

## CHƯƠNG 1 Các khái niệm cơ bản về lập trình

**Câu 1: Khẳng định nào sau đây là SAI khi nói về ngôn ngữ lập trình C ?**

- A. C phân biệt chữ hoa và chữ thường.
- B. C là ngôn ngữ lập trình bậc cao.
- C. C có thể dùng để phát triển các phần mềm ứng dụng, điều khiển robot...
- D. C là ngôn ngữ máy tính có thể trực tiếp hiểu và thực hiện được.**

**Câu 2: Lý do ta nên dùng ngôn ngữ lập trình bậc cao khi viết chương trình?**

- A. Máy tính có thể trực tiếp hiểu và thực hiện.
- B. Gần với ngôn ngữ tự nhiên, cú pháp đơn giản, dễ hiểu, dễ học.**
- C. Có thể viết thoải mái không cần theo quy tắc của ngôn ngữ lập trình.
- D. Chỉ dùng phục vụ trong học tập, không có tính ứng dụng trong phát triển ứng dụng phần mềm, lập trình games...

**Câu 3: Phương pháp nào Không dùng để biểu diễn thuật toán**

- A. Dùng ngôn ngữ tự nhiên
- B. Dùng sơ đồ tư duy**
- C. Dùng sơ đồ khói
- D. Dùng mã giả

**Câu 4: Ngôn ngữ lập trình là gì?**

- A. Ngôn ngữ Pascal hoặc C
- B. Phương tiện để soạn thảo văn bản trong đó có chương trình
- C. Phương tiện diễn đạt thuật toán
- D. Công cụ để diễn tả thuật toán thành chương trình cho máy tính thực hiện**

**Câu 5: Thuật toán là gì?**

- A. Các mô hình và xu hướng được sử dụng để giải quyết vấn đề
- B. Một dãy các chỉ dẫn từng bước để giải quyết vấn đề.**
- C. Một ngôn ngữ lập trình.

D. Một thiết bị phần cứng lưu trữ dữ liệu.

**Câu 6: Lợi thế của việc sử dụng sơ đồ khối so với sử dụng ngôn ngữ tự nhiên để mô tả thuật toán là gì?**

- A. Sơ đồ khối dễ vẽ.
- B. Vẽ sơ đồ khối không tốn thời gian
- C. Sơ đồ khối tuân theo một tiêu chuẩn quốc tế nên con người dù ở bất kể quốc gia nào cũng có thể hiểu.**
- D. Sơ đồ khối dễ thay đổi.

**Câu 7: Mục đích của sơ đồ khối là gì?**

- A. Để mô tả các chỉ dẫn cho máy tính "hiểu" về thuật toán.
- B. Để mô tả các chỉ dẫn cho con người hiểu về thuật toán.**
- C. Để chỉ dẫn cho máy tính thực hiện thuật toán.
- D. Để soạn thảo ra các chương trình

**Câu 8: Phát biểu nào sau đây là đúng?**

- A. Trình tự thực hiện các bước trong thuật toán không quan trọng.
- B. Mỗi bài toán chỉ có duy nhất một thuật toán để giải.
- C. Trong thuật toán, với dữ liệu đầu vào luôn xác định được kết quả đầu ra.**
- D. Một thuật toán có thể không có đầu vào và đầu ra.

**Câu 9: Ngôn ngữ C thuộc loại?**

- A. Ngôn ngữ bậc cao
- B. Hợp ngữ
- C. Ngôn ngữ máy
- D. Cả A, B, C đều sai

**Câu 10: Hàm nào là hàm mà mọi chương trình C đều cần có?**

- A. start()
- B. system()

- C. main()
- D. program()

**Câu 11: Dấu nào được dùng để đánh dấu điểm kết thúc một dòng code?**

- A. :
- B. ;**
- C. {
- D. }

**Câu 12: Lệnh nào sau đây là đúng?**

- A. /\* Chú thích \*/
- B. \*\* Chú thích \*\*
- C. /\* Chú thích \*/**
- D. { Chú thích }

**Câu 13: Kiểu nào không phải là kiểu dữ liệu trong ngôn ngữ lập trình C?**

- A. float
- B. real**
- C. int
- D. double

**Câu 14: Toán tử nào sau đây dùng để so sánh 2 biến với nhau**

- A. =
- B. :=
- C. ==**
- D. equal

**Câu 15: Chọn toán tử logic and?**

- A. &
- B. |
- C. |&
- D. &&**

**Câu 16: Cần thêm thư viện nào để chương trình sau hoạt động chính xác?**

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

```
void main()
{
    printf("%f", log(1.9));
    getch();
}
```

- A. stdlib.h
- B. math.h**
- C. log.h
- D. dos.h

**Câu 17: Giả sử có 2 số nguyên, biểu thức nào dưới đây viết không đúng theo cú pháp của ngôn ngữ lập trình C?**

- A. a\*=b
- B. a=b
- C. a/=b
- D. a&=b**

**Câu 18: Kiểu dữ liệu nào dưới đây được coi là kiểu dữ liệu cơ bản trong ngôn ngữ lập trình C**

- A. Kiểu con trỏ
- B. Kiểu char**
- C. Kiểu mảng
- D. Kiểu hợp

**Câu 19: Tìm giá trị biểu thức: !(1 && !(0 || 1))**

- A. True**
- B. False
- C. Không tính được

**Câu 20: Hàm printf() nằm trong tệp thư viện chuẩn nào?**

- A. stdio.h
- B. conio.h
- C. stdlib.h
- D. ouput.h

## Chương 2: Các phần tử cơ bản của ngôn ngữ C

**Câu 1: Chọn phát biểu đúng khi nói về biến?**

- A. Biến là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị có thể được thay đổi trong quá trình thực hiện chương trình.
- B. Biến là đại lượng bất kì.
- C. Biến là đại lượng không thay đổi trong quá trình thực hiện chương trình.
- D. Biến là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị không thay đổi trong quá trình thực hiện chương trình.

**Câu 2: Phát biểu nào dưới đây là hợp lý nhất khi nói về hằng?**

- A. Hằng có thể lưu trữ nhiều loại giá trị khác nhau.
- B. Hằng là đại lượng được đặt tên và có giá trị thay đổi trong quá trình thực hiện chương trình.
- C. Hằng được chương trình dịch bỏ qua.
- D. Hằng là đại lượng nhận giá trị trước khi chương trình thực hiện.**

**Câu 3: Mối ngôn ngữ lập trình có thành phần cơ bản là?**

- A. Cú pháp và ngữ nghĩa
- B. Bảng chữ cái, cú pháp
- C. Bảng chữ cái, cú pháp và ngữ nghĩa**
- D. Bảng chữ cái và ngữ nghĩa

**Câu 4: Lệnh #include dùng để làm gì?**

- A. Khai báo các câu lệnh được sử dụng trong chương trình. Phải tạo các câu lệnh trước thì mới sử dụng được trong chương trình C
- B. Thông báo cho bộ tiền biên dịch thêm các thư viện chuẩn trong C. Các lệnh được sử dụng trong thân chương trình phải có prototype nằm trong các thư viện chuẩn này.**
- C. Thông báo trong chương trình sử dụng các lệnh tính toán, thông báo các biến sử dụng trong thân chương trình.
- D. Không có đáp án đúng.

**Câu 5: Trong cấu trúc chương trình C có bao nhiêu hàm main()?**

- A. 1
- B. 2
- C. 3
- D. 4

**Câu 6: Kết quả của chương trình sau khi thực hiện?**

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

```
int main()
{
    const int i = 5;
    i++;
    printf("%d", i);
    return 0;
}
```

- A. 5
- B. 6
- C. 0
- D. Lỗi**

**Câu 7: Kết quả của chương trình sau khi thực hiện?**

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

```
int N = 10;
```

```
void main()
{
    int N = 20;
    printf("N = %d", N);
    getch();
}
```

A. **N = 20**

B. N = 10

C. N=0

D. Lỗi

**Câu 8: Câu lệnh nào sau đây không đúng?**

A. int i = 35; i = i%5

B. short int j = 5; j = j;

C. long int k = 123L; k = k;

**D. float a = 3.14; a = a%3;**

**Câu 9: Chọn thứ tự ưu tiên các phép toán trong biểu thức  $z = x + y * z / 4 \% 2 - 1$ ?**

A. \* / % + - =

B. = \* / % + -

C. / \* % - + =

D. \* % / - + =

**Câu 10: Kết quả hiển thị ra màn hình của chương trình sau là gì?**

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

```
void main()
{
    int x = 4, y, z;
    y = --x;
    z = x--;
    printf("%d %d %d", x, y, z);
    getch();
}
```

- A. 4 3 2
- B. 4 3 3
- C. 2 3 2
- D. 2 3 3**

**Câu 11: Kết quả hiển thị ra màn hình của chương trình sau là gì?**

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

void main()
{
    int a = 10, b = 19;
    int c;
    c = (a == 10 || b < 20);
    printf("c = %d", c);
    getch();
}
```

- A. c = 1**
- B. c = 29
- C. c = 10
- D. c = 19

**Câu 12: Kết quả hiển thị ra màn hình của chương trình sau là gì?**

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

int N = 10;
void main()
{
    int N = 20;
    printf("N = %s", N);
    getch();
}
```

- A. N = 20

B. N = 10

C. N = 30

**D. Lỗi**

**Câu 13: Khai báo nào sau đây đúng?**

A. int length;

B. char int;

C. int long;

D. float double;

**Câu 14: Kết quả hiển thị ra màn hình của chương trình sau là gì?**

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

```
int main()
{
    int w = 3;
    int x = 31;
    int y = 10;
    double z = x / y % w;
    printf("%f\n", z);

    return 0;
}
```

A. 1

**B. 0**

C. 0.1

D. 10

**Câu 15: Kết quả hiển thị ra màn hình của chương trình sau là gì?**

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

```
void main()
{
    int x;
    {
        x++;
    }
    printf("%d", x);
```

```
    getch();  
}
```

- A. 1
- B. 0
- C. 2
- D. Lỗi**

**Câu 16:** Đầu nào để đánh dấu bắt đầu và kết thúc 1 khối lệnh?

- A. → và ←
- B. BEGIN và END
- C. {}**
- D. ( và )

**Câu 17:** Xâu định dạng nào dưới đây dùng để in ra một kí tự?

- A. "%f".**
- B. "%x".
- C. "%s".
- D. "%c".

**Câu 18:** Kiểu dữ liệu int (kiểu số nguyên) có thể xử lý số nguyên nằm trong khoảng nào:

- A. 0.. 255.
- B. -32768.. 32767.**
- C. -128.. 127.
- D. 0.. 65535.

**Câu 19:** Dữ liệu kí tự bao gồm:

- A. Các kí tự số chữ số.
- B. Các kí tự chữ cái.
- C. Các kí tự đặc biệt.
- D. Cả a, b và c.**

**Câu 20:** Kiểu dữ liệu nào dưới đây không được coi là kiểu dữ liệu cơ bản trong ngôn ngữ lập trình C:

- A. Kiểu enum.
- B. Kiểu char
- C. Kiểu short int.
- D. **Kiểu mảng.**

**Câu 21: Khai báo các biến: *int m,n; float x,y;* Lệnh nào sau đây sai?**

- A. n=8 ;
- B. x=0 ;
- C. y=1.5 ;
- D. **m=2.5**

### CHƯƠNG 3. CÁU TRÚC ĐIỀU KHIỂN

**Câu 1:** Lệnh nào trong các lệnh sau cho phép nhảy ra khỏi vòng lặp đến vị trí bất kì mong muốn:

- a. break;
- b. continue;
- c. goto;**
- d. Không có phương án nào.

**Câu 2 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

```
#include <stdio.h>
void main()
{
    int x = 2;
    x--;
    if(x == 1)
    {
        printf("Đại Học Công nghệ Đông Á xin chào! ");
    }
    else
        printf("Đại Học Công nghệ Đông Á ");
}
```

- a. Lỗi biên dịch
- b. Đại Học Công nghệ Đông Á xin chào!**
- c. Đại Học Công nghệ Đông Á
- d. Nothing (Không hiện thị ra kết quả gì.)

**Câu 3:** Kết quả của chương trình sau khi thực hiện?

```
#include <stdio.h>
int main()
{
    int a = 0;
    switch (a)
    {
        case '0': printf("A"); break;
        case '1': printf("B"); break;
        default : printf("ABC");
    }
    return 0;
}
```

- a. A
- b. B
- c. ABC**
- d. Cá 3 đáp án đều sai.

**Câu 4:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```
#include <stdio.h>
#define A 0
#define B 1
int main()
{
    int i = 3;
    switch ( i & 1 )
    {
        case A : printf("FALSE"); break;
        case B : printf("TRUE"); break;
        default : printf("Default");
    }
    return 0;
}
```

- a. FALSE
- b. TRUE**

C. Default

d. Cá 3 đáp án đều sai.

**Câu 5:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```
#include <stdio.h>
int main()
{
    int a = 5 ;
    switch (a)
    {
        default: a = 4 ;
        case 6 : a-- ;
        case 5 : a = a+1 ;
        case 1 : a = a-1 ;
    }
    printf("%d \n", a) ;
    return 0;
}
```

a. 4

**b. 5**

c. 3

d. 6

**Câu 6:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```
#include <stdio.h>
void main()
{
    int i ;
    for ( i = -3; i < -5; i++)
    {
        printf("Hello World");
    }
}
```

a. Hello World

**b. Nothing ( Không hiện thị ra kết quả gì ).**

c. Lỗi biên dịch chương trình.

d. Cá 3 đáp án đều sai.

**Câu 7:** Kết quả chương trình sau khi thực hiện chương trình sau ?

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

```

int main()
{
    int a = 1 ;
    while (a < 1 )
        printf("True \n");
        printf("False \n");
        return 0;
}

```

- a. True  
**b. False**  
c. Lỗi biên dịch chương trình  
d. khác

**Câu 8:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
int main()
{
    int a = 0 , b = 0 ;
    while (a < 5 , b < 10 )
    {
        a++;
        b++;
    }
    printf("%d , %d \n", a, b);
    return 0;
}

```

- a. 5, 5                   **b. 10, 10**                   c. Lỗi cú pháp                   d. Khác

**Câu 9 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
int main()
{
    int a = 0 , i = 0, b = 0 ;
    for (i = 0 ; i < 5 ; i++ )
    {
        a++;
        continue ;
        b++;
    }
    printf("\n a = %d, b = %d \n", a, b);
}

```

```

        return 0;
    }
a. a = 5, b = 5           b. a = 4, b = 4           c. a = 5, b = 0           d. Khác

```

**Câu 10 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
void main()
{
    int i ;
    for ( i = 0; i < 5; i++)
    {
        if ( i < 4)
        {
            printf ("Đại học Công nghệ Đông Á ");
            break ;
        }
    }
}

```

- a. Đại học Công nghệ Đông Á
- b. Đại học Công nghệ Đông Á ( được in 3 lần).
- c. Đại học Công nghệ Đông Á ( được in 4 lần)
- d. Đại học Công nghệ Đông Á ( được in 5 lần).

**Câu 11 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
void main()
{
    int x ;
    for ( x = 1; x <= 5; x++)
        printf("%d",x);
}

```

- a. 12345
- b. 123456
- c. 6
- d. 1234

**Câu 12 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

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

```

void main()
{
    int x , y ;
    x = y = 2 ;
    while ( --x && y++)
        printf("%d %d" , x , y);
}

```

- a. 1 3                    b. 1 2                    c. Không có gì                    d. Khác

**Câu 13 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
int main()
{
    int x = 011 , i ;
    for (i =0; i < x ; i+=3)
    {
        printf("Start");
        continue;
        printf("End");
    }
    return 0;
}

```

- a. Start End Start End

**b. Start Start Start**

c. Start Start Start Start

d. Khác

**Câu 14 :** Kết quả chương trình sau khi thực hiện chương trình sau ?

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

```
void main()
```

```
{
```

```
    int x = 3;
```

```
    switch ( x )
```

```
{
```

```

        case 0 + 1 : printf("A") ; break ;
        case 1 + 2 : printf("B") ; break ;
        default : printf("ABC");
    }
}

```

- a. A                    b. B                    c. ABC                    D. Khác

**Câu 15:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```

#include <stdio.h>
void main()
{
    int a = 40, b = 4;
    while( a!= b)
    {
        if (a>b)
            a = a-b;
        else
            b = b-a;
    }
    printf("%d",a);
}

```

- a. 2                    b. 16                    c. 4                    d. Kết quả khác

**Câu 16 :** Tìm lỗi sai trong chương trình sau( in ra kết quả là tổng của 100 và 200):

```

#include <stdio.h>
void main()
{
    int sum;

    sum= 100 + 200

    printf(" Ket qua la: " sum)
}

```

- a. Thiếu dấu chấm phẩy();.  
b. Thiếu dấu phẩy (,).  
c. Thiếu kí tự đặc tả.

d. Cả 3 ý trên đều đúng.

Câu 17. Kết quả chương trình sau khi thực hiện chương trình sau ?

```
#include <stdio.h>
void main()
{
    int a;
    for (a = 2 ; a <= 4 ; a+=2)
    {
        printf("%3d", a);
    }
}
```

- a. 2 4
- b. 1 2 3 4
- c. 2 3 4
- d. Chương trình không chạy được.

Câu 18: Kết quả in ra màn hình của chương trình sau ?

```
#include <stdio.h>
void main()
{
    int x, y;
    x = 5 ; y = 6;
    if( x <= y)
        x = x - --y ;
    printf ("%d", x);
}
```

- a. 0
- b. 1
- c. 5
- d. 6

Câu 19: Kết quả in ra màn hình của chương trình sau ?

```
#include <stdio.h>
void main()
{
    int a = 3, b = 4, x = 0;
```

```

if (a != 3)
    x++;
    x += b;

if (b == 4) {
    x--;
    x *= a;
}
printf("%d",x);
}

```

a. -3

b. 0

c. 13

**d. 9**

**Câu 20:** Kết quả in ra màn hình của chương trình sau ?

```

#include <stdio.h>
void main()
{
    int x = 3;

    if (x == 1)
        printf("One");
    else if (x == 2)
        printf("Two");
    else if (x == 3)
        printf("Three");
    else
        printf("Other");
}

```

a. One

b. Two

**c. Three**

d. Other

**Câu 21:** Kết quả in ra màn hình của chương trình sau ?

```

#include <stdio.h>
void main()
{
    int x = 3;

    switch(x) {
        case 1:
            printf("One");
        case 2:
            printf("Two");
    }
}

```

```

case 3:
    printf("Three");
default:
    printf("Other");
}

```

a. Three      b. OneTwoThree      **c. ThreeOther**      d. OneTwoThreeOthe

**Câu 22:** Kết quả in ra màn hình của chương trình sau ?

```

#include <stdio.h>
void main()
{
    int i, N = 10;
    for (i = 0; i < N; i += 2) {
        printf("%d ", i);
    }
}

```

- a. 0 1 2 3 4 5 6 7 8 9 10  
b. 0 1 2 3 4 5 6 7 8 9  
c. 0 2 4 6 8 10  
**d. 0 2 4 6 8**

**Câu 23:** Kết quả in ra màn hình của chương trình sau ?

```

#include <stdio.h>
void main()
{
    int x = 4;
    while (x) {
        printf("%d ", x--);
    }
}

```

- a. 4 3 2 1 0  
**b. 4 3 2 1**  
c. 3 2 1 0  
d. 4 3 2 1 0 -1 -2 ... (vô hạn)

**Câu 24:** Cách viết nào sau đây là đúng với cú pháp khai báo của câu lệnh if?

- a. if biểu\_thức
- b. if <biểu\_thức>
- c. if {biểu\_thức}
- d. if (biểu\_thức)**

**Câu 25:** Kết quả in ra màn hình bao nhiêu lần chuỗi Hello Wordl! của chương trình sau ?

```
#include <stdio.h>
void main()
{
    int x = 1;
    while (x <= 10) {
        printf("Hello World! ");
    }
}
```

- a. 1 lần
- b. 9 lần
- c. 10 lần
- d. Vô hạn**

**Câu 26:** Kết quả chương trình sau khi thực hiện chương trình sau ?

```
void main()
{
    int i = 3;
    i++;
    if (i == 3)
        i += 2;
    i += 2;
    printf("%d\n", i);
    getch();
}
```

- A. 7
- B. 5
- C. 6**
- D. 8

**Câu 27: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
int i;
for (i=2; i<=4; i+=2)
printf("%3d",i);
};
```

- A. 1 2 3 4
- B. 2 3 4
- C. 2 4
- D. Không đáp án nào đúng

**Câu 28: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
int i = 3;
if (!i)
    i++;
i++;

if (i == 3)
i += 2;
i += 2;
printf("%d\n", i);
getch();
```

- A. 7
- B. 5
- C. 6
- D. Không đáp án nào đúng

**Câu 29: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x = 3;
    if (x == 2); x = 0;
    if (x == 3) x++;
    else x += 2;
```

```
    printf("x = %d", x);
    getch();
}
A. 2
B. 6
C. 0
D. 4
```

**Câu 30: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x = 3;
    if (x == 2) x = 0;
    if (x == 3) x++;
    else x += 2;

    printf("x = %d", x);
    getch();
}
```

- A. 2
- B. 6
- C. 0
- D. 4

**Câu 31: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
    int i;
    i = 10;
    if(i == 20 || 30<31)
        printf("True");
    else
        printf("False");

    getch();
}
```

A. True

B. False

C. Lỗi

**Câu 32: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
    int a = 80;
    if(a++ > 80)
        printf("C/C++ %d", a);
    else
        printf("Java %d", a);

    getch();
```

A. C/C++ 80

B. C/C++ 81

C. Java 80

**D. Java 81**

**Câu 33: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
    int a;
    a = 1;
    while(a <= 1)
        if(a%2)
            printf("%d ", a++);
        else
            printf("%d ", ++a);
    printf("%d ", a+10);

    getch();
}
```

**A. 1 12**

B. 2 12

C. 2 11

D. 1 11

**Câu 34: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```

int main()
{
    int i = 0;
    while (i == 0)
        printf("True\n");
    printf("False\n");
    getch();
}

```

- A. True (1 lần)
- B. False (1 lần)
- C. True (vô hạn lần)**
- D. False (vô hạn lần)

**Câu 35: Phát biểu nào sau đây là sai?**

- A. Cấu trúc lặp bao giờ cũng có điều kiện để vòng lặp kết thúc.
- B. Cấu trúc lặp có số lần lặp luôn được xác định trước.**
- C. Cấu trúc lặp có hai loại là lặp với số lần biết trước và lặp với số lần không biết trước.
- D. Cấu trúc lặp có loại kiểm tra điều kiện trước và loại kiểm tra điều kiện sau.

**Câu 36: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```

void main()
{
    int x;
    if(x = 0)
        printf ("Value of x is 0");
    else
        printf ("Value of x is not 0");
    getch();
}

```

- A. Value of x is 0
- B. Value of x is not 0**
- C. Lỗi
- D. A,B,C đều sai.

**Câu 37:** Cho bài toán tính tổng  $s=1+2+3+\dots+n$ . Để giải bài toán trên ta có thể dùng?

- A. Cấu trúc rẽ nhánh.
- B. Cấu trúc lặp.**
- C. Hàm sqrt()
- D. Cả 3 đáp án trên

**Câu 38:** Vòng lặp với số lần không biết trước kết thúc khi?

- A. <Điều kiện> sai.
- B. <Điều kiện> đúng.
- C. <Điều kiện> lớn hơn 0.
- D. <Điều kiện> bằng 0.

**Câu 39:** Trong quá trình thực hiện thuật toán, khi nào cần dùng cấu trúc rẽ nhánh?

- A. Khi phải dựa trên một điều kiện cụ thể nào đó để xác định bước thực hiện tiếp theo.
- B. Khi có các phép tính toán.
- C. Khi lặp đi lặp lại một công việc nào đó.
- D. Khi sử dụng các hàm toán học.

**Câu 40: Giá trị cuối cùng của x khi đoạn mã sau được chạy ?**

```
int x;  
for(x = 0; x < 10; x++) { }
```

**A. 10**

- B. 9
- C. 0
- D. 1

**Câu 41: Khi nào khối lệnh sau while( $x < 100$ ) được thực thi?**

- A.  $x < 100$**
- B.  $x > 100$
- C.  $x=100$
- D. Không đáp án nào đúng

**Câu 42 : <Điều kiện> trong câu lệnh lặp với số lần không biết trước là?**

- A. Hàm toán học.
- B. Biểu thức logic.**
- C. Biểu thức quan hệ.
- D. Biểu thức tính toán.

**Câu 43: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
```

```
{  
    int a = 15, b = 10, c = 5;  
    if(a > b > c)  
        printf("True");  
    else  
        printf("False");  
    getch();  
}
```

- A. True
- B. False**
- C. Lỗi
- D.

**Câu 44: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()  
{  
    if ((1 || 0) && (0 || 1))  
    {  
        printf("ABC");  
    }  
    else  
    {  
        printf("DEF");  
    }  
    getch();  
}
```

- A. ABC**
- B. DEF
- C. Lỗi

**Câu 45: Lệnh nào không phải cấu trúc vòng lặp?**

- A. for
- B. do while
- C. while
- D. repeat until**

**Câu 46: Lệnh trong vòng lặp do..while được thực hiện ít nhất?**

- A. 0 lần
- B. 1 lần**
- C. 2 lần

D. vô hạn

**Câu 47: Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
    int x = 0;
    switch(x)
    {
        case 1: printf( "One" );
        case 0: printf( "Zero" );
        case 2: printf( "Hello World" );
    }
}
```

- A. One
- B. Zero
- C. Hello World
- D. ZeroHello World**

**Câu 48: Phát biểu nào đúng cho hàm sau?**

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

- A. Hàm tính mũ n của 1 số nguyên
- B. Hàm tính giai thừa của một số nguyên**
- C. Hàm tính căn bậc 2 của 1 số nguyên
- D. Không đáp án nào đúng

**Câu 49 : Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
void main()
{
    double k = 0;
    for (k = 0.0; k < 3.0; k++);
    printf("%lf", k);
    getch();
```

}

A. 012

B. 3

C. 2

D. Lỗi

**Câu 50 : Kết quả chương trình sau khi thực hiện chương trình sau ?**

```
#include <stdio.h>
#include "conio.h"
void main()
{
    int i = 3;
    switch (i)
    {
        case 0+1: printf("A");
                    break;
        case 1+2: printf("B");
                    break;
        default: printf("ABC");
    }
    getch();
}
```

A. A

B. B

C. ABC

D. Cả A,B,C đều sai

### CÂU HỎI TRẮC NGHIỆM PHẦN HÀM

**Câu 1.** Phát biểu nào sau đây là đúng khi nói về chương trình con?

A. Chương trình con là một lệnh mô tả một thao tác nhất định và có thể được thực hiện (được gọi) từ nhiều vị trí trong chương trình.

B. Chương trình con là một dãy lệnh mô tả một số thao tác nhất định và được thực hiện (được gọi) từ 1 vị trí trong chương trình.

C. Chương trình con là một dãy lệnh mô tả một số thao tác nhất định và không thể thực hiện từ nhiều vị trí trong chương trình.

**D. Chương trình con là một dãy lệnh mô tả một số thao tác nhất định và có thể được thực hiện (được gọi) từ nhiều vị trí trong chương trình.**

Câu 2. Cấu trúc của hàm nhất thiết phải bao gồm:

A. <phần đầu>  
    <phần thân>

- B. < phần đầu>
- C. < phần thân>
- D. < phần khai báo>  
    < phần thân>

Câu 3. Tham số hình thức là:

- A. Các biến được khai báo ở < phần đầu>.
- B. Các biến được khai báo ở chương trình chính.
- C. Các biến được khai báo bắt kì trong chương trình.
- D. Tất cả các biến của chương trình.

Câu 4. Biến cục bộ là:

- A. Các biến được khai báo để dùng riêng trong thân hàm.
- B. Biến của chương trình con và chương trình chính.
- C. Các biến được khai báo cho dữ liệu vào/ra.
- D. Các biến của chương trình chính.

Câu 5. Trong lời gọi hàm sin(90), thì giá trị 90 là:

- A. Biến toàn cục.
- B. Tham số hình thức.
- C. Tham số thực sự.
- D. Biến cục bộ

Câu 6. Khẳng định nào sau đây là đúng khi nói về sin(x) :

- A. Hàm trả về giá trị sin(x).
- B. Không phải là hàm.
- C. Không phải là hàm nhưng trả về 1 giá trị nào đó.
- D. Hàm thực hiện một số thao tác nhất định nhưng không trả về giá trị nào.

Câu 7. Khẳng định nào sau đây là sai khi nói về hàm không có kết quả (Void functions)?

- A. Hàm không có kết quả là hàm thực hiện một số thao tác nhất định nhưng không trả về giá trị nào.
- B. Hàm không có kết quả là loại hàm thực hiện một số thao tác nào đó và trả về một số giá trị theo sau lệnh return.
- C. setw() là hàm không có kết quả (Void functions)
- D. setprecision() là hàm không có kết quả (Void functions)

Câu 8. Trong các hàm sau đây, hàm nào là hàm không có kết quả (Void functions)?

- A. sin(x)
- B. abs(x)
- C. sqrt(x)
- D. setw()

Câu 9. Cho đoạn chương trình con sau:

```
double Luythua(double x, int k)
{
    double lt = 1.0;
    for(int i = 1; i < k; i++) lt *= x;
    return lt;
}
```

Biến x và k được gọi là:

- A. Tham số hình thức.
- B. Tham số thực sự.

C. Biến toàn cục.

D. Biến địa phương.

**Câu 10.** Biến toàn cục là:

A. Các biến được khai báo để dùng riêng trong thân hàm.

B. Biến được khai báo trong chương trình con.

C. Các biến được khai báo cho dữ liệu vào/ra.

**D. Các biến của chương trình chính.**

Câu 11 Cho đoạn chương trình sau:

Biến x, y trong hàm main được gọi là:

```
double Luythua(double x, int k){  
    double lt = 1.0;  
    for(int i = 1; i < k; i++) lt *= x;  
    return lt;  
}  
int main()  
{  
    scanf("%d %d", &x,&y);  
    printf ("%f", Luythua(x,y));  
    return 0;  
}
```

A. Tham số hình thức.

B. Biến toàn cục.

**C. Tham số thực sự.**

D. Biến cục bộ

Câu 12.

```
#include<stdio.h>  
void change(int num) {  
    num = num + 1;  
}  
  
int main() {  
    int x = 100;  
    printf("Truoc khi goi phuong thuc x = %d \n", x);  
    change(x); // truyen tham tri vao phuong thuc  
    printf("Sau khi goi phuong thuc x = %d \n", x);  
    return 0;  
}
```

Kết quả đúng là:

**A. Truoc khi goi phuong thuc x = 100**

**Sau khi goi phuong thuc x = 100**

B. Truoc khi goi phuong thuc x = 100

Sau khi goi phuong thuc x = 101

C. Truoc khi goi phuong thuc x = 101

Sau khi goi phuong thuc x = 101

D. Truoc khi goi phuong thuc x = 101

Sau khi goi phuong thuc x = 102

Câu 13.

```
#include<stdio.h>

void change(int *num) {
    *num = *num + 1;
}

int main() {
    int x = 100;
    printf("Truoc khi goi phuong thuc x = %d \n", x);
    change(&x); // truyen tham chieu vao phuong thuc
    printf("Sau khi goi phuong thuc x = %d \n", x);
    return 0;
}
```

Kết quả chạy chương trình đúng là:

A. Truoc khi goi phuong thuc x = 100

Sau khi goi phuong thuc x = 101

B. Truoc khi goi phuong thuc x = 101

Sau khi goi phuong thuc x = 101

B. Truoc khi goi phuong thuc x = 101

Sau khi goi phuong thuc x = 102

B. Truoc khi goi phuong thuc x = 100

Sau khi goi phuong thuc x = 102

Câu 14.

```
#include <stdio.h>
void incValue(int x)
{
    x++;
    printf("Bien trong ham = %d \n", x);
}
int main ()
{
    int a = 10;
    incValue(a);
    printf("Bien ngoai ham ham = %d \n", a);
    return 0;
}
```

Kết quả của chương trình là:

A. Bien trong ham = 11

Bien ngoai ham ham = 10

B. Bien trong ham = 11

Bien ngoai ham ham = 11

C. Bien trong ham = 11

Bien ngoai ham ham = 12

D. Bien trong ham = 10

Bien ngoai ham ham = 10

**Câu 15.** Trong lời gọi hàm Giaithua(5), thì giá trị 5 là:

- A. Biến toàn cục.
- B. Tham số hình thức.
- C. Tham số thực sự.**
- D. Biến cục bộ

**Câu 16.** Trong lời gọi hàm Fibonaci(100), thì giá trị 100 là:

- A. Biến toàn cục.
- B. Tham số hình thức.
- C. Tham số thực sự.**
- D. Biến cục bộ

**Câu 17.** Khẳng định nào sau đây là đúng khi nói về  $\cos(x)$  :

- A. Hàm trả về giá trị  $\cos(x)$ .**
- B. Không phải là hàm.
- C. Không phải là hàm nhưng trả về 1 giá trị nào đó.
- D. Hàm thực hiện một số thao tác nhất định nhưng không trả về giá trị nào.

**Câu 18.** Khẳng định nào sau đây là đúng khi nói về  $\text{giaithua}(x)$  :

- A. Hàm trả về giá trị  $\text{giaithua}(x)$ .**
- B. Không phải là hàm.
- C. Không phải là hàm nhưng trả về 1 giá trị nào đó.
- D. Hàm thực hiện một số thao tác nhất định nhưng không trả về giá trị nào.

Câu 19.

```
#include <stdio.h>
void changeValue(int *contro)
{
    *contro = 20;
}

int main ()
{
    int a = 10;
    changeValue(&a);
    printf("gia tri cua a: %d\n", a );
    return 0;
}
```

Kết quả của chương trình là:

- A. giá trị cua a: 20**
- B. giá tri cua a: 10
- C. giá tri cua a: 21
- D. giá tri cua a: 21

Câu 20.

```
#include <stdio.h>
void changeValue(int *contro)
{
    *contro = 20;
}
```

```
int main()
{
    int a = 21;
    changeValue(&a);
    printf("gia tri cua a: %d\n", a );
    return 0;
}
```

Kết quả của chương trình là:

- A. **gia tri cua a: 20**
- B. gia tri cua a: 10
- C. gia tri cua a: 21
- D. gia tri cua a: 21

## CON TRỎ, MÃNG, XÂU

1. Lựa chọn kết quả của đoạn chương trình sau:

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

void myfunc(char** param)
{
    ++param;
}
void main()
{
    char* string = (char*)malloc(64);
    strcpy(string, "hello_World");
    myfunc(&string);
    myfunc(&string);
    printf("%s\n", string);
    getch();
}
```

- A. **hello\_World**
- B. ello\_World
- C. llo\_World
- D. lo\_World

2. Lựa chọn kết quả của đoạn chương trình sau:

```
void myfunc(char** param)
{
    ++*param;
```

```
}
```

```
void main()
```

```
{
```

```
    char* string = (char*)malloc(64);
```

```
    strcpy(string, "hello_World");
```

```
    myfunc(&string);
```

```
    myfunc(&string);
```

```
    printf("%s\n", string);
```

```
    getch();
```

```
}
```

- A. hello\_World
- B. ello\_World
- C. llo\_World
- D. lo\_World

3. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
```

```
{
```

```
    int ints[] = { 0, 1, 2, 3 };
```

```
    int* i1 = ints + 1;
```

```
    int a = ++*i1;
```

```
    int b = a + *i1;
```

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

```
    getch();
```

```
}
```

- A. 4
- B. 3
- C. 6
- D. 5

4. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
```

```
{
```

```
    int ints[] = { 0, 5, 10, 15 };
```

```
    int* i2 = ints + 2;
```

```
    int a = *i2++; // a = *(i2++);
```

```
    printf("%d#%d\n", a, *i2);
```

```
    getch();
```

```
}
```

- A. 10#15  
B. 10#10  
C. 15#15  
D. 11#15

5. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
{
    int ints[] = { 0, 1, 2, 3 };
    int* i1 = ints + 1;
    int* i2 = ints + 2;
    int a = ++*i1 + *i2++;
    int b = *++i1 + *i2--;
    printf("%d#%d", a, b);
    getch();
}
```

- A. 4#4  
**B. 4#5**  
C. 5#6  
D. 4#6

6. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
{
    int i = 400;
    int *ptr = &i;
    *++ptr = 2;
    printf("%d %d", i, *ptr);
    getch();
}
```

- A. 400 2**  
B. 400 400  
C. 400 401  
D. Complier error

7. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
{
    char str[] = {"pvpit"};
    char *s1 = str;
    s1++;
```

```
    printf("%c", *s1);
    getch();
}
```

- A. pvpit
- B. vpit
- C. v
- D. vp

8. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
{
    char *s = "\12345s\n";
    printf("%d", strlen(s));
    printf("\n%s", s);
    getch();
}
```

- A. 5
- B. 7
- C. 9
- D. 10

9. Lựa chọn kết quả của đoạn chương trình sau:

```
int main(int argc, char** argv)
{
    const char* foo = "wow"; // line 1
    foo = "top"; // line 2
    foo[0] = 1; // line 3
    return 0;
}
```

- A. 1
- B. 2
- C. 3
- D. 4

10. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()
```

```
{  
    int x = 5,y = 6;  
    int* const p = &x;  
    p = &y;  
    printf("%d", (*p));  
    getch();  
}
```

**A. Lỗi biên dịch**

- B. 6
- C. 5
- D. 11

11. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()  
{  
    int x = 5, y = 8;  
    const int* p;  
    p = &x;  
    p = &y;  
    x++;  
    printf("%d", *p);  
    getch();  
}
```

- A. 5
- B. 6
- C. 8**
- D. Lỗi biên dịch

12. Lựa chọn kết quả của đoạn chương trình sau:

```
void main()  
{  
    int x = 5;  
    const int* p;  
    p = &x;  
    x++;  
    *p = 4;  
    printf("%d", *p);  
    getch();  
}
```

- A. 5
- B. 6**

C. 4

#### D. Lỗi biên dịch

13. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>

int main()
{
    int a = 320;
    char *ptr;
    ptr = (char*)&a;
    printf("%d ", *ptr);
    return 0;
}
```

A. 320

**B. 64**

C. Lỗi biên dịch

D. 0

14. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>

int main()
{
    int i = 3;
    int *j;
    int **k;
    j = &i;
    k = &j;
    printf("%u , %u , %d ", k, *k, **k);
    return 0;
}
```

**A. Địa chỉ của j, địa chỉ của biến i, 3**

B. Lỗi biên dịch

C. 3 , 3 , 3

D. 0, 0, 3

17. Lựa chọn kết quả của đoạn chương trình sau:

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

```
#include <conio.h>

int main()
{
    int a = 5, b = 10, c;
    int *p = &a, *q = &b;
    c = p - q;
    printf("%d", c);
    getch();
}
```

- A. 3  
B. 4  
C. 6  
D. Lỗi biên dịch

20. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int array[2][2][3]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
    printf("%d", array[1][0][2]);
    getch();
}
```

- A. 6  
B. 7  
C. 8  
D. 9

21. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char arr[8]={'V','I','E','T','N','A','M'};
    char *p;
    p=(char *)arr+2;
    printf("%c", p);
    getch();
}
```

- A. I
- B. E
- C. M

21. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char arr[8]={'V','I','E','T','N','A','M'};
    char *p;
    p=(char *)(arr+2)[2];
    printf("%c", p);
    getch();
}
```

- A. I
- B. E
- C. M
- D. N

26. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[3][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
    printf("%u, %u, %u\n", a[0]+1, *(a[0]+1), *(*(a+0)+1));
    getch();
}
```

- A. 1006, 2, 2
- B. 1006, 4, 4
- C. 1002, 5, 5
- D. Lỗi biên dịch

28. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char str[] = "VNCODING\0.NET\0";
    printf("%s\n", str);
    getch();
```

}

- A. VNCODING  
B. VNCODING\0\.NET\0  
C. VNCODING\0\.NET  
D. Lỗi biên dịch

29. Lựa chọn kết quả của đoạn chương trình sau:

```
#include <stdio.h>
#include <conio.h>
void swap(char *, char *);

int main()
{
    char *pstr[2] = {"VNCODING", ".NET"};
    swap(pstr[0], pstr[1]);
    printf("%s%s", pstr[0], pstr[1]);
    getch();
}
void swap(char *t1, char *t2)
{
    char *t;
    t=t1;
    t1=t2;
    t2=t;
}
```

- A. VNCODING.NET  
B. NETVNCODING  
C. Giá trị pstr[0]  
D. Giá trị pstr[1]

## KIẾU DỮ LIỆU TỆP

### Câu 1

Hàm nào dưới đây chỉ dùng để đọc 1 ký tự từ tệp

- A. getch();  
B. fscanf();  
**C. fgetc();**  
D. scanf();

### Câu 2

Hàm nào dưới đây chỉ dùng để ghi một khối dữ liệu vào file:

- A. fgets();
- B. fputs();
- C. fwrite();
- D. fread();

### **Câu 3**

Giả sử tệp DATA chứa 5 byte: “70, 26, 13, 10, 44”. Cho đoạn lệnh. Hãy cho biết kết quả của đoạn chương trình trên:

```
FILE * f;  
char ch;  
f = fopen("DATA", "r"); while (!feof(f)) {  
    ch = fgetc(f);  
    printf("% c", ch);  
}  
fclose(f);
```

- A. “70, 26, 13, 13, 10, 44”
- B. **“70, 26, 13, 10, 44”**
- C. “70, 26, 10, 44”
- D. Đoạn lệnh có lỗi

### **CÂU 4:**

Cho tệp nhị phân DATA chứa dãy các cấu trúc liền nhau L struct T{...} x; và f là con trỏ tệp đến DATA. Chọn câu đúng nhất trong các câu sau:

- A. Lệnh fseek(f, 10, SEEK\_END) định vị con trỏ tệp đến byte thứ 10 trong tệp.
- B. Đoạn lệnh fseek(f, 10, SEEK\_END); định vị con trỏ tới cuối tệp.
- C. Giả sử tệp có n cấu trúc, lệnh fseek(f, n, SEEK\_SET) sẽ đặt con trỏ tới cấu trúc cuối cùng của tệp.**
- D. Giả sử tệp có n cấu trúc, lệnh fseek(f, n, SEEK\_CUR); sẽ không làm thay đổi vị trí con trỏ của tệp.

### **CÂU 5:**

f là con trỏ tệp nhị phân DATA chứa không quá 100 bản ghi (kiểu cấu trúc) liền nhau. Cho khai báo. Đoạn chương trình nào sau đây đọc toàn bộ các bảng ghi trong DATA vào x: struct T {...}; struct T x[100]; int i = 0, n = 0;

- A. fread(x+i++, sizeof(T), 1, f); while (!feof(f)) fread(x+i++, sizeof(T), 1, f); fclose(f);
- B. fseek(f, 0, SEEK\_END); n=f.tell(f)/sizeof(T); fseek(f, 0, SEEK\_SET); fread(x, sizeof(T), n, f); fclose(f);
- C. A và B đúng**
- D. A và B sai

### **CÂU 6:**

Phát biểu đúng nhất về mẫu lệnh( f là con trỏ tệp): while ((ch = fgetc(f)) != EOF)...

- A. Chọn 1 phần tử và kiểm tra xem có phải là EOF hay không.
- B. Đọc các phần tử của tệp chừng nào kí tự đó chưa phải là kí tự xuống dòng
- C. Ghi một phần tử và kiểm tra xem có phải là EOF hay không.
- D. Đọc các phần tử của tệp chừng nào kí tự đó chưa phải là kí tự kết thúc.**

#### CÂU 7:

Dấu hiệu nào là đúng khi sử dụng hàm fread();

- A.int fread( void \*ptr, int size, FILE \*f, int n);
- B.int fread( FILE \*f, void \*ptr, int size, int n);
- C.int fread( int size, void \*ptr, int n, FILE \*f);
- D. int fread( void \*ptr, int size, int n, FILE \*f);**

#### CÂU 8:

Dấu hiệu nào là đúng khi sử dụng hàm fputs();

- A. int puts(const char \*s, FILE \*f);**
- B. int puts( const char \*s);
- C. int puts(FILE \*f, const char \*s);
- D. int puts(FILE \*f);

#### CÂU 9:

Dấu hiệu nào là đúng khi sử dụng hàm fopen() để mở một file đã tồn tại đọc theo kiểu nhị phân:

- A. f=fopen("du\_lieu","r");
- B. f=fopen("du\_lieu","r+b");
- C. f=fopen("du\_lieu","a+t");**
- D. f=fopen("du\_lieu","a+b");

#### CÂU 10:

Hàm nào trả lại kích thước của một tệp được mở:

- A. long filelength(int the\_file);**
- B. double filelength(int the\_file);
- C. int filelength(int the\_file);
- D. Không có đáp án nào đúng.

#### CÂU 11:

Câu lệnh sau có ý nghĩa gì? int open(tep1, O\_TRUNC);

- A. Mở tệp nhị phân để ghi.
- B. Xóa nội dung của tệp.**
- C. Mở tệp văn bản để đọc và ghi.
- D. Tất cả các đáp án trên.

### CÂU 12:

Câu lệnh sau có ý nghĩa gì: FILE \* fopen(tep1, "ab");

- A. Mở tệp nhị phân để ghi.
- B. Mở tệp nhị phân đã có và ghi thêm dữ liệu nối tiếp vào tệp này.**
- C. Mở tệp nhị phân để ghi mới.
- D. Mở tệp nhị phân để đọc.

### CÂU 13:

Hàm nào có thể thay đổi kích thước của tệp đang mở:

- A. double chsize(int handle, long size);
- B. long chsize(int handle, long size);
- C. int chsize(int handle, long size);**
- D. Tất cả các đáp án trên

### CÂU 14:

FILE \* fopen(tep1, "r + ");[/CODE]

Câu lệnh sau có ý nghĩa gì?

- A. Mở tệp văn bản cho phép ghi.
- B. Mở tệp văn bản cho phép đọc.
- C. Mở tệp văn bản cho phép cả đọc cả ghi.**
- D. Mở tệp văn bản đã tồn tại để đọc.

### CÂU 15:

Trong chế độ văn bản, dấu hiệu kết thúc tệp là:

- A. EOF**
- B. NULL
- C. \n
- D. Ctrl\_Z

### CÂU 16:

FILE \* fopen(tep1, "ab");[/CODE]

Câu lệnh sau có ý nghĩa gì:

- A. Mở tệp nhị phân để ghi.
- B. Mở tệp nhị phân đã có và ghi thêm dữ liệu nối tiếp vào tệp này.**
- C. Mở tệp nhị phân để ghi mới.
- D. Mở tệp nhị phân để đọc.

### Câu 17

Hãy chọn câu đúng

```
FILE * f = fopen("FL.txt", "r");
```

```
int n = 7;
```

```
fprintf(f, "%d", n);
```

**A. Đoạn code gây lỗi**

B. Đoạn code không lỗi

C. Đoạn code này sẽ ghi trị 7 lên file "FL.txt"

D. Đoạn code này sẽ đọc một trị từ file "FL.txt" vào biến n

### CÂU 18:

Giả sử f là con trỏ tệp nhị phân. Để đọc từ f cho biến x kiểu int, ta có thể dùng:

A. fscanf(f,"%d",x);

B. fread(&x,sizeof(int),1,f);

C. getw(f);

**D. B và C đúng**

### CÂU 19:

Cho khai báo FILE \*f1,\*f2; int c; và đoạn lệnh. Trong trường hợp tổng

quát, hãy cho biết các khẳng định sau, khẳng định nào đúng:

```
f1 = fopen("source", "rt");
```

```
f2 = fopen("TARGET", "wt"); while ((c = fgetc(f1)) != EOF) fputc(c, f2);
```

A. Độ dài file "source" luôn bé hơn độ dài file "TARGET".

**B. Độ dài file "source" bằng độ dài file "TARGET".**

C. Độ dài file "source" nói chung lớn hơn độ dài file "TARGET".

D. Độ dài file "source" nói chung sẽ lớn hơn độ dài file "TARGET" 1 byte.

### Câu 20

Hàm nào Mở file "time.txt" để đọc

**A. fopen("time.txt","r");**

B. open("time.txt","read");

C. open("time.txt");

D. fopen("time.txt","r+");

### CÂU 21:

Khai báo hàm đọc một text file vào một mảng các số nguyên, hãy chọn khai báo đúng

A. void Read( char\* fName, int a[]);

B. void Read(char\* fName, int a);

C. void Read(char\* fName, int \*a);

**D. void Read(char\* fName, int \*&a, int &n);**

## PHẦN 2 STRUCT

### CÂU 1:

Kết quả của chương trình sau là gì:

```
#include <stdio.h>
void main() {
    struct diem {
        float k;
        float a;
        float l;
    };
    struct diem m; m.k = 8; m.a = 6.5; m.l = 6;
    printf(" % .1f % .1f % .1f", m.k, m.a, m.l);
}
```

- A. “**8.06.56.0**”
- B. “86.56”.
- C. “8.0000006.5000006.000000”
- D. “86.500006”

### CÂU 2:

Toán tử nào sau đây sẽ truy xuất 1 thành phần của con trỏ cấu trúc:

- A. “.”(Toán tử chấm)
- B. “->”(Toán tử mũi tên)
- C. **Cả hai đều đúng.**
- D. Cả hai đều sai.

### CÂU 3:

Cho khai báo sau. Số lượng bit sử dụng trong biến cấu trúc trên là bao nhiêu:

```
struct Date {
    unsigned int ngay: 5;
    unsigned int thang;
    unsigned int nam: 11;
}
```

sn1;

A.16

B.20

**C.32**

D.48

### CÂU 4:

Chọn câu đúng:

- A. “struct” là một kiểu dữ liệu do người dùng định nghĩa bao gồm nhiều thành phần có kiểu khác nhau.
- B. “struct” là sự kết hợp của nhiều thành phần có thể có thể có kiểu khác nhau
- C. Cả 2 ý đều đúng.**
- D. Cả 2 ý đều sai.

#### CÂU 5:

Có thể truy cập thành phần của cấu trúc thông qua con trỏ như (với p là con trỏ cấu trúc, a là thành phần cấu trúc):

- A. (\*p).a;**
- B. \*p->a;
- C. A và B đều đúng
- D. A và B đều sai

#### CÂU 6:

```
struct T {  
    int x;  
    float y;  
}  
t, * p, a[10];[/CODE]
```

Cho khai báo. Câu lệnh nào không hợp lệ?

- A. p=&t;
- B. p=&t.x;
- C. p=&a[5];
- D. p=&a;**

#### CÂU 7:

```
struct ngay {  
    int ng, th, n;  
}  
vaotruong, ratruong;  
typedef struct sinhvien {  
    char hoten;  
    ngay ngaysinh;  
};[/CODE]
```

Cho các khai báo sau. Hãy chọn câu đúng nhất:

- A. Không được phép gán: vaotruong = ratruong;

**B. “sinhvien” là tên cấu trúc, “vaotruong”, “ratruong” là tên biến cấu trúc.**

C. Có thể viết “vaotruong.ng”, “ratruong.th”, “sinhvien.vaotruong.n” để truy xuất đến các thành phần tương ứng.

D. A, B, C đều đúng

**CÂU 8:**

```
struct S1 {  
    int ngay, thang, nam;  
};  
S1 = (2, 1, 3);  
struct S2 {  
    char hoten[10];  
    struct S1 ngaysinh;  
};  
S2 = {  
    “  
    Ly Ly”,  
    {  
        4,  
        5,  
        6  
    }  
};[/CODE]
```

Trong các khởi tạo giá trị cho các cấu trúc sau, hãy chọn câu đúng:

A. S1 đúng

**B. S2 đúng**

C. S1 và S2 đều đúng

D. S1 và S2 đều sai

**CÂU 9:**

Đối với kiểu có cấu trúc, cách gán nào sau đây là không được phép:

A. Gán biến cho nhau.

B. Gán hai phần tử mảng( kiểu cấu trúc) cho nhau.

C. Gán một phần tử mảng( kiểu cấu trúc) cho một biến hoặc ngược lại.

**D. Gán hai mảng có cấu trúc có cùng số phần tử cho nhau.**

**CÂU 10:**

1. Sinh viên 1:

```
struct SV {  
    char ht[25];  
    int tuoi;  
    struct Sv * tiep;  
};[/CODE]
```

2. Sinh viên 2

```
typedef  
struct SV node;  
struct SV {  
    char ht[25];  
    int tuoi;  
    node * tiep;  
};[/CODE]
```

3. Sinh viên 3

```
typedef  
struct SV {  
    char ht[25];  
    int tuoi;  
    struct SV * tiep;  
}  
node;[/CODE]
```

Để tạo danh sách liên kết, theo bạn sinh viên nào dưới đây là khai báo đúng cấu trúc tự trả sẽ được dùng:

- A. 1
- B. 2
- C. 2, 3
- D. 1,2 và 3**

### CÂU 11:

```
struct {  
    int to;  
    float soluong;  
}
```

x[10];

for (int i = 0; i < 10; i++) scanf(" % d % f", &x.to, &x.soluong);[/CODE]

Cho đoạn chương trình sau. Chọn câu đúng nhất trong các câu sau:

- A. Đoạn chương trình trên có lỗi cú pháp.
- B. Không được phép sử dụng toán tử lấy địa chỉ đối với các thành phần “to” và “soluong”.
- C. Lấy địa chỉ thành phần “soluong” dẫn đến chương trình hoạt động không đúng dẫn.**
- D. Cả 3 ý trên đều sai.

### CÂU 12:

1. Khai báo 1:

```
struct SV {  
    thongtin;  
    struct SV * tiep;  
};[/CODE]
```

2. Khai báo 2:

```
struct SV {  
    thongtin  
};  
struct DS {  
    struct SV * sv;  
    struct DS * tiep;  
};[/CODE]
```

Để dùng danh sách liên kết, xét hai khai báo sau(cần 1KB để lưu dữ thông tin về một sinh viên).

(Với “thongtin” là một thành phần dữ liệu của cấu trúc); Chọn câu đúng nhất trong các câu sau:

- A. Khai báo 1 tốn nhiều bộ nhớ hơn khai báo 2.
- B. Khai báo 2 sẽ giúp chương trình chạy nhanh hơn khi đổi vị trí 2 sinh viên.**
- C. Khai báo 1 sẽ giúp tiết kiệm câu lệnh hơn khi viết hàm đổi vị trí 2 sinh viên.
- D. Khai báo 2 sẽ giúp chương trình chạy nhanh hơn khi duyệt danh sách.

### CÂU 13:

Chọn câu đúng:

- A. “struct” là một kiểu dữ liệu do người dùng định nghĩa bao gồm nhiều thành phần có kiểu khác nhau.
- B. “struct” là sự kết hợp của nhiều thành phần có thể có thể có kiểu khác nhau.

**C. Cả 2 ý đều đúng.**

D. Cả 2 ý đều sai.

**CÂU 14:**

```
typedef struct ST {
```

```
    int d1, d2, d3;
```

```
};
```

```
ST v = {
```

```
    5,
```

```
    6,
```

```
    7
```

```
};
```

```
ST * p = & v;
```

```
p.d1++;[/CODE]
```

Xem đoạn code sau. Giá trị của biến v là:

A. {6, 6, 7};

B. {5, 6, 7};

C. Nhóm trị khác

**D. Đoạn code gây lỗi**

**CÂU 15:**

Toán tử nào sau đây sẽ truy xuất 1 thành phần của con trỏ cấu trúc:

A. ":"(Toán tử chấm)

B. "->"(Toán tử mũi tên)

**C. Cả hai đều đúng.**

D. Cả hai đều sai.

**CÂU 16:**

```
typedef struct STUDENT {
```

```
    int d1, d2, d3;
```

```
};
```

```
STUDENT v = {
```

```
    2,
```

```
    3,
```

```
    4
```

```
;}
```

v.d1++;[/CODE]

Xem đoạn code sau. Giá trị trung bình của các trường trong biến v là gì:

A. 3.0

B. 4.0

**C. Giá trị khác**

D. Có lỗi trong đoạn code

**CÂU 17:**

Hãy khai báo một cấu trúc mô tả sinh viên, thông tin về mỗi sinh viên bao gồm: tên, điểm 3 môn học. Khai báo nào sau đây là đúng:

A. structure STUDENT {char Name[]; int s1,s2,s3;};

B. struct STUDENT {char Name[]; int s1, s2, s3;};

**C. typedef struct STUDENT {char Name[]; float s1,s2,s3;};**

D. typedef STUDENT { char Name[]; int s1, s2,s3;};

**CÂU 18:**

Kiểu dữ liệu nào có thể chứa nhiều thành phần dữ liệu có thể có kiểu dữ liệu khác nhau thành một nhóm duy nhất:

A. Mảng

B. Con trỏ

C. Tập tin

**D. Cấu trúc(struct)**

**CÂU 19:**

```
#include <stdio.h>
void main() {
    clrscr();
    struct sv {
        float d;
        char ht[10];
    };
    struct sv m, * p;
    p = & m;
    (* p).d = p -> d = 10;
    strcpy(m.ht, "NguyenVanTuan");
    printf(" % 0.1 f", m.d);
```

```
printf(" % s", m.ht);
};[/CODE]
```

Chọn kết quả đúng cho chương trình sau:

- A. “10.000000NguyenVanTuan”
- B. “10.0NguyenVanTuan”
- C. Chương trình bị lỗi

**D. Kết quả khác**

**CÂU 20:**

```
void main() {
    struct sv {
        float d;
        char ht[10];
    };
    struct sv m, * p;
    p = & m;
    printf(" % p", & m);
    printf(" % p", p);
};[/CODE]
```

Chọn kết quả hợp lý cho chương trình sau:

- A. “FFE6FFE6”
- B. “FFE6FFE7”
- C. “FFE66EFF”
- D. “FFE4FFE4”**

**Câu 21**

```
#include <stdio.h>
#include <string.h>
void main() {
    struct S1 {
        float d;
        float d1;
        float d2;
    };
    struct S1 m = {
```

```
12,  
1  
};  
printf(" % 0.1 f % 0.1 f % 0.1 f", m.d, m.d1, m.d2);  
};[/CODE]
```

Chọn kết quả đúng cho chương trình sau:

- A. **“12.01.00.0”**
- B. “12.0000001.0000000.000000”
- C. Kết quả khác
- D. Chương trình bị lỗi

### Câu 22

Có bao nhiêu cách khai báo biến cấu trúc?

- A. 1
- B. 2**
- C. 3
- D. 4

### Câu 23

Khi sử dụng từ khóa typedef trước định nghĩa cấu trúc thì:

- A. Khai báo biến cho cấu trúc đó ta không cần sử dụng từ khóa “struct” nữa.**
- B. Khai báo 1 biến cho loại cấu trúc đó ta cần sử dụng từ khóa “struct”.
- C. Không thể khai báo thêm biến cấu trúc nào nữa.

### Câu 24

Trong các khai báo sau, khai báo nào không đúng:

- A. struct Date{int ngay, thang, nam;};
- B. struct { int ngay, thang, nam;} D1,D2;
- C. typedef struct { int ngay, thang, nam;} Date;

#### D. struct Date

```
{  
long int ngay:7;  
long int thang:6;  
long int nam:5;  
}
```

### Câu 25

```
struct Employee {  
char Code[], name[];  
long Salary;  
};
```

```
Employee e1 = {  
    “  
    E089”,  
    “Hoang so”,  
    12000  
}, e2 = e1;  
printf(“ % ld”, el.Salary + e2 -> Salary);[/CODE]
```

Đâu là kết quả của đoạn mã sau:

A. 24000

B. 12000

**C. Đoạn mã bị lỗi**

D. Kết quả khác