhập vào 2 số a, b
- B. Xuất ra kết quả là tích của 2 số a, b
- C. Xuất ra kết quả tổng của 2 số a, b
- D. Xuất ra kết quả của 2 số a, b

(Đúng)

**Câu 163**

Đoạn chương trình sau cho thực hiện:

```
#include <stdio.h>

#include <conio.h>

int main()
{
    int n;

    scanf("%d",&n);

}
```

- A. Nhập vào số n có kiểu số nguyên (Đúng)
- B. Xuất ra số n kiểu số nguyên
- C. Nhập vào n có kiểu tùy ý
- D. Xuất ra số n có kiểu tùy ý

**Câu 164**

Hàm cho nhập một kí tự từ bàn phím ngay sau khi gõ, không chờ nhấn enter và không hiện ra màn hình là hàm:

A.     scanf();

B.     getchar();

C.     getch();

(Đúng)

D.     getche();

**Câu 165**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int n;
```

```
    long tong(int);
```

```
        printf("nhập vào số n:");
```

```
        scanf("%d",&n);
```

```
        printf("%d tong =%1d. \n", n, tong(n));
```



```
    getch();  
}  
  
long tong (int n)  
{  
    int i;  
    long s=0;  
    {  
        for (i=1;i<=n;i++)  
            s=s+i;  
    }  
    return (s1);  
}  
}
```

Với  $n = 3$ , giá trị của  $s$  là?

A. Lỗi chương trình

(Đúng)

B. 1

C. 2

**Câu 166**

Cho đoạn chương trình sau:

```
#include <stdio.h>

#include <conio.h>

int main()
{
    int n;

    long tong(int);

    printf("nhập vào số n:");

    scanf("%d",&n);

    printf("%d tong =%1d. \n", n, tong(n));

    getch();
}

long tong (int n)
{
    int i;

    long s=0;

    {
        for (i=1;i<=n;i++)

            s=s+i;
```

```
return ();  
}  
}
```

Với  $n = 3$ , giá trị của  $s$  là:

A. 0

B. 1

C. 2

D. Lỗi chương trình

(Đúng)

**Câu 167**

Cho đoạn chương trình sau:

```
#include <stdio.h>  
#include <conio.h>  
int main()  
{
```

```
int n;

long tong(int);

    printf("nhập vào số n:");

    scanf("%d",&n);

    printf("%d tong =%1d. \n", n, tong(n));

    getch();
}

long tong (int n)
{
    int i;
    long s=0;
    {
        for (i=1;i<=n;i++)

            s=s+i;
    }
    return (s);
}
}
```

Với n =3, giá trị của s là:

A. 3

B. 4

C. 5

D. 6

(Đúng)

**Câu 168**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n;
    long tong(int);
    printf("nhập vào số n:");
    scanf("%d",&n);
    printf("%d tong =%1d. \n", n, tong(n));
    getch();
}
long tong (int n)
{
    int i;
```

```
long s=1;
{
    for (i=1;i<=n;i++)
        s=s*i;
    return (s);
}
```

Với  $n = 2$ , giá trị của  $s$  là:

A. 0

B. 1

C. 2

(Đúng)

D. 6

**Câu 169**

Cho đoạn chương trình sau:

```
#include <stdio.h>

#include <conio.h>

int main()
{
    int n;

    long tong(int);

    printf("nhập vào số n:");

    scanf("%d",&n);

    printf("%d tong =%1d. \n", n, tong(n));

    getch();
}

long tong (int n)
{
    int i;

    long s=0;

    {
        for (i=1;i<=n;i++)

            s=s*i;

        return (s);
    }
}
```

Với  $n = 3$ , giá trị của  $s$  là:

A. 0

(Đúng)

B. 1

C. 2

D. 3

**Câu 170**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n;
    long tong(int);
    printf("nhập vào số n:");
    scanf("%d",&n);
    printf("%d tong =%1d. \n", n, tong(n));
    getch();
}
```



```
}  
  
long tong (int n)  
{  
    int i;  
    long s=1;  
    {  
        for (i=1;i<=n;i++)  
            s=s+i;  
    }  
    return (s);  
}  
}
```

Với  $n = 3$ , giá trị của  $s$  là:

A. 0

B. 3

C. 5

D. 7

(Đúng)

**Câu 171**

Mảng là:

- A. Một tập hợp các phần tử có kiểu dữ liệu riêng và chung tên gọi.
- B. Một tập hợp các phần tử có kiểu dữ liệu riêng và có tên gọi riêng cho mỗi phần tử.
- C. Một tập hợp các phần tử không cùng kiểu dữ liệu và tên gọi
- D. Một tập hợp các phần tử có cùng kiểu dữ liệu và chung tên gọi. (Đúng)

**Câu 172**

Khai báo sau cho biết?

```
int m[10];
```

- A. Mảng có tên m và có 10 phần tử

B. Mảng có tên m và có 10 phần tử kiểu số nguyên

(Đúng)

C. Mảng có tên m và có số phần tử tùy ý kiểu số nguyên

D. Mảng có tên m và có 9 phần tử

**Câu 173**

Cho khai báo mảng:

```
int m[3];
```

Hãy cho biết vị trí phần tử thứ 2 của mảng là:

A. m[0]

B. m[1]

(Đúng)

C. m[2]

D.      $m[3]$

**Câu 174**

Cho khai báo mảng:

`int m[5]={4,2,6,1,7};`

Hãy cho biết giá trị của phần tử  $m[3] = ?$

A.     1

(Đúng)

B.     6

C.     7

D.     4

**Câu 175**

Cho khai báo mảng:

`int x[3]={4,2,6};`

phát biểu nào dưới đây là đúng?

A.  $x[1]=4, x[2]=2, x[3]=6$

B.  $x[0]=4, x[1]=2, x[2]=6$

(Đúng)

C.  $x[0]=2, x[1]=4, x[2]=6$

D.  $x[0]=6, x[1]=2, x[2]=4$

**Câu 176**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int a[10], i, n;
```

```
    printf("nhap vao gia tri n:");
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)  
    {  
        printf("nhap vao phan tu thu%d:",i+1);  
        scanf("%d",& a[i]);  
    }  
    getch();  
}
```

với n = 5, chương trình trên cho thực hiện:

- A. Nhập vào 1 mảng 5 phần tử.
- B. Nhập vào 1 mảng 5 phần tử kiểu số nguyên. (Đúng)
- C. Nhập vào một mảng với số phần tử tùy ý.
- D. Nhập vào 1 mảng 10 phần tử.

**Câu 177**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[10], i, n, tong=0;
    printf("nhap vao gia tri n:");
    scanf("%d",&n);
        for(i=0;i<n;i++)
        {
            printf("nhap vao phan tu thu%d:",i+1);
            scanf("%d",&a[i]);
        }
        for(i=0;i<n;i++)
            tong=tong + a[i];
        printf("%d\n",tong);
    getch();
}
```

Với n = 5, chương trình trên thực hiện:

A. Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.

(Đúng)

B. Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.

C. Tính tổng giá trị 4 phần tử của mảng kiểu số nguyên.

D. Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.

**Câu 178**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int m[10], i, n, tich=1;
    printf("nhap vao gia tri n:");
    scanf("%d",&n);
        for(i=0;i<n;i++)
        {
            printf("nhap vao phan tu thu%d:",i+1);
            scanf("%d",&m[i]);
        }
        for(i=0;i<n;i++)
            tich=tich * m[i];
```



```
printf("%d\n",tich);  
getch();  
}
```

Với  $n = 6$ , chương trình trên thực hiện:

- A. Tính tích giá trị 6 phần tử của mảng kiểu số nguyên. **(Đúng)**
- B. Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.
- C. Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.
- D. Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.

**Câu 179**

Cho đoạn chương trình sau:

```
#include <stdio.h>  
#include <conio.h>  
int main()  
{
```

```
int a[10], i, n, min =a[0];  
printf("nhap vao gia tri n:");  
scanf("%d",&n);  
    for(i=0;i<n;i++)  
    {  
printf("nhap vao phan tu thu%d:",i+1);  
scanf("%d",&a[i]);  
    }  
    for(i=0;i<n;i++)  
    if (min >a[i])  
        min=a[i];  
    printf("%d\n",min);  
getch();  
}
```

với n = 3, chương trình trên thực hiện:

A. Tìm giá trị phần tử bé nhất trong 3 phần tử mảng.

**(Đúng)**

B. Tìm giá trị phần tử lớn nhất trong 3 phần tử mảng.

C. Tính tổng 3 phần tử của mảng.

D. Tính tích 3 phần tử của mảng.

**Câu 180**

Cho đoạn chương trình sau:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a[6], i, n, tich, tong;
    tich = 1;
    tong = 0;
    printf("nhap vao gia tri n:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("nhap vao phan tu thu %d:",i+1);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
        if (a[i]%2==0)
```

```
tich=tich*a[i];  
  
else  
  
    tong = tong +a[i];  
  
    printf("tich cac phan tu chan:%d\n",tich);  
  
    printf("tong cac phan tu le:%d\n",tong);  
  
getch();  
}
```

với n = 4, chương trình trên cho thực hiện:

- A. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số lẻ, tích các phần tử số chẵn. **(Đúng)**
- B. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số chẵn, tích các phần tử số lẻ.
- C. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng 4 phần tử trên.
- D. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tích 4 phần tử trên.

Ngôn ngữ lập trình C được phát triển dựa trên các ngôn ngữ lập trình nào?

A. Ngôn ngữ Assembler.

B. Ngôn ngữ COBOL.

C. Ngôn ngữ DEC PDP.

D. Ngôn ngữ B và BCPL.

(Đúng)

**Câu 182**

Ngôn ngữ lập trình C là ngôn ngữ lập trình:

A. Cấp cao, có cấu trúc.

(Đúng)

B. Cấp cao, không có cấu trúc.

- C. Cấp thấp, có cấu trúc.
- D. Cấp thấp, không có cấu trúc.

**Câu 183**

Trong lập trình C, cách viết nào là đúng khi cần khai báo dòng ghi chú:

- A. `/* dòng ghi chú*/` hoặc `//dòng ghi chú` (Đúng)
- B. `/dòng ghi chú/`
- C. “dòng ghi chú”
- D. dòng ghi chú

**Câu 184**

Để dừng màn hình xem kết quả chương trình viết bằng ngôn ngữ C (trên phần mềm Visual Studio), ta sử dụng lệnh:

A. `printf()`

B. `getch()`

(Đúng)

C. `scanf()`

D. `exit()`

**Câu 185**

Trong ngôn ngữ lập trình C, từ khóa bắt buộc phải được khai báo dưới dạng:

A. Ký tự chữ in.

B. Ký tự chữ thường.

(Đúng)

C. Ký tự chữ in hoặc chữ thường.

D. Các ký tự dạng số.

**Câu 186**

Biến là một đại lượng:

A. Thay đổi giá trị khi thực thi chương trình.

(Đúng)

B. Không thay giá trị đổi khi thực thi chương trình.

C. Không thể chứa giá trị.

D. Có giá trị tùy ý.

**Câu 187**

Hằng là một đại lượng:



A. Thay đổi giá trị khi thực thi chương trình.

B. Không đổi giá trị khi thực thi chương trình.

(Đúng)

C. Không thể chứa giá trị.

D. Không cần khai báo trong chương trình.

**Câu 188**

Cho biểu thức sau:

tong=a+b-c;

Phát biểu nào sau đây là đúng:

A. Các toán hạng là "=", "+", "-"

B. Các toán tử là "tong, a, b, c"

C. Biểu thức trên chỉ bao gồm các toán hạng.

D. Các toán hạng là “tong, a, b, c”, toán tử là “=”, “+”, “-”

(Đúng)

**Câu 189**

Trong ngôn ngữ lập trình C, tên biến nào được khai báo dưới đây là đúng:

A. diem toan

B. 3diemtoan

C. diemtoan

(Đúng)

D. -diemtoan

**Câu 190**

Miền giá trị xử lý dữ liệu kiểu **short int** là:

A. 0...255

B. -32768 ...32767

(Đúng)

C. -128...127

D. 0...65535

**Câu 191**

Cho biểu thức:  $chu\_vi = 2 * \pi * r$ ;

Với  $\pi = 3.14$ ,  $r$  là bán kính đường tròn. Phát biểu nào dưới đây là đúng:

A.  $\pi$  là hằng số,  $chu\_vi$ ,  $r$  là các biến số.

(Đúng)

B.  $chu\_vi$ ,  $\pi$ ,  $r$  là các biến số.

C.  $chu\_vi$ ,  $\pi$ ,  $r$  là các hằng số.

D. chu\_vì, pi là các biến số, r là hằng số.

**Câu 192**

Cho biểu thức:  $y = ++x$ ;

Với  $x = 10$ ,  $y$  có giá trị bằng bao nhiêu?

A. 9

B. 10

C. 11

(Đúng)

D. 12

**Câu 193**

Chương trình sau có kết quả bằng:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int i=10;
```

```
printf("%o",i);
```

```
}
```

A. 12

(Đúng)

B. 10

C. 8

D. 5

**Câu 194**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int i=5, j=6, tong;
```

```
tong=i+j;
```

```
printf("%d",tong);
```

```
}
```

Hãy cho biết biến tong có giá trị bằng bao nhiêu?

A. 6

B. 5

C. 2

D. 11

(Đúng)

**Câu 195**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int a, tich;
```

```
printf("nhập vào số a:");
```

```
    scanf("%d",&a);
```

```
    tich = a*a*a;
```

```
    printf("%d\n",tich);
```

```
}
```

Với a=3, hãy cho biết biến tich có giá trị bằng bao nhiêu?

A. 9

B. 27

(Đúng)

C. 36

D. 12

Cách viết nào sau đây là đúng với cú pháp khai báo của câu lệnh if?

- A. `if bieu_thuc {khai_lenh};`
- B. `if {khai_lenh};`
- C. `if {bieu_thuc}{khai_lenh};`
- D. `if (bieu_thuc) {khai_lenh};`

(Đúng)

**Câu 197**

Vòng lặp for() đầy đủ là một dạng vòng lặp:

- A. Biết trước được số lần lặp.
- B. Không xác định được số lần lặp.
- C. Có số lần lặp tùy ý.
- D. Có số lần lặp cố định.

(Đúng)

**Câu 198**

Vòng lặp while() đầy đủ là một dạng vòng lặp:



A. Biết trước số lần lặp.

B. Không biết trước số lần lặp.

(Đúng)

C. Lặp không cần kiểm tra biểu thức điều kiện.

D. Có số lần lặp cố định.

**Câu 199**

Lệnh nào trong các lệnh sau cho phép chuyển sang lần lặp tiếp theo mà không cần phải thực hiện phần còn lại của vòng lặp?

A. break

B. goto

C. continue

(Đúng)

D. return

**Câu 200**

Lệnh nào trong các lệnh sau cho phép chuyển tới một nơi nào đó đã được gán nhãn?

A. break

B. goto (Đúng)

C. continue

D. exit

**Câu 201**

Cấu trúc switch...case là một dạng cấu trúc:

A. Lựa chọn 1 trong nhiều giá trị.

(Đúng)

B. Lựa chọn 1 giá trị.

C. Không lựa chọn giá trị nào cả.

D. Chức năng khác.

**Câu 202**

Cho cấu trúc sau:

```
if(bieu_thuc_dieu_kien)
{
    khai_lenh;
}
```

Cấu trúc if() là một dạng cấu trúc:

A. Kiểm tra bieu\_thuc\_dieu\_kien trước, nếu bieu\_thuc\_dieu\_kien có giá trị đúng mới thực hiện khai\_lenh. (Đúng)

- B. Thực hiện `khoi_lenh` trước, sau đó mới kiểm tra `bieu_thuc_dieu_kien`.
- C. Thực hiện `khoi_lenh` mà không cần kiểm tra `bieu_thuc_dieu_kien`.
- D. Kiểm tra `bieu_thuc_dieu_kien` trước, sau đó mới thực hiện `khoi_lenh`.

**Câu 203**

Cho cấu trúc sau:

```
for (bieu_thuc_1; bieu_thuc_2; bieu_thuc_3)
{
    khoi_lenh;
}
```

`bieu_thuc_1` có ý nghĩa:

- A. Khởi tạo giá trị ban đầu cho biến điều khiển. (Đúng)
- B. Điều kiện để thực hiện tiếp tục vòng lặp.

C. Thay đổi giá trị biến điều khiển cho lần lặp tiếp theo.

D. Không có nghĩa.

**Câu 204**

Cho cấu trúc sau:

```
for (bieu_thuc_1; bieu_thuc_2; bieu_thuc_3)
{
    khai_lenh;
}
```

bieu\_thuc\_2 có ý nghĩa:

A. Khởi tạo giá trị ban đầu cho biến điều khiển.

B. Kiểm tra biểu thức điều kiện đúng để thực hiện vòng lặp.

(Đúng)

C. Thay đổi giá trị biến điều khiển cho lần lặp tiếp theo.

- D. Không có nghĩa.

**Câu 205**

Cho cấu trúc vòng lặp sau:

```
while (bieu_thuc_dieu_kien)
```

```
{
```

```
    khai_lenh;
```

```
}
```

Cấu trúc vòng lặp while là cấu trúc:

- A. Kiểm tra bieu\_thuc\_dieu\_kien đúng, sau đó mới thực hiện (Đúng)  
khai\_lenh.
- B. Thực hiện khai\_lenh trước, sau đó mới kiểm tra  
bieu\_thuc\_dieu\_kien.
- C. Không cần kiểm tra bieu\_thuc\_dieu\_kien vẫn thực hiện khai\_lenh.
- D. Kiểm tra bieu\_thuc\_dieu\_kien trước, sau đó thực hiện khai\_lenh

**Câu 206**

Cho đoạn chương trình sau:

```
#include<stdio.h>

#include<conio.h>

int main()
{
    int n;
    printf("nhap vao so n:");
    scanf("%d",&n);
    switch (n%2)
    {
        case 0:
            printf("so chan");
            break;
        case 1:
            printf("so le");
            break;
    }
    getch();
}
```

Đoạn chương trình trên thực hiện:

A. Nhập vào 1 số n, cho biết số n là số chẵn hay lẻ.

(Đúng)

B. Nhập vào 1 số n, cho biết số n là số lẻ.

C. Nhập vào 1 số n, cho biết số n là số chẵn.

D. Kết quả khác.

**Câu 207**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i=0, n =3,t=0;
    while (i<=n)
    {
        t=t+i;
        i++;
    }
```



```
printf("%d\n",t);
```

```
}
```

Chương trình xuất ra màn hình giá trị bao nhiêu?

A. 0

B. 1

C. 3

D. 6

(Đúng)

**Câu 208**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int i, n, tong;
```

```
printf("nhap vao so n:");  
  
scanf("%d",&n);  
  
for (i=2; ; i++)  
  
    printf("%d",tong);  
  
}
```

Kết quả đoạn chương trình trên:

- A. Lặp vô tận vì thiếu biểu thức điều kiện lặp. (Đúng)
- B. Lỗi vì thiếu biểu thức khởi tạo giá trị ban đầu.
- C. Lỗi vì thiếu biểu thức thay đổi giá trị biến điều khiển.
- D. Lỗi vì thiếu khai báo biến.

**Câu 209**

Cho chương trình sau:

```
#include <stdio.h>  
  
#include <conio.h>
```

```
int main()
```

```
{
```

```
int i, n, solan = 0;
```

```
    printf("nhap vao so n:");
```

```
    scanf("%d",&n);
```

```
    for (i=1; i<n; i++)
```

```
        solan = solan +1;
```

```
    printf("%d",solan);
```

```
}
```

với  $n = 4$ , biến solan có kết quả bằng bao nhiêu khi chương trình thực hiện đến dòng lệnh `printf("%d",solan)`?

A. 1

B. 2

C. 3

(Đúng)

D. 4

**Câu 210**

Cho chương trình sau:

```
#include <stdio.h>

#include <conio.h>

int main()
{
    tong= 453+343;
    printf("\Ket qua la %d ", tong) ;
}
```

Đoạn chương trình trên báo lỗi do:

A. Thiếu khai báo dấu chấm phẩy (;)

B. Thiếu khai báo dấu phẩy (,)

C. Thiếu khai báo biến “tong”

(Đúng)

D. Thiếu hàm xuất giá trị.

**Câu 211**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, n,t=1;
    printf("nhap vao so n:");
    scanf("%d",&n);
    for(i=1;i<=n; i=i+2)
        t = t*i;
    printf("%d",t);
    getch();
}
```

Với  $n = 5$ , biến  $t$  có giá trị bằng bao nhiêu khi chương trình thực hiện đến dòng lệnh `printf("%d",t)`?

A. 0

B. 5

C. 15

(Đúng)

D. 16

**Câu 212**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, x=6, y=5, tam;
    if(x>y)
    {
        tam = x;
        x = y;
        y = tam;
    }
    printf("Giá trị của x là %d",x);
}
```

Biến x có giá trị bằng bao nhiêu khi chương trình thực hiện đến dòng lệnh printf("Giá trị của x là %d",x)?

A. 5

(Đúng)

B. 6

C. 1

D. 0

**Câu 213**

Cho đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, n;
    printf("nhap vao so n:");
    scanf("%d",&n);
    for (i=1;i<=n; i++)
        if(n%i==0)
            printf("%3d",i);
    getch();
}
```

```
}
```

Với  $n = 4$ , chương trình in ra màn hình các giá trị:

A.    1 2 4

(Đúng)

B.    1 4

C.    2 4

D.    Kết quả khác

**Câu 214**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int a=40,b=4;
```

```
    while(a!=b)
```



```
    if (a>b)
        a=a-b;
    else
b=b-a;
    printf("%d",a);
}
```

Biến a có giá trị cuối cùng bằng bao nhiêu?

A. 2

B. 16

C. 4

(Đúng)

D. 0

**Câu 215**

Cho đoạn chương trình sau:

```
#include <stdio.h>
```

```
#include <conio.h>

int main()
{
    int sole=0, i, n;

    printf("nhap vao so n:");

    scanf("%d",&n);

    for(int i=1;i<=n;i++)

        if(i%2!=0)

            sole = sole+1;

    printf("%d",sole);

    getch();
}
```

Đoạn chương trình trên cho thực hiện:

A. Nhập vào số n, đếm có bao nhiêu số lẻ từ 1 đến n.

(Đúng)

B. Nhập vào số n, in ra các số lẻ từ 1 đến n.

C. Nhập vào số n, in ra tổng các số lẻ từ 1 đến n.

- D. Nhập vào số  $n$ , đếm có bao nhiêu số chẵn từ 1 đến  $n$ .

**Câu 216**

Chương trình con là:

- A. Một chương trình để thực hiện một công việc được lặp đi lặp lại nhiều lần. **(Đúng)**
- B. Một chương trình để thực hiện một công việc cố định.
- C. Một chương trình để thực hiện một khối công việc cố định.
- D. Một chương trình để thực hiện một công việc một lần.

**Câu 217**

Hàm `printf()` dùng để:

- A. Xuất dữ liệu. **(Đúng)**

B. Nhập dữ liệu.

C. Nhập xuất dữ liệu.

D. Xóa dữ liệu.

**Câu 218**

Hàm scanf() dùng để:

A. Nhập xuất dữ liệu.

B. Nhập dữ liệu.

(Đúng)

C. Xuất dữ liệu.

D. Xóa dữ liệu.

**Câu 219**

Cách truyền nào trong hàm sẽ không làm thay đổi giá trị của biến trong chương trình chính?

- A. Truyền bằng tham trị. (Đúng)
- B. Truyền bằng tham biến.
- C. Truyền bằng địa chỉ của biến.
- D. Truyền bằng tham biến hoặc tham trị

**Câu 220**

Hàm đệ quy là hàm:

- A. Bên trong có lệnh gọi đến chính nó. (Đúng)

- B. Bên trong không có lệnh gọi đến chính nó.
- C. Xuất giá trị.
- D. Nhập giá trị.

**Câu 221**

Hàm return() dùng để:

- A. Gọi và trả về kết quả từ chương trình. **(Đúng)**
- B. Trả về kết quả mà không cần gọi hàm từ chương trình.
- C. Gọi và trả về kết quả tùy ý.
- D. Không trả về kết quả khi gọi hàm.

**Câu 222**

Đoạn chương trình sau cho thực hiện:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a=2,b=3;
        printf("tich cua 2 so a, b la: %d",a*b);
        getch();
}
```

- A. Cho nhập vào 2 số a, b.
- B. Xuất ra kết quả là tích của 2 số a, b.
- C. Khi biên dịch chương trình xảy ra lỗi.
- D. Xuất ra kết quả của 2 số a, b.

**(Đúng)**

**Câu 223**

Đoạn chương trình sau cho thực hiện:

```
#include <stdio.h>

#include <conio.h>

int main()

{
int n;
scanf("%d",&n);

}
```

A. Nhập vào số n có kiểu số nguyên.

(Đúng)

B. Xuất ra số n kiểu số nguyên.

C. Nhập vào n có kiểu tùy ý.

D. Xuất ra số n có kiểu tùy ý.

**Câu 224**



Hàm cho nhập một kí tự từ bàn phím ngay sau khi gõ, không chờ nhấn enter và không hiện ra màn hình là hàm:

A.    `scanf();`

B.    `getchar();`

C.    `getch();`

(Đúng)

D.    `getche();`

**Câu 225**

Cho đoạn chương trình sau:

```
int ketqua(int n)
{
    int s=0;
        for (int i=1;i<=n;i++)
            s=s+i;
    return (s1);
}
```

Với  $n = 3$ , s có giá trị bằng:

A. Lỗi chương trình.

(Đúng)

B. 1

C. 2

D. 3

**Câu 226**

Cho đoạn chương trình sau:

```
int ketqua(int n)
```

```
{
```

```
int s=0;
```

```
    for (int i=1;i<=n;i++)
```

```
        s=s+i;
```

```
    return ();
```

```
}
```

Với  $n = 3$ ,  $s$  có giá trị bằng:

A. 0

B. 1

C. 2

D. Lỗi chương trình.

(Đúng)

**Câu 227**

Cho đoạn chương trình sau:

```
int ketqua(int n)
```

```
{
```

```
int s=0;
```

```
    for (int i=1;i<=n;i++)
```

```
        s=s+i;
```

```
    return (s);
```

```
}
```

Với  $n = 3$ , Hàm trả về giá trị bằng bao nhiêu?

A. 3

B. 4

C. 5

D. 6

(Đúng)

**Câu 228**

Cho đoạn chương trình sau:

```
int ketqua(int n)
```

```
{
```

```
int s=1;
```

```
    for(int i=1;i<=n;i++)
```

```
        s=s*i;
```

```
    return (s);
```

```
}
```

Với  $n = 2$ , Hàm trả về giá trị bằng bao nhiêu?

A. 0

B. 1

C. 2

(Đúng)

D. 6

**Câu 229**

Cho đoạn chương trình sau:

```
int ketqua(int n)
```

```
{
```

```
int s=0;
```

```
    for (int i=1;i<=n;i++)
```

```
        s=s*i;
```

```
    return (s);
```

```
}
```

Với  $n = 3$ , Hàm trả về giá trị bằng bao nhiêu?

A. 0

(Đúng)

B. 1

C. 2

D. 3

**Câu 230**

Cho đoạn chương trình sau:

```
int ketqua(int n)
{
    int s=1;
    for (int i=1;i<=n;i++)
        s=s+i;
    return (s);
}
```

Với  $n = 3$ , Hàm trả về giá trị bằng bao nhiêu?

A. 0

B. 3

C. 5

D. 7

(Đúng)

**Câu 231**

Mảng là:

A. Một tập hợp các phần tử có kiểu dữ liệu riêng và chung tên gọi.

B. Một tập hợp các phần tử có kiểu dữ liệu riêng và có tên gọi riêng cho mỗi phần tử.

C. Một tập hợp các phần tử không cùng kiểu dữ liệu và tên gọi

D. Một tập hợp các phần tử có cùng kiểu dữ liệu và chung tên gọi. (Đúng)

**Câu 232**

Khai báo `int m[10];` cho biết?

A. Mảng có tên `m` và có 10 phần tử.

B. Mảng có tên `m` và có 10 phần tử cùng kiểu số nguyên. (Đúng)

C. Mảng có tên `m` và có số phần tử tùy ý kiểu số nguyên.

D. Mảng có tên `m` và có 9 phần tử.

**Câu 233**

Cho khai báo mảng:



```
int m[3];
```

Giá trị phần tử thứ 2 của mảng là:

A. m[0]

B. m[1]

(Đúng)

C. m[2]

D. m[3]

**Câu 234**

Cho khai báo mảng:

```
int m[]={4,2,6,1,7};
```

Phần tử m[3] có giá trị bằng:

A. 1

(Đúng)

B. 6

C. 7

D. 4

**Câu 235**

Cho khai báo mảng:

```
int x[]={4,2,6};
```

Phát biểu nào dưới đây là đúng:

A.  $x[1]=4, x[2]=2, x[3]=6$

B.  $x[0]=4, x[1]=2, x[2]=6$

**(Đúng)**

C.  $x[0]=2, x[1]=4, x[2]=6$

D.  $x[0]=6, x[1]=2, x[2]=4$

**Câu 236**

Cho đoạn chương trình sau:

```
void mang1chieu(int a[], int n)
{
    for(int i=0;i<n;i++)
    {
        printf("nhap vao phan tu thu%d:",i+1);
        scanf("%d",&a[i]);
    }
}
```

Với  $n = 5$ , chương trình trên cho thực hiện:

- A. Nhập vào 1 mảng 5 phần tử.
- B. Nhập vào 1 mảng 5 phần tử kiểu số nguyên.
- C. Nhập vào một mảng với số phần tử tùy ý.

**(Đúng)**

- D. Nhập vào 1 mảng 10 phần tử.

**Câu 237**

Cho đoạn chương trình sau:

```
void mang1chieu(int a[], int n)
{
    int t=0;
        for(int i=0;i<n;i++)
            t=t+a[i];
        printf("%d\n",t);
}
```

Với  $n = 5$ , chương trình trên thực hiện:

- A. Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.
- B. Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.
- C. Tính tổng giá trị 4 phần tử của mảng kiểu số nguyên.

(Đúng)

- D. Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.

**Câu 238**

Cho đoạn chương trình sau:

```
void mang1chieu(int a[], int n)
{
    int t=1;
        for(int i=0;i<n;i++)
            t=t*a[i];
        printf("%d\n",t);
}
```

Với  $n = 6$ , chương trình trên thực hiện:

- A. Tính tích giá trị 6 phần tử của mảng kiểu số nguyên.

(Đúng)

- B. Tính tổng giá trị 6 phần tử của mảng kiểu số nguyên.

- C. Tính tích giá trị 5 phần tử của mảng kiểu số nguyên.

- D. Tính tổng giá trị 5 phần tử của mảng kiểu số nguyên.

**Câu 239**

Cho đoạn chương trình sau:

```
void mang(int a[], int n)
{
    int m=a[0];
        for(int i=1;i<n;i++)
            if (m >a[i])
                m=a[i];
        printf("%d\n",m);
}
```

Với  $n = 3$ , chương trình trên thực hiện:

- A. Tìm giá trị phần tử bé nhất trong 3 phần tử của mảng. (Đúng)
- B. Tìm giá trị phần tử lớn nhất trong 3 phần tử của mảng.
- C. Tính tổng 3 phần tử của mảng.

D. Tính tích 3 phần tử của mảng.

**Câu 240**

Cho đoạn chương trình sau:

```
void mang1chieu(int a[], int n)
{
    int tich=1,tong=0;
    for(int i=0;i<n;i++)
        if (a[i]%2==0)
            tich=tich*a[i];
        else
            tong=tong +a[i];
    printf("tich cac phan tu chan:%d\n",tich);
    printf("tong cac phan tu le:%d\n",tong);
}
```

Với  $n = 4$ , chương trình trên cho thực hiện:

A. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số lẻ, tích các phần tử số chẵn. **(Đúng)**

- B. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng các phần tử số chẵn, tích các phần tử số lẻ.
- C. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tổng 4 phần tử trên.
- D. Nhập vào 4 phần tử mảng kiểu số nguyên, in ra tích 4 phần tử trên.

**Câu 241**

Điền nội dung còn thiếu trong câu sau: “Giải thuật đệ quy là giải thuật của nó có lời gọi tới ....”

- A. Một giải thuật khác đã biết kết quả.
- B. Chính nó nhưng với phạm vi nhỏ hơn. (Đúng)
- C. Chính nó.
- D. Chính nó nhưng với phạm vi lớn hơn.



**Câu 242**

Phát biểu nào sau đây là đúng với giải thuật đệ quy:

- A. Giải thuật đệ quy không bao giờ dừng.
- B. Giải thuật đệ quy vừa tốn bộ nhớ vừa chạy chậm. **(Đúng)**
- C. Sau mỗi lần có lời gọi đệ quy thì kích thước của bài toán lớn hơn trước.
- D. Giải thuật đệ quy ít tốn bộ nhớ vì cấu trúc lệnh gọn gàng.

**Câu 243**

Cho hàm đệ quy sau:

```
int F(int n)
{
    if (n <= 0) return 1;
    return n * F(n-1);
}
```

Sau mỗi lần gọi đệ quy thì giá trị của n là:

A.  $n=0$

B.  $n=1$

C. Tăng lên 1

D. Giảm đi 1

(Đúng)

**Câu 244**

Cho hàm đệ quy sau:

```
int F(int n)
```

```
{
```

```
if (n<= 0) return 1;
```

```
return n*F(n-1);
```

```
}
```

Dòng lệnh "if (n<= 0) return 1;" có nghĩa là:

A. Lặp 1 lần.

B. Lặp vô hạn.

C. Điều kiện không thực hiện đệ quy.

D. Điều kiện dừng đệ quy

(Đúng)

**Câu 245**

Hàm đệ quy sau giải bài toán gì ?

```
int S(int n)
{
    if (n <= 0) return 0;
    return n + S(n-1);
}
```

A. Tính tổng các số từ 0 đến n

(Đúng)

B. Tính tổng các số chẵn từ 0 đến n

C. Tính tổng các số lẻ từ 1 đến n

D. Tìm số Fibonacci ở vị trí thứ  $n$ .

**Câu 246**

Cho hàm đệ quy sau:

```
int U( int n)
```

```
{
```

```
if (n<6) return n;
```

```
return U(n-1)+U(n-2)+U(n-3)+U(n-4)+U(n-5);
```

```
}
```

Khi  $n=6$ , kết quả  $U(n)$  bằng bao nhiêu ?

A. 6

B. 9

C. 15

(Đúng)

D. 20

**Câu 247**

Hàm đệ quy cho kết quả thế nào ?

```
int U( int n)
```

```
{
```

```
return U(n-1)+U(n-2)+U(n-3)+U(n-4)+U(n-5);
```

```
}
```

A. 15

B. Chương trình báo lỗi.

C. Lặp vô hạn vì không có điều kiện dừng

(Đúng)

D. 20

**Câu 248**

Cho giải thuật đệ quy sau:

```
int F(int n)
```

```
{
```

```
if (n<=2) return 1;
```

```
return F(n-1) + F(n-2);
```

```
}
```

Ý nghĩa của giải thuật trên là gì ?

A. Tính tổng các số nguyên lớn hơn 2.

B. Tính tổng các số nguyên nhỏ hơn n.

C. Tính tổng 2 số nguyên gần nhau.

D. Tìm số Fibonacci ở vị trí thứ n.

(Đúng)

**Câu 249**

Cho giải thuật đệ quy sau:

```
int Fibo(int n)
{
    if (n<=2) return 1;
    return Fibo(n-1) + Fibo(n-2);
}
```

Khi n=4 kết quả của bài toán trên là:

A. 3

(Đúng)

B. 10

C. 11

D. 8

**Câu 250**

Danh sách liên kết là:

- A. Danh sách liên kết là một cấu trúc dữ liệu bao gồm một tập các nút, mà mỗi nút bao gồm: Dữ liệu và con trỏ liên kết đến nút tiếp theo. **(Đúng)**
- B. Danh sách được lưu dưới dạng mảng.
- C. Danh sách liên kết là một cấu trúc dữ liệu tĩnh.
- D. Danh sách có dạng  $(a_1, a_2, \dots, a_n)$ .

**Câu 251**

```
void Initialize (NODEPTR &First)
```

```
{  
First = NULL;  
}
```

Thủ tục trên thực hiện chức năng gì trong danh sách liên kết ?

A. Khởi tạo một danh sách liên kết rỗng. (Đúng)

B. Cấp phát bộ nhớ cho nút

C. Kiểm tra danh sách liên kết rỗng

D. Huỷ một nút

**Câu 252**

```
void Free_Node(NODEPTR p)
```

```
{  
free(p);
```



}

Thủ tục trên thực hiện chức năng gì trong danh sách liên kết ?

A. Khởi tạo nút mới.

B. Huỷ một nút đã cấp phát.

(Đúng)

C. Cấp phát bộ nhớ cho 1 nút.

D. Kiểm tra danh sách rỗng.

**Câu 253**

Ưu điểm của danh sách liên kết so với mảng:

A. Bổ sung hoặc xóa một phần tử bất kỳ trong danh sách liên kết dễ dàng hơn mảng.

(Đúng)

B. Tìm kiếm trên danh sách liên kết nhanh.

- C. Kích thước của danh sách liên kết phải khai báo trước.
- D. Khi khai báo danh sách liên kết ít tốn bộ nhớ hơn mảng.

**Câu 254**

Ngăn xếp (Stack) là:

- A. Là một danh sách liên kết trong đó phép bổ sung một phần tử và phép loại bỏ một phần tử khỏi ngăn xếp luôn luôn thực hiện ở đỉnh của ngăn xếp. **(Đúng)**
- B. Là một danh sách liên kết trong đó phép bổ sung một phần tử và phép loại bỏ một phần tử khỏi ngăn xếp luôn luôn thực hiện ở tại một vị trí bất kì trong danh sách.
- C. Là một danh sách liên kết trong đó phép bổ sung một phần tử được thực hiện ở một đầu và phép loại bỏ được thực hiện ở đầu kia của ngăn xếp.
- D. Là một danh sách tuyến tính trong đó phép bổ sung một phần tử vào

ngăn xếp được thực hiện ở một đầu và phép loại bỏ không thực hiện được.

**Câu 255**

Ngăn xếp hoạt động theo cấu trúc:

A. FIFO (First In First Out)

B. FOLO (Fisrt Out Last Out)

C. LILO (Last In Last Out)

D. LIFO (Last In First Out)

(Đúng)

**Câu 256**

Khi đổi một số nguyên từ hệ thập phân sang hệ nhị phân thì ta thực hiện phép chia liên tiếp cho 2 và lấy các số dư theo chiều ngược lại. Cách sắp xếp này chính là cơ chế hoạt động của:

A. Bản ghi (Record)

B. Ngăn xếp (Stack)

(Đúng)

C. Hàng đợi (Queue)

D. Mảng (Array)

**Câu 257**

Khi thêm phần tử vào ngăn xếp (Stack) sẽ thực hiện ở đâu ?

A. Vị trí đỉnh của ngăn xếp.

(Đúng)

B. Vị trí đầu của ngăn xếp.

C. Ở vị trí bất kì trừ đỉnh và đầu của ngăn xếp.

- D. Không thêm được.

**Câu 258**

Hàng đợi (Queue) là:

- A. Một danh sách liên kết trong đó phép bổ sung và phép loại bỏ một phần tử được thực hiện ở tại một vị trí bất kì trong danh sách.
- B. Một danh sách liên kết trong đó phép bổ sung một phần tử được thực hiện ở một đầu. Phép loại bỏ không thực hiện được.
- C. Một danh sách liên kết trong đó phép bổ sung hay loại bỏ phần tử được thực hiện ở một đầu danh sách gọi là đỉnh (Top)
- D. Một danh sách liên kết trong đó phép bổ sung phần tử ở một đầu gọi là **(Đúng)** lối sau (rear) và phép loại bỏ phần tử được thực hiện ở đầu kia, gọi là lối trước (front).

**Câu 259**

Hàng đợi hoạt động theo cấu trúc:

A. LIFO (Last In First Out)

B. FIFO (First In First Out)

(Đúng)

C. LILO (Last In Last Out)

D. FILO (First In Last Out)

**Câu 260**

Khi cài đặt hàng đợi (Queue) bằng mảng, đoạn chương trình sau thực hiện chức năng gì ?

```
int IsQueue(Queue Q)
{
    if (Q.count == 0) return 1;
    return 0;
}
```

A. Kiểm tra hàng đợi đầy

B. Kiểm tra hàng đợi rỗng

(Đúng)

C. Xóa phần tử trong hàng đợi

D. Khởi tạo hàng đợi rỗng

**Câu 261**

Khi cài đặt hàng đợi (Queue) bằng mảng, đoạn chương trình sau thực hiện chức năng gì ?

```
void IsQueue(Queue &Q)
```

```
{
```

```
    Q.front=0;
```

```
    Q.rear=-1;
```

```
    Q.count=0;
```

```
}
```

A. Kiểm tra hàng đợi đầy

B. Khởi tạo hàng đợi rỗng

(Đúng)

- C. Kiểm tra hàng đợi rỗng
- D. Xóa phần tử ở đầu hàng đợi

**Câu 262**

Hoạt động nào giống với cơ chế hoạt động của ngăn xếp (Stack) ?

- A. Xếp hàng mua vé xem phim
- B. Xếp chồng hàng hóa vào thùng.
- C. Mua hàng online.
- D. Không phải các đáp án trên.

**(Đúng)**

**Câu 263**



Khi cài đặt ngăn xếp (Stack) bằng mảng, hàm Pop() trong đoạn chương trình sau dùng để:

```
int Pop(Stack &s)
{
if(StackEmpty(s))
printf("Ngan xep da rong!");
else
return s.nut[s.top--];
}
```

- A. Lấy phần tử đầu tiên ra khỏi Stack
- B. Xóa bỏ một phần tử bất kì khỏi Stack
- C. Xóa bỏ một dãy các phần tử ra khỏi Stack
- D. Lấy một phần tử cuối cùng ra khỏi đỉnh Stack

**(Đúng)**

**Câu 264**

Khi cài đặt ngăn xếp (Stack) bằng mảng, hàm Push() trong đoạn chương trình sau dùng để:

```
void Push(Stack &s, int x)
{
if(StackFull(s))
printf("\nNgan xep da day!");
else
s.nut[++s.top]=x;
}
```

A. Bổ sung một dãy các phần tử vào đỉnh Stack.

B. Bổ sung một phần tử vào đầu Stack

C. Bổ sung một phần tử vào đỉnh Stack

(Đúng)

D. Bổ sung một phần tử bất kì vào Stack

**Câu 265**

Nếu các phần tử được đưa vào ngăn xếp (Stack) theo thứ tự "1 1 0 1" thì các phần tử được loại khỏi Stack theo thứ tự nào ?

A. 1101

B. 1, 1, 0, 1

C. 1, 0, 1, 1

(Đúng)

D. 1001

**Câu 266**

Khi bổ sung một phần tử mới vào hàng đợi (Queue) cần kiểm tra:

A. Hàng đợi có đầy không.

(Đúng)

B. Hàng đợi có rỗng không.

C. Hàng đợi có bao nhiêu phần tử.

D. Hàng đợi có bao nhiêu giá trị bằng 0.

**Câu 267**

Thao tác nào là đúng với hàng đợi (Queue) ?

A. Thêm phần tử vào cuối hàng đợi.

(Đúng)

B. Xóa phần tử ở cuối hàng đợi.

C. Thêm phần tử vào đầu hàng đợi

D. Không thể xóa phần tử ở vị trí đầu tiên của hàng đợi.

**Câu 268**

Cây nhị phân là cây:

A. Có tối đa 2 nút con

(Đúng)

B. Có nhiều nút con.

C. Bậc của cây lớn hơn 2

D. Không có nút lá trong cây.

**Câu 269**

Nút gốc trên cây nhị phân tìm kiếm có đặc điểm gì ?

A. Lớn hơn nút cây con bên phải.

B. Nhỏ hơn nút cây con bên trái.

C. Là nút có bậc lớn nhất trong cây.

- D. Là nút có giá trị lớn hơn các nút cây con bên trái và nhỏ hơn các nút cây con bên phải. (Đúng)

**Câu 270**

Trong biểu diễn dữ liệu dưới dạng cây, bậc của cây chính là:

- A. Bậc cao nhất của một nút trên cây. (Đúng)
- B. Bậc cao nhất của nút gốc.
- C. Bậc cao nhất của nút lá.
- D. Tổng số nút trên cây.

**Câu 271**

Trong biểu diễn dữ liệu dưới dạng cây, nút có bậc bằng 0 gọi là:

- A. Phần tử cuối cùng trong cây

B. Không có đáp án nào đúng

C. Gốc

D. Lá (Đúng)

**Câu 272**

Mỗi nút trong cây có bao nhiêu nút con ?

A. 2 nút con

B. 3 nút con

C. 1 nút con

D. Nhiều nút con hoặc không có nút con nào. (Đúng)

**Câu 273**

Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là  $i$  thì vị trí của nút con bên trái là:

A.  $2*i$

(Đúng)

B.  $2*i + 1$

C.  $i-1$

D.  $i+1$

**Câu 274**

Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là  $i$  thì vị trí của nút con bên phải là:

A.  $i-1$



B.  $2*i$

C.  $2*i + 1$

(Đúng)

D.  $i+1$

**Câu 275**

Cây nhị phân tìm kiếm có đặc điểm gì ?

A. Giá trị nút gốc lớn hơn nút con bên trái và nhỏ hơn nút con bên phải.

(Đúng)

B. Có nhiều nút con.

C. Chỉ có 1 nút con

D. Cây nhị phân tìm kiếm không có nút nhánh.

**Câu 276**

Nút nhánh trong cây là nút:

- A. Có bậc khác 0.
- B. Có bậc lớn nhất trong cây.
- C. Không có nút cha.
- D. Có bậc khác 0 và không phải nút gốc.

**(Đúng)**

**Câu 277**

Bậc của cây phụ thuộc vào gì ?

- A. Nút gốc

B. Nút lá

C. Nút nhánh

D. Nút có nhiều con nhất.

(Đúng)

**Câu 278**

Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là 3 thì vị trí tương ứng của nút con bên trái sẽ là:

A. 4

B. 7

C. 6

(Đúng)

D. 2

**Câu 279**

Khi lưu trữ cây nhị phân dưới dạng mảng, nếu vị trí của nút cha trong mảng là 3 thì vị trí tương ứng của nút con bên phải sẽ là:

A. 4

B. 7 (Đúng)

C. 6

D. 2

**Câu 280**

Có 3 cách duyệt cây nhị phân: “Duyệt trước – Duyệt giữa – Duyệt sau”, được bố trí theo thứ tự là:

A. NLR – LRN – LNR

B. NLR – LNR – LRN

(Đúng)

C. LNR – NLR – LRN

D. LNR – LRN – NLR

**Câu 281**

Duyệt cây nhị phân theo thứ tự trước được thực hiện theo thứ tự :

A. Duyệt cây con trái theo thứ tự trước, thăm gốc, duyệt cây con phải theo thứ tự sau.

B. Thăm gốc trước, duyệt cây con trái theo thứ tự giữa, duyệt cây con phải sau.

C. Duyệt cây con trái theo thứ tự trước, thăm gốc, duyệt cây con phải theo thứ tự sau.

- D. Thăm gốc, duyệt cây con trái theo thứ tự trước, duyệt cây con phải theo thứ tự sau. (Đúng)

**Câu 282**

Duyệt cây nhị phân theo thứ tự sau được thực hiện theo thứ tự:

- A. Duyệt cây con trái theo thứ tự trước, thăm gốc, duyệt cây con phải theo thứ tự sau.
- B. Duyệt cây con trái theo thứ tự trước, duyệt cây con phải và thăm gốc sau cùng. (Đúng)
- C. Thăm gốc trước, duyệt cây con trái theo thứ tự giữa, duyệt cây con phải sau.
- D. Thăm gốc, duyệt cây con trái , duyệt cây con phải theo thứ tự sau.

**Câu 283**

Ý tưởng phương pháp sắp xếp chọn (Selection Sort)

- A. Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp.
- B. Chọn phần tử bé nhất đối chỗ cho phần tử đầu tiên. Tương tự đối với phần tử nhỏ thứ hai, ba... **(Đúng)**
- C. Lần lượt lấy phần tử của dãy chèn vị trí thích hợp của nó trong dãy.
- D. Bắt đầu từ cuối dãy đến đầu dãy, ta lần lượt so sánh hai phần tử kế tiếp nhau, nếu phần tử nào bé hơn được cho lên vị trí trên.

**Câu 284**

Ý tưởng phương pháp sắp xếp nổi bọt (Bubble Sort) là:

- A. Chọn phần tử bé nhất đối chỗ cho phần tử đầu tiên. Tương tự đối với phần tử nhỏ thứ hai, ba...
- B. Xuất phát từ cuối dãy số, đổi chỗ các cặp phần tử kế cận, để đưa phần tử nhỏ nhất lên đầu dãy. **(Đúng)**

- C. Lần lượt lấy phần tử của dãy chèn vị trí thích hợp của nó trong dãy bằng cách đẩy các phần tử lớn hơn xuống.
- D. Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp.

**Câu 285**

Ý tưởng phương pháp sắp xếp chèn (Insertion Sort) là:

- A. Bắt đầu từ cuối dãy đến đầu dãy, ta lần lượt so sánh hai phần tử kế tiếp nhau, nếu phần tử nào nhỏ hơn được đứng vị trí trên.
- B. Chọn phần tử bé nhất đổi chỗ cho phần tử đầu tiên. Tương tự đối với phần tử nhỏ thứ hai, ba...
- C. Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp.
- D. Lần lượt lấy phần tử của danh sách chèn vị trí thích hợp của nó trong dãy (Đúng)



bằng cách đẩy các phần tử lớn hơn xuống.

**Câu 286**

Cho dãy số {6 1 3 0 5 7 9 2 8 4}. Khi áp dụng phương pháp sắp xếp chọn (Selection Sort) để sắp xếp dãy số tăng dần thì sau lần lặp đầu tiên của giải thuật kết quả sẽ là:

A. {0 1 3 6 5 7 9 2 8 4}

(Đúng)

B. {0 1 2 6 5 7 9 3 4 8}

C. {0 1 2 6 5 7 9 3 8 4}

D. {0 1 2 3 4 5 6 7 8 9}

**Câu 287**

Cho dãy số {4 7 0 9 2 5 3 1 8 6}. Khi áp dụng phương pháp sắp xếp nổi bọt (Bubble Sort) để sắp xếp dãy số tăng dần thì sau lần lặp đầu tiên của giải thuật kết quả sẽ là:

A. {0 4 7 1 9 2 5 3 6 8}

(Đúng)

B. {0 1 2 3 4 7 5 9 6 8}

C. {0 7 4 1 9 2 5 3 6 8}

D. {0 1 2 4 7 3 9 5 6 8}

**Câu 288**

Cho dãy số {4 0 2 8 5 9 6 1 3 7}. Khi áp dụng phương pháp sắp xếp chèn (Insert Sort) để sắp xếp dãy số tăng dần thì sau lần lặp đầu tiên của giải thuật kết quả sẽ là:

A. {0 2 4 8 5 9 6 1 3 7}

B. {0 1 4 8 5 9 6 2 3 7}

C. {0 1 2 8 5 9 6 4 3 7}

D. {0 4 2 8 5 9 6 1 3 7}

(Đúng)

**Câu 289**

Thao tác “Xuất phát từ cuối dãy, đổi chỗ các cặp phần tử kế cận....” là ý tưởng của giải thuật sắp xếp nào sau đây:

A. Sắp xếp chọn

B. Sắp xếp chèn

C. Sắp xếp nổi bọt

(Đúng)

D. Sắp xếp nhanh

**Câu 290**

Thao tác “Chọn phần tử nhỏ nhất trong dãy số, đổi chỗ cho phần tử đầu tiên...” là ý tưởng của giải thuật sắp xếp nào sau đây:

A. Sắp xếp chọn

(Đúng)

B. Sắp xếp chèn

C. Sắp xếp nổi bọt

D. Sắp xếp trộn

**Câu 291**

Thao tác “Xem phần tử đầu tiên  $a_1$  như một dãy đã sắp xếp, thêm phần tử  $a_2$  vào vị trí  $a_1$  để được đoạn  $a_1a_2$  có thứ tự....” là ý tưởng của giải thuật sắp xếp nào sau đây:

A. Sắp xếp chọn

B. Sắp xếp chèn

(Đúng)

C. Sắp xếp nổi bọt

D. Sắp xếp trộn

**Câu 292**

Ý tưởng phương pháp sắp xếp nhanh (Quick Sort) là:

- A. Bắt đầu từ cuối dãy đến đầu dãy, ta lần lượt so sánh hai phần tử kế tiếp nhau. Nếu phần tử nào nhỏ hơn được đứng vị trí trên.
- B. Chọn phần tử bé nhất xếp vào vị trí thứ nhất bằng cách đổi chỗ phần tử bé nhất với phần tử thứ nhất. Tương tự đối với phần tử nhỏ thứ hai, ba...
- C. Lần lượt chia dãy phần tử thành hai dãy con bởi một phần tử khoá (dãy con trước khoá gồm các phần tử nhỏ hơn khoá và dãy còn lại gồm các phần tử lớn hơn khoá). **(Đúng)**
- D. Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp.

**Câu 293**

Phương pháp sắp xếp nhanh (Quick Sort) chính là phương pháp:

- A. Trộn
- B. Phân đoạn (Đúng)
- C. Vun đống
- D. Chèn

**Câu 294**

Ý tưởng phương pháp sắp xếp trộn (Merge Sort) là:

- A. Lần lượt chia dãy phần tử thành hai dãy con bởi một phần tử khoá (dãy con trước khoá gồm các phần tử nhỏ hơn khoá và dãy còn lại gồm các phần tử lớn hơn khoá).
- B. Chọn phần tử bé nhất xếp vào vị trí thứ nhất bằng cách đổi chỗ phần tử

bé nhất với phần tử thứ nhất; Tương tự đối với phần tử nhỏ thứ hai,ba...

- C. Bắt đầu từ cuối dãy đến đầu dãy, ta lần lượt so sánh hai phần tử kế tiếp nhau, nếu phần tử nào nhỏ hơn được đứng vị trí trên.
- D. Phân đoạn dãy thành nhiều dãy con và lần lượt trộn hai dãy con thành dãy lớn hơn, cho đến khi thu được dãy ban đầu đã được sắp xếp. **(Đúng)**

**Câu 295**

Trong các giải thuật sắp xếp dưới đây, giải thuật nào dùng phương pháp "Chia để trị" ?

- A. Sắp xếp chèn (Insert Sort)
- B. Sắp xếp nổi bọt (Bubble Sort)
- C. Sắp xếp chọn (Selection Sort)
- D. Sắp xếp trộn (Merge Sort) **(Đúng)**

**Câu 296**

Ý tưởng của giải thuật tìm kiếm nhị phân:

- A. So sánh X lần lượt với các phần tử thứ nhất, thứ hai,... của dãy cho đến khi gặp phần tử có khoá cần tìm.
- B. Lần lượt chia dãy thành hai dãy con dựa vào phần tử khoá, sau đó thực hiện việc tìm kiếm trên hai đoạn đã chia.
- C. Tìm kiếm dựa vào cây nhị tìm kiếm.
- D. Tại mỗi bước tiến hành so sánh từ khoá X với phần tử ở giữa của dãy đã được sắp xếp tăng dần. Kết quả so sánh này quyết định tìm kiếm nằm ở nửa trên hay nửa dưới của dãy hiện hành. **(Đúng)**

**Câu 297**

Ý tưởng của giải thuật tìm kiếm tuần tự:



- A. Nếu giá trị cần tìm nhỏ hơn gốc thì thực hiện tìm kiếm trên cây con trái, ngược lại ta việc tìm kiếm được thực hiện trên cây con phải.
- B. Tại mỗi bước tiến hành so sánh từ khoá X với phần tử ở giữa của dãy đã được sắp xếp tăng dần. Kết quả so sánh này quyết định tìm kiếm nằm ở nửa trên hay nửa dưới của dãy hiện hành.
- C. Lần lượt chia dãy thành hai dãy con dựa vào phần tử khoá, sau đó thực hiện việc tìm kiếm trên hai đoạn đã chia.
- D. So sánh từ khoá X lần lượt với các phần tử thứ nhất, thứ hai,... của dãy (Đúng)  
cho đến khi gặp phần tử có khoá cần tìm.

**Câu 298**

Ưu điểm của tìm kiếm nhị phân so với tìm kiếm tuần tự là:

- A. Nhanh hơn vì giảm được nửa thời gian tìm kiếm. (Đúng)
- B. Nhanh hơn vì giải thuật đơn giản hơn.

C. Không cần sắp xếp có thứ tự trước.

D. Ít tốn bộ nhớ hơn.

**Câu 299**

Thao tác nào cần làm trước khi thực hiện tìm kiếm nhị phân trong mảng:

A. Tính tổng các phần tử trong mảng

B. Sắp xếp mảng có thứ tự (tăng dần hoặc giảm dần)

**(Đúng)**

C. Xác định từ khóa cần tìm phải có trong mảng.

D. Đếm có bao nhiêu phần tử trong mảng.

**Câu 300**

Trong tìm kiếm nhị phân, phần tử trong dãy ở vị trí nào quyết định thành công của thuật toán:

A. Vị trí đầu tiên.

B. Vị trí cuối cùng.

C. Vị trí giữa mảng.

(Đúng)

D. Vị trí bất kì.

**Câu 301**

Xét bảng Employee(emp\_code, full\_name, basic, date\_of\_birth)

Để hiển thị lương nhân viên (basic) sau khi tăng 10% từ bảng employee, ta viết câu lệnh

A. `Select emp_code, basic*.10 from employee order by emp_code`

B. `Select emp_code, basic+basic*.01 from employee order by emp_code`

C. `Select emp_code, basic+10 from employee order by emp_code`

- D.      `Select emp_code, basic+basic*.10 from employee order by emp_code`      **(Đúng)**

**Câu 302**

Câu lệnh tạo bảng nào dưới đây là **ĐÚNG**:

- A.      `CREATE TABLE VatTu (mavt CHAR(4) NOT NULL, tenvt VARCHAR(30),  
phantram TINYINT  
CONSTRAINT PRKVatTumavt PRIMARY KEY(mavt), CONSTRAINT  
DEF_VatTu_phantram DEFAULT 20 FOR phantram)`
- B.      `CREATE TABLE VatTu (mavt CHAR(4) NOT NULL, tenvt VARCHAR(30),  
phantram TINYINT  
CONSTRAINT DEF_VatTu_phantram DEFAULT 20 FOR phantram  
CONSTRAINT PRKVatTumavt PRIMARY KEY(mavt))`
- C.      `CREATE TABLE VatTu (mavt CHAR(4) NOT NULL, tenvt VARCHAR(30),  
phantram TINYINT  
CONSTRAINT DEF_VatTu_phantram DEFAULT 20 CONSTRAINT  
PRKVatTumavt PRIMARY KEY(mavt))`
- D.      `CREATE TABLE VatTu (mavt CHAR(4) NOT NULL, tenvt VARCHAR(30),`      **(Đúng)**

phantram TINYINT

CONSTRAINT DEF\_VatTu\_phantram DEFAULT 20, CONSTRAINT  
PRKVatTumavt PRIMARY KEY(mavt))

**Câu 303**

Xét bảng Sinh\_Vien( MaSV, Ho, Ten, So\_dien\_thoai) với So\_dien\_thoai có kiểu VARCHAR(10) và cho phép NULL.

Sinh_Vien			
MaSV	Ho	Ten	So_dien_thoai
A001	Lê Văn	Nghĩa	8965134
A002	Nguyễn Thị	Lan	
A003	Trần Văn	Thái	NULL

Thực hiện câu lệnh truy vấn sau:

```
SELECT MaSV, ISNULL(so_dien_thoai, 'Chưa có') FROM SinhVien
```

Cho biết kết quả nào dưới đây là **ĐÚNG**:

**A.** Kết quả

MaSV	So_dien_thoai
A001	8965134
A002	Chưa có

A003

Chưa có

**B.**    Kết quả

MaSV

So\_dien\_thoai

A001

8965134

A002

Chưa có

A003

**C.**    Kết quả

(Đúng)

MaSV

So\_dien\_thoai

A001

8965134

A002

A003

Chưa có

**D. Kết quả**

MaSV	So_dien_thoai
A001	8965134
A002	
A003	

**Câu 304**

Thực hiện các câu lệnh sau:

```
DECLARE @thong_bao CHAR(10), @ma_loi INT
```

```
SET @thong_bao = 'Lỗi:'
```

```
SET @ma_loi = 1
```

```
IF @ma_loi=1
```

```
SET @thong_bao = @thong_bao+'không xác định'
```

```
PRINT @thong_bao
```

Khi in ra biến @thong\_bao thì kết quả nào dưới đây là **ĐÚNG**:

**A.** 'Lỗi: '

**(Đúng)**

**B.** 'Lỗi: không xác định'

C. Hệ thống báo lỗi tràn chuỗi trong biến @thong\_bao

D. Tất cả đều không đúng

**Câu 305**

Xét bảng Sinh\_Vien(Masv, Ho, Ten, Makh) và bảng Ket\_Qua(Masv, Mamh, Diem).

Tên cột	Ý nghĩa
Masv	Mã Sinh viên
Makh	Mã khoa
Mamh	Mã môn
Diem	Điểm số

Để liệt kê các Sinh viên trong khoa **TH** có điểm trung bình lớn nhất thì câu lệnh truy vấn nào dưới đây là **ĐÚNG**:

A. `SELECT kq.Masv, Ho, Ten, AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv WHERE`  
`makh='TH' AND AVG(Diem) IN (SELECT TOP 1 AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`WHERE makh='TH' GROUP BY kq.Masv ORDER BY AVG(Diem) DESC)`

(Đúng)



GROUP BY kq.Masv, Ho, Ten

- B. `SELECT kq.Masv, Ho, Ten, AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`GROUP BY kq.Masv, ho, ten HAVING AVG(Diem) NOT IN (SELECT TOP 1`  
`AVG(Diem) FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON`  
`kq.masv=sv.masv WHERE makh='TH' GROUP BY kq.masv ORDER BY`  
`AVG(Diem) DESC)`
- C. `SELECT kq.Masv, Ho, Ten, AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`WHERE Makh='TH' GROUP BY kq.Masv, ho, ten`  
  
`HAVING AVG(Diem) > ALL ( SELECT TOP 1 AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`WHERE Makh='TH' GROUP BY kq.Masv ORDER BY AVG(Diem) DESC)`
- D. `SELECT kq.Masv, Ho, Ten, AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`WHERE Makh='TH' GROUP BY kq.Masv, ho, ten`  
  
`HAVING AVG(Diem) EXIST ( SELECT TOP 1 AVG(Diem)`  
`FROM Ket_Qua kq INNER JOIN Sinh_Vien sv ON kq.Masv=sv.Masv`  
`WHERE Makh='TH' GROUP BY kq.Masv ORDER BY AVG(Diem) DESC)`

**Câu 306**

Xét bảng PHIEU\_NHAP(sopn, ngaynhap) có các mẫu tin sau:

Phieu_Nhap	
Sopn	Ngaynhap
N001	2014-04-01
N002	2014-04-03
N003	2014-04-10
N004	2014-05-01
N005	2014-05-01

Để lọc ra các phiếu nhập phát hành trong tháng 4/2014 thì câu lệnh nào dưới đây sẽ lọc có dữ liệu:

A. `SELECT * FROM Phieu_Nhap  
WHERE CONVERT(CHAR(6),ngaynhap, 112)='2014-04'`

B. `SELECT * FROM Phieu_Nhap  
WHERE CONVERT(CHAR(6),ngaynhap, 112)='201404'`

(Đúng)

C. `SELECT* FROM Phieu_Nhap  
WHERE CONVERT(CHAR(7),ngaynhap,112) ='2014-04'`

- D. `SELECT * FROM Phieu_Nhap  
WHERE CONVERT(CHAR(6),ngaynhap,12)-201404'`

**Câu 307**

Để tính, tổng các số nguyên từ 1 đến 10 bằng cách sử dụng vòng lặp WHILE. Xét đoạn lệnh dưới đây:

```
DECLARE @i INT, @tong INT
```

```
SET @i=1
```

```
SET @tong=0
```

```
WHILE @i<=10
```

```
SET @tong=@tong+@i
```

```
SET @i=@i+1
```

```
PRINT @tong
```

Trong các trường hợp thì trường hợp nào dưới đây là **ĐÚNG**:

- A. Kết quả in ra của biến @tong là: 55
- B. Kết quả in ra của biến @tong là: 0
- C. Kết quả in ra của biến @tong là: 1

D. Vòng lặp không bao giờ dừng

(Đúng)

**Câu 308**

Xét các bảng Sinh\_Vien và Khoa



Để thống kê tổng số Sinh viên từng Khoa chưa có học bổng (cột hocbong=0) thì câu lệnh nào sau đây là **ĐÚNG**

A. SELECT Khoa.makh, tenkh. Count(\*) as dem

(Đúng)

FROM Sinh\_Vien Inner Join Khoa On Sinh\_Vien.makh = Khoa.makh  
WHERE hocbong = 0 GROUP BY Khoa.makh, tenkh

B. SELECT Khoa.makh, tenkh. Count(\*) as dem

FROM Sinh\_Vien Inner Join Khoa On Sinh\_vien.makh = Khoa.makh  
WHERE hocbong > 0 GROUP BY Khoa.makh, tenkh

- C.     SELECT Khoa.makh, tenkh. Count(\*) as dem  
  
FROM Sinh\_Vien Inner Join Khoa On Sinh\_vien.makh = Khoa.makh  
  
GROUP BY Khoa.makh, tenkh WHERE hocbong = 0
- D.     SELECT Khoa.makh, tenkh, Count(\*) as dem  
  
FROM Sinh\_Vien Inner Join Khoa On Sinh\_Vien.makh = Khoa.makh  
GROUP BY Khoa.makh, tenkh HAVING hocbong=0

**Câu 309**

Xét bảng Sinh\_Vien.

MaSV	Ho	Ten	MaKH
A01	Nguyễn Văn	Hùng	AV
A02	Trần Thị	Tuyết	AV
B01	Lê Văn	Nghĩa	TH
B02	Hoàng Ngọc	Phương	TH
C01	Trương Thị	Châu	TR
C02	Mai Tứ	Phượng	TR

Câu lệnh:

SELECT MaSV, Ho, Ten, MaKH FROM Sinh\_Vien WHERE MaKH IN

(SELECT MaKH FROM Sinh\_Vien WHERE MaSV IN

(SELECT MaSV FROM Sinh\_Vien WHERE Ten = N'Phượng'))

Kết quả nào dưới đây là **ĐÚNG** khi thực hiện câu lệnh trên

<b>A.</b>	MaSV	Ho	Ten	MaKH
	C01	Trương Thị	Châu	TR
	C02	Mai Từ	Phượng	TR

<b>B.</b>	MaSV	Ho	Ten	MaKH	<b>(Đúng)</b>
	B01	Lê Văn	Nghĩa	TH	
	B02	Hoàng Ngọc	Phượng	TH	

<b>C.</b>	MaSV	Ho	Ten	MaKH
	B02	Hoàng Ngọc	Phượng	TH
	C02	Mai Từ	Phượng	TR

D.	MaSV	Ho	Ten	MaKH
	B01	Lê Văn	Nghĩa	TH
	B02	Hoàng Ngọc	Phương	TH
	C01	Trương Thị	Châu	TR

**Câu 310**

Xét các bảng Sinh_Vien và Khoa
--------------------------------



Câu lệnh:

```
SELECT Khoa.makh, tenkh, Count(*) as dem
FROM Sinh_Vien Inner Join Khoa On Sinh_Vien.makh =Khoa.makh
WHERE nam=1
GROUP BY Khoa.makh, tenkh
UNION ALL
SELECT Khoa.makh, tenkh, Count(*) as dem FROM Sinh_Vien Inner Join Khoa On
Sinh_Vien.makh = Khoa.makh
WHERE nam=0 GROUP BY Khoa.makh, tenkh
```

Kết quả nào dưới đây là **ĐÚNG** khi thực hiện câu lệnh trên

A.	MaKH	TenKH	Dem
	AV	Anh van	4



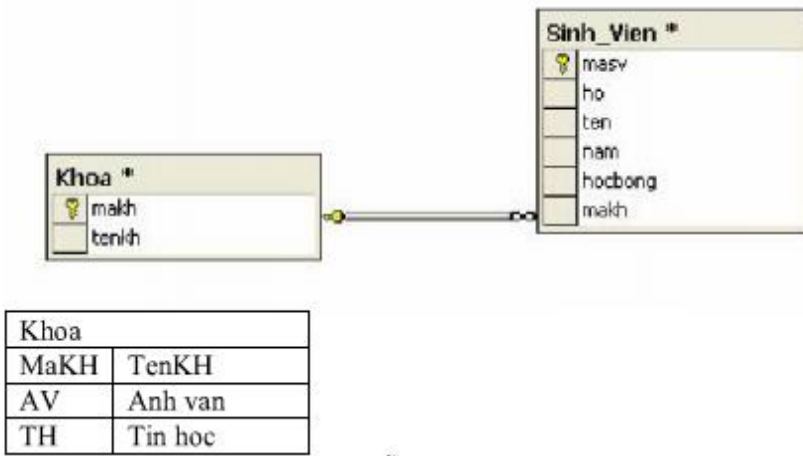
<b>B.</b>	AV	Anh van	2
	TH	Tin hoc	2

<b>C.</b>	MaKH	TenKH	Dem	<b>(Đúng)</b>
	AV	Anh van	1	
	TH	Tin hoc	1	
	AV	Anh van	1	
	TH	Tin hoc	1	

<b>D.</b>	MaKH	TenKH	Dem
	AV	Anh van	1
	TH	Tin hoc	1

**Câu 311**

**Xét bảng Sinh\_Vien**



Giả sử bảng Sinh\_Vien là bảng rỗng. Câu lệnh:

```
INSERT INTO Sinh_Vien (masv, ho, ten, nam, makh)
VALUES('A01','Le Van','Nghĩa', 1, 'av')
```

Khi thực hiện câu lệnh trên thì xảy ra lỗi. Chọn các nguyên nhân có thể gây ra lỗi

- A. Giá trị của cột hocbong không cho phép NULL
  - B. Trùng mã Sinh viên
  - C. Mã khoa phải là chữ HOA. Ví dụ là AV
  - D. Giá trị của cột hocbong bắt buộc phải là một số dương và không cho
- (Đúng)**

phép NULL

**Câu 312**

Câu lệnh ALTER TABLE khi sử dụng để thao tác với các CONSTRAINT thì không thể:

- A. Thêm hoặc xoá các constraint
- B. Bật hay tắt chế độ kiểm tra dữ liệu trên constraint
- C. Đổi tên một constraint
- D. Xác định thứ tự các constraint mà SQL sẽ dùng để kiểm tra dữ liệu **(Đúng)**

**Câu 313**

Chọn lệnh để định nghĩa kiểu dữ liệu mới có tên DiaChi.

Kiểu dữ liệu DiaChi dùng để khai báo những cột dữ liệu có thể chứa tối đa 100 ký tự và KHÔNG cho phép Null.

- A. `sp_addtype DiaChi, 'Varchar(100)', 'Not Null'` **(Đúng)**
- B. `spaddtype DiaChi, 'varbinary(100)', 'Not Null'`
- C. `sp_addtype DiaChi, 'varchar()', 'Not Null'`
- D. `sp_addtype DiaChi, 'varchar(100)'`

**Câu 314**

Giá trị Null từng cột của biểu thức ON được xử lý thế nào khi tiến hành JOIN hai bảng

- A. Các dòng có chứa giá trị NULL của bảng này chỉ kết hợp các dòng có giá trị NULL ở bảng còn lại
- B. Các dòng có chứa giá trị NULL sẽ không được trả về trừ khi sử dụng OUTER JOIN
- C. Các dòng có chứa giá trị NULL của bảng này kết hợp với tất cả các dòng bảng còn lại **(Đúng)**

- D. Các dòng có chứa giá trị NULL của bảng này không kết hợp được với bất cứ dòng nào ở bảng còn lại

**Câu 315**

Để tạo bảng sao lưu VatTu\_BK từ bảng VatTu. Chọn câu lệnh **ĐÚNG**:

- A. `SELECT * FROM VatTu INTO VatTu BK`
- B. `CREATE TABLE VatTu_BK FROM VatTu`
- C. `CREATE TABLE VatTu_BK AS SELECT * FROM VatTu`
- D. `SELECT * INTO VatTu_BK FROM VatTu`

**(Đúng)**

**Câu 316**

Bạn cần tạo CSDL tên Toys có ít nhất 50MB trống dành cho dữ liệu và kích cỡ lúc đầu 15MB dành cho transaction log. Lệnh nào sau đây bạn nên dùng để tạo CSDL?

- A. CREATE DATABASE Toys ON (NAME = Toys\_dat, FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.mdf',SIZE = 50 ) LOG ON (NAME = Toys\_log, FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.ldf', SIZE = 15 ) **(Đúng)**
- B. CREATE DATABASE Toys ON (NAME = Toys\_dat,FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.mdf',SIZE = 60 )
- C. CREATE DATABASE Toys ON (FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.mdf',SIZE = 51200 ) LOG ON (FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.ldf',SIZE = 15360 )
- D. CREATE DATABASE Toys ON (NAME = Toys\_dat,FILENAME = 'D:\Microsoft SQL Server\MSSQL\Data\Toys.mdf',SIZE = 50 )

**Câu 317**

Đổi đổi từ mã ASCII sang ký tự. Chọn phát biểu **ĐÚNG**:

A. Hàm ASCII

B. Hàm CHR

C. Hàm CHAR (Đúng)

D. Hàm STUFF

**Câu 318**

Lệnh nào dùng để thêm cột Emp\_EmailId vào bảng Emp\_details với kiểu dữ liệu là varchar và kích cỡ là 10:

A. Alter COLUMN EMP\_Details ADD (Emp\_EmailId) vchar(10) NOT NULL

B. Alter COLUMN EMP\_Details ADD (Emp\_EmailId) varchar(10)

C. Alter EMP\_Details ADD Emp\_EmailId varchar(10)

D. Alter Table EMP\_Details ADD Emp\_EmailId varchar(10)

(Đúng)

**Câu 319**

Đối tượng nào sau đây để tăng tốc truy tìm các hàng trong 1 bảng hay view

A. Ràng buộc (constraints)

B. Triggers

C. Stored procedures

D. Chỉ mục (Indexes)

(Đúng)

**Câu 320**

Chọn câu lệnh thích hợp để xuất ra cửa sổ kết quả của Query chuỗi có dạng sau (lưu ý giá trị 17/06/2014 là ngày hiện hành khi thực thi câu lệnh):



Ngày hiện hành là:17/06/2014

A. `print 'Ngày hiện hành là:' + convert(char(10),date(),103)`

B. `print 'Ngày hiện hành là:' + convert(char(10),getdate(), 103)` (Đúng)

C. `print 'Ngày hiện hành là:' + convert(getdate(),103)`

D. `print 'Ngày hiện hành là:' + getdateO`

**Câu 321**

Giả sử bảng KHOA là bảng rỗng. Xét giao tác sau:

BEGIN TRAN

INSERT INTO Khoa (makh,tenkh) VALUES ('AV', 'Khoa Anh')

INSERT INTO Khoa (makh,tenkh) VALUES (TH,'Khoa Tin hoc')

SAVE TRAN sp\_l

INSERT INTO Khoa (makh,tenkh) VALUES ('LY', 'Khoa Vat ly')

ROLLBACK TRAN sp\_l

## COMMIT TRAN

Cho biết kết quả nào dưới đây là **ĐÚNG** khi kết thúc giao tác trên và thực hiện lệnh SELECT makh FROM KHOA:

A. AV, TH, LY

B. AV, TH

C. LY

(Đúng)

D. Tất cả đều không đúng

## **Câu 322**

Rắc rối nào xảy ra khi primary key cho phép có giá trị NULL:

A. Khó xác định tính duy nhất của các hàng

(Đúng)

B. Khó xác định tính duy nhất của các cột

- C. Khó kết nối các bảng
- D. Khó xác định foreign key

**Câu 323**

Bảng tmpCount được tạo ra với khoá chính ID là trường số tự động và trường page (số trang) phải nhỏ hơn 50. Lệnh nào sau đây là hợp lệ:

- A. Create table tmpCount (id int identity(1,1), page int constraint check page between 1 and 49) **(Đúng)**
- B. Create table tmpCount (id int identity(1,1), page char constraint check page between 1 and 49)
- C. Create table tmpCount (id char identity(1,1), page int constraint check page between 1 and 49)
- D. Create table tmpCount (id int identity(1,1), page int constraint check page between 1 and 50)

**Câu 324**

Giả sử bảng KHOA là bảng rỗng và có cấu trúc: KHOA(makh,tenkh). Xét giao tác sau:

BEGIN TRAN Cap1

INSERT INTO Khoa (makh,tenkh) VALUES ('AV','Khoa Anh')

BEGIN TRAN Cap2

INSERT INTO Khoa (makh,tenkh) VALUES ('LY','Khoa Vat ly')

COMMIT TRAN Cap2

ROLLBACK TRAN Cap1

Cho biết kết quả nào dưới đây là **ĐÚNG** khi thực hiện lệnh:

SELECT makh FROM Khoa

A. AV, LY

B. AV

C. LY

D. Bảng KHOA vẫn là bảng rỗng

(Đúng)

**Câu 325**

Trong SQL Server, để tạo và xóa một user\_defined data type, ta có thể sử dụng:

- A. create uddt and drop uddt
- B. sp\_addtype and sp\_unbindtype
- C. sp\_bindtype and sp\_droptype
- D. sp\_addtype and sp\_droptype

(Đúng)

**Câu 326**

Constraint nào được dùng để gán giá trị hằng cho 1 cột?

- A. PRIMARY KEY

B. UNIQUE

C. CHECK

D. DEFAULT

(Đúng)

**Câu 327**

john muốn tạo thêm 1 ràng buộc CHECK mà ràng buộc này sẽ bỏ qua không kiểm tra điều kiện của các hàng đã có sẵn trong 1 bảng. Tùy chọn nào nên được dùng trong khi tạo ràng buộc này?

A. WITH NOCHECK

(Đúng)

B. WITH CHECK

C. WITH RECOMPILE

D. NOCHECK

**Câu 328**

Câu lệnh SQL nào sau đây dùng để xóa cả cấu trúc và dữ liệu của bảng có tên là STUDENT?

- A. DROP TABLE STUDENT (Đúng)
- B. DELETE TABLE STUDENT
- C. REMOVE TABLE STUDENT
- D. DELETE FROM STUDENT

**Câu 329**

Câu lệnh SQL nào sau đây dùng để loại bỏ chỉ dữ liệu từ bảng có tên là STUDENT, trong khi để lại nguyên vẹn cấu trúc bảng?

- A. DROP TABLE STUDENT

B. DELETE TABLE STUDENT

C. REMOVE TABLE STUDENT

D. DELETE FROM STUDENT

(Đúng)

**Câu 330**

Từ khóa SQL nào dùng để loại bỏ các hàng trùng lặp trong kết quả của một truy vấn SQL SELECT?

A. UNIQUE

B. SORT

C. ORDER BY

D. DISTINCT

(Đúng)



**Câu 331**

Câu truy vấn nào hiển thị tổng số dòng dữ liệu có trong bảng emp?

- A.     Select rowcount from emp
- B.     Select totalrows from emp
- C.     Select count(\*) from emp
- D.     Select count\_rows from emp

**(Đúng)**

**Câu 332**

Mệnh đề được dùng để nhóm các record giống nhau lại với nhau?

- A.     GROUP BY

**(Đúng)**

- B. HAVING
- C. WHERE
- D. ORDER BY

**Câu 333**

Phát biểu nào sau đây là đúng khi nói về hàm COUNT(\*)?

- A. Nó có thể truyền một tham số
- B. Nó trả về số các dòng khác nhau sử dụng từ khóa DISTINCT
- C. Nó trả về số các dòng kể cả dòng có chứa giá trị NULL (Đúng)
- D. Nó trả về số các dòng bỏ qua các dòng có chứa giá trị NULL

**Câu 334**

Khi thiết kế bảng, bạn đặt trường Emp\_id làm khóa chính. Toàn vẹn dữ liệu nào hỗ trợ cho trường hợp này?

- A. Toàn vẹn thực thể (Đúng)
- B. Toàn vẹn tham chiếu
- C. Toàn vẹn miền
- D. Toàn vẹn do người dùng định nghĩa

**Câu 335**

Update Products set UnitPrice = UnitPrice + (UnitPrice \*.10) where UnitPrice between 60 and 70  
Kết quả của câu lệnh trên là?

- A. Tăng 10% giá (UnitPrice) trong bảng Products, cho những sản phẩm có giá là 60 hoặc 70

B. Tăng 10% giá (UnitPrice) trong bảng Products, cho những sản phẩm có giá từ 60 đến 70 (Đúng)

C. Tăng 10% giá (UnitPrice) trong bảng Products, cho những sản phẩm có giá từ 60 đến 70, không kể 2 giá trị 60 và 70

D. Tất cả các phương án đều không đúng.

**Câu 336**

Cho lược đồ cơ sở dữ liệu sau:

SinhVien(MaSV, HoSV, TenSV, Phai, NgaySinh, DiaChi, DienThoai, MaLop) Lop(MaLop, TenLop, MaKhoa, GVCN)

KetQua(MaSV, MaMH, LanThi, Diem)

MonHoc(MaMH, TenMH, SoTinChi)

Liệt kê danh sách các Sinh viên gồm (MaSV, HoSV, TenSV) có điểm thi môn CSDL cao nhất?

A. **Select K.MaSV, HoSV, TenSV From SinhVien S, KetQua K Where S.MaSV=K.MaSV And MaMH='CSDL' And Diem>=ALL(Select Diem From KetQua where MaMH ='CSDL')** (Đúng)

B. **Select K.MaSV, HoSV, TenSV From SinhVien S, KetQua K Where**

And MaMH='CSDL' And Diem <=(Select max(Diem) From KetQua where MaMH ='CSDL')

- Câu 337**

**A.**     SELECT ProductName FROM products WHERE (UnitPrice < 10),  
          (UnitsInStock > 5) **(Đúng)**

**B.**     SELECT ProductName FROM products WHERE (UnitPrice < 10) OR NOT  
          (UnitsInStock > 5)

- C. `SELECT ProductName FROM products WHERE UnitPrice < 10 OR UnitsInStock > 5`
- D. `SELECT ProductName FROM products WHERE UnitPrice < 10 AND UnitsInStock > 5`

**Câu 338**

Cho 1 bảng với cấu trúc sau: NhanVien(MaNV, TenNV, Luong, NgayBatDauVaoLam) hãy tạo câu truy vấn SQL để tìm nhân viên có lương cao nhất.

- A. `SELECT *`  
`FROM NhanVien ORDER BY Luong DESC;`
- B. `SELECT top 1 *`  
`FROM NhanVien ORDER BY Luong;`
- C. `SELECT top 1 *`  
`FROM NhanVien ORDER BY Luong DESC;`
- D. `SELECT top 1 *`

**(Đúng)**

FROM NhanVien;

**Câu 339**

Cho 1 bảng với cấu trúc sau: NhanVien(MaNV, TenNV, Luong, NgayBatDauVaoLam), Tạo câu truy vấn SQL để tìm tất cả các nhân viên bắt đầu vào làm việc trong năm 2015?

- A.     SELECT \*  
        FROM NhanVien  
        WHERE NgayBatDauVaoLam=2015;
- B.     SELECT \*  
        FROM NhanVien  
        WHERE NgayBatDauVaoLam>='01/01/2015';
- C.     SELECT  
        FROM NhanVien  
        WHERE NgayBatDauVaoLam>='01/01/2015';
- D.     SELECT \*  
        FROM NhanVien  
        WHERE NgayBatDauVaoLam>=2015;

**(Đúng)**

**Câu 340**

Một bảng có n dòng và có một khoá duy nhất (unique key). Giả sử trong 1 dòng của bảng, khoá có giá trị null. Nếu 1 lệnh truy vấn thực hiện một phép kết bằng trên bảng này với chính nó thông qua khoá duy nhất này. Kết quả của câu truy vấn trả về bao nhiêu dòng?

- A. ít hơn n dòng.
- B. Nhiều hơn n dòng.
- C. không có dòng nào cả.
- D. n dòng

**(Đúng)**

**Câu 341**

Cho lược đồ quan hệ sau:

SinhVien (MaSV, HoSV, TenSV, DiaChi, ThanhPho, MaVung)

Dựa vào lược đồ quan hệ trên, thay đổi mã vùng của Sinh viên ở Tp.HCM thành 08?

- A. Update SinhVien Set MaVung='08' Where ThanhPho='Tp.HCM'

**(Đúng)**



- B. Update MaVung='08' Where ThanhPho='Tp.HCM'
- C. Update SinhVien Where ThanhPho='Tp.HCM' And MaVung='08'
- D. Tất cả các phương án trên.

**Câu 342**

Trong lệnh cập nhật dữ liệu UPDATE, nếu ta không dùng mệnh đề WHERE thì:

- A. Sẽ không thực hiện lệnh cập nhật
- B. Chỉ một số dòng được cập nhật
- C. Không có dòng nào được cập nhật
- D. Tất cả các dòng sẽ được cập nhật

**(Đúng)**

**Câu 343**

Cho câu lệnh SQL:

```
SELECT count(SOHD) FROM HOADON GROUP BY NGÀYXUAT;
```

Câu lệnh này sẽ cho kết quả:

- A. Số hóa đơn tương ứng với ngày xuất
- B. Tổng số hóa đơn
- C. Số lượng hóa đơn xuất ra theo ngày (Đúng)
- D. Câu lệnh không đúng

**Câu 344**

Cho câu truy vấn sau:

```
SELECT E.EMPLOYEE_NAME, P.PAYSCALE  
FROM EMPLOYEES E  
WHERE 500>ALL(SELECT P.BONUSPAY
```

FROM PAYLIST P

WHERE P.EMPLOYEE\_ID = E.EMPLOYEE\_ID);

Câu truy vấn trên **KHÔNG ĐÚNG** ở điểm nào:

- A. Câu lệnh SELECT không thể lấy cột PAYSCALE (Đúng)
- B. Subquery không thể trả về nhiều dòng
- C. Subquery không thể truy cập dữ liệu ở bảng EMPLOYEES
- D. Không có lỗi nào.

**Câu 345**

Cho quan hệ Cungcap

MaNCC	Mahang	Soluong
101	1	20
101	2	30
102	1	14
103	4	21

104	4	10
105	1	5

Kết quả câu truy vấn SQL:

Select Mahang From Cungcap Group By Mahang

Having Count(\*) = (Select Max(Count(\*)) From Cungcap Group By Mahang)

A. 1 (Đúng)

B. 2

C. 3

D. 4

**Câu 346**

Sara muốn bảo đảm rằng chỉ có 1 một bản ghi cho mỗi sinh viên trong bảng students. Loại bảo toàn dữ liệu nào sau đây được sử dụng:

- A. Bảo toàn thực thể (Entity integrity) (Đúng)
- B. Bảo toàn miền (Domain integrity)
- C. Bảo toàn tham chiếu (Referential integrity)
- D. Bảo toàn thủ tục ( Procedure integrity)

**Câu 347**

Trong database Sales có bảng Khachhang và bảng Hoadon. Mỗi một hoá đơn chỉ bán cho một khách hàng, và mỗi một khách hàng có thể không có hoặc có nhiều hoá đơn. Nên chọn các cột làm khóa chính và khoá ngoại trong các bảng như thế nào?

- A. Trong bảng Khachhang chọn khoá chính là Mã hóa đơn (MaHD). Trong bảng Hoadon chọn khoá chính là Mã khách hàng (MaKH)
- B. Trong bảng Khachhang chọn khoá chính là Mã khách hàng (MaKH). Trong bảng Hoadon chọn khoá chính là Mã khách hàng (MaKH)

C. Trong bảng Kháchhang chọn khoá chính là Mã khách hàng (MaKH).  
Trong bảng Hoadon chọn khoá ngoại là Mã hoa don (MaHD)

D. Trong bảng Kháchhang chọn khoá chính là Mã khách hàng (MaKH). (Đúng)  
Trong bảng Hoadon chọn khoá ngoại là Mã khách hàng (MaKH)

**Câu 348**

Lệnh nào sau đây là **không đúng** khi nói về tính bảo toàn dữ liệu?

A. Ràng buộc PRIMARY KEY được dùng để bảo đảm là không có giá trị NULL hay trùng nhau trong 1 hàng đã cho của bảng

B. Ràng buộc CHECK được dùng để bảo đảm là chỉ có 1 bộ các giá trị xác định được phép nhập vào 1 cột đã cho của bảng

C. Ràng buộc UNIQUE KEY được dùng để bảo đảm là các cột giống nhau trong bảng chứa các giá trị giống nhau (Đúng)

D. Ràng buộc DEFAULT để bảo đảm là 1 giá trị hằng sẽ được đưa vào 1 cột khi người dùng không xác định nó

**Câu 349**

Xác định toán tử nào cho phép chọn 1 giá trị phù hợp với 1 trong các giá trị của một danh sách:

A. IN

(Đúng)

B. AND

C. OR

D. NOT

**Câu 350**

Lệnh nào sau đây dùng để xoá tất cả các hàng của 1 một bảng mà không kích hoạt trigger

A. DELETE

B. TRUNCATE

(Đúng)

C. DROP

D. REMOVE

**Câu 351**

Tính chất nào sau đây là cần thiết để 2 bảng có thể union (hợp) với nhau:

A. Các thuộc tính phải cùng loại dữ liệu

(Đúng)

B. Các thuộc tính phải cùng dữ liệu

C. Các thuộc tính có dữ liệu là duy nhất

D. Các thuộc tính có loại dữ liệu khác nhau



**Câu 352**

Mệnh đề nào sau đây được dùng để hiển thị tên của tất cả các nhóm, kể cả những nhóm bị loại trừ khỏi mệnh đề WHERE từ 1 bảng được cho

A. GROUP BY

B. HAVING

C. GROUP BY ALL

(Đúng)

D. COMPUTE

**Câu 353**

Phát biểu SQL nào sau đây được dùng để hiển thị chỉ những hàng của bảng Titles mà trường ytd\_sales chưa được tính toán:

A. SELECT \* FROM Titles WHERE Ytd\_Sales = NULL

B. `SELECT * FROM Titles WHERE Ytd_Sales is NULL` (Đúng)

C. `SELECT * FROM Titles WHERE Ytd_Sales =0`

D. `SELECT * FROM Titles WHERE Ytd_Sales =??`

**Câu 354**

Bảng Student được tạo ra bởi lệnh sau:

`CREATE TABLE Student`

`( OwnerID int IDENTITY (1, 1) NOT NULL, FirstName char(20) NULL, LastName char(30) NULL, BirthDate date NULL, CONSTRAINT PK_Owners PRIMARY KEY (Owner ID) )`

Khi thông tin về 1 sinh viên (student) nào đó được tạo ra, ngày sinh của sinh viên đó nếu không được nhập vào thì database cần tạo 1 thông báo lỗi. Bạn cần chọn cách thực hiện nào để kiểm tra tính hợp lệ của trường ngày sinh và tạo thông báo lỗi khi không hợp lệ

A. Thêm ràng buộc CHECK vào trường BirthDate (ngày sinh)

B. Tạo 1 rule rồi gán nó cho trường BirthDate

C. Thay đổi cấu trúc bảng sao cho trường BirthDate không được phép giải trị null

- D. Tạo trigger trên bảng Student để xét tính hợp lệ của trường birthDate (Đúng)

**Câu 355**

lệnh SQL sau:

```
SELECT Customers.name, Customers.ord_id, Orders.ord_date  
FROM Customers INNER JOIN Orders  
WHERE Customers.cust_id = Orders.cust_id
```

Cả hai bảng đều có 10000 hàng. Truy vấn này mất nhiều thời gian để thực hiện, Để cải thiện việc thực thi truy vấn ta nên dùng đối tượng nào sau đây:

- A. Triggers
- B. Stored procedures
- C. Views (Đúng)
- D. Batches

**Câu 356**

Select \* from ExternalCandidate Where Testscore < 20 and Testscore >=80

Đầu ra của lệnh trên là gì

- A. Truy vấn sẽ hiển thị các bản ghi của những ứng viên (candidate) mà có điểm thi (test score) nhỏ hơn 20 hoặc lớn hơn hay bằng 80
- B. Truy vấn sẽ hiển thị các bản ghi của chỉ những ứng viên nào mà điểm lớn hơn 20
- C. Truy vấn sẽ hiển thị các bản ghi của chỉ những ứng viên nào mà điểm lớn hơn hay bằng 80
- D. Truy vấn sẽ thực thi được nhưng không hiển thị bất kỳ bản ghi nào **(Đúng)**

**Câu 357**

Thủ tục hệ thống (system-stored procedure) nào dùng để cung cấp thông tin về các đối tượng Db hay loại dữ liệu của người dùng

- A. sp\_droptype

B. Sp\_helptext

C. Sp\_help

(Đúng)

D. Sp\_bindrule

**Câu 358**

Một kết nối giữa 2 bảng mà không có bất kỳ điều kiện nào trong mệnh đề ON thì được gọi là:

A. Inner join

B. Outer join

C. Cross join

(Đúng)

D. Equi join

**Câu 359**

Sử dụng lệnh Select với tùy chọn nào để giới hạn số hàng được trả về:

A. Với mệnh đề WHERE

B. Với mệnh đề GROUP BY

C. Với mệnh đề TOP

D. Với từ khoá DISTINCT

(Đúng)

**Câu 360**

Lệnh nào sau đây dùng để hiển thị 10 hàng đầu tiên trong bảng Titles:

A. `SELECT * FROM Titles WHERE count(*) = 10`

B. `SELECT count(10) FROM Titles`

C. `SELECT TOP(10) FROM Titles`

D. `SELECT TOP 10 * FROM Titles`

**(Đúng)**