



# 200 BÀI TẬP TRẮC NGHIỆM LẬP TRÌNH C



[www.28tech.com.vn](http://www.28tech.com.vn)

## NỘI DUNG

I.Kiểu dữ liệu, toán tử và cấu trúc rẽ nhánh .....	3
II.VÒNG LẶP .....	15
III. HÀM.....	31
IV.ĐỘ PHỨC TẠP CỦA THUẬT TOÁN .....	43
V.ĐỀ QUY .....	47
VI. MẢNG.....	54
VII.CON TRỎ .....	59
VIII. DANH SÁCH LIÊN KẾT.....	74
IX. CÂY NHỊ PHÂN .....	96

## I.KIỂU DỮ LIỆU, TOÁN TỬ VÀ CẤU TRÚC RỄ NHÁNH

1.Trong ngôn ngữ lập trình C, phát biểu nào dưới đây về kiểu dữ liệu là chính xác

A.Kiểu dữ liệu int được dùng để lưu số nguyên, có kích thước là 4byte, có thể lưu được giá trị từ  $-2^{32}$  tới  $2^{32}$ .

B. Kiểu dữ liệu unsigned int được dùng để lưu số nguyên có dấu, có kích thước 4 byte và lưu được giá trị từ  $-2^{31}$  tới  $2^{31}$

C. Kiểu dữ liệu long long được dùng để lưu số nguyên có dấu, có kích thước 8byte, có thể lưu được giá trị từ  $-2^{63}$  tới  $2^{63} - 1$

D. Kiểu dữ liệu unsigned long long có thể lưu được bất cứ giá trị nào mà kiểu dữ liệu int có thể lưu

2.Đặc tả “%x” và đặc tả “%o” trong ngôn ngữ lập trình C được dùng để in ra số nguyên dưới dạng nào?

A. Hệ nhị phân và hệ bát phân

B. Hệ bát phân và hệ thập lục phân

C. Hệ thập lục phân và bát phân

D. Hệ nhị phân và thập lục phân

3.Kết quả của câu lệnh printf(“%d”, ‘A’); trong ngôn ngữ lập trình C sẽ cho kết quả gì?

A. A

B. 65

C. d

D. Lỗi

4.Kết quả của chương trình sau là gì ?

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

int main(){
    int a = 1000000.7129;
```

```
int b = 1000000.5567;  
printf("%d", a + b);  
}
```

A. 2000001

B. 2000000

C. 2000001.2

D. Kết quả khác

5. Kết quả của chương trình sau là gì ?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int a = 100;  
    char c = "A";  
    printf("%d %d", a, c);  
}
```

A. 100 65

B. 100 A

C. 100 97

D. Lỗi

6. Kết quả của chương trình sau là gì ?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int a = 100;  
    {  
        float a = 100.5;  
        printf("%.2f ", a);  
    }  
    printf("%d", a);  
}
```

```
    return 0;  
}
```

- A. 100.50 100
- B. 100.00 100
- C. 100
- D. Lỗi

7. Kết quả của chương trình sau là gì ?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int a = 100;  
    printf("%d ", a++);  
    printf("%d", ++a);  
    return 0;  
}
```

- A. 101 101
- B. 100 101
- C. 100 102
- D. 102 102

8. Hệ thập phân, nhị phân và hệ thập lục phân nào sau đây là cùng một số?

- A. 18 00010010 12A
- B. 8987 0010001110011011 231B
- C. 1081 010000111001 4E9
- D. 897 001110000001 381

9. Phát biểu nào sau đây về quy tắc đặt tên biến trong C là sai?

- A. Trong C biến không được bắt đầu bằng chữ số

- B. Biến không được chứa dấu cách và chữ số xuất hiện ở đầu
- C. Biến không được chứa các kí tự đặc biệt
- D. Trong C 2 biến có thể đặt cùng tên miễn là nó có kiểu dữ liệu khác nhau

10. Tên biến nào sau đây trong C là hợp lệ?

- A. int 28tech;
- B. int laptrinhC#;
- C. long long int;
- D. int int\_4byte;

11. Kết quả của chương trình sau là gì ?

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

int main(){
    printf("%d %d %d %d %d", 10 != 10, 10 <= 20, 50 == 50, 20 == 100, 5^2
    == 25);
    return 0;
}
```

- A. 0 1 1 0 1
- B. 0 1 1 0 5
- C. 0 1 1 0 25
- D. 0 1 1 1 25

12. Kết quả của chương trình sau là gì ?

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

int main(){
    int a = 0, b = 0;
    {a = 2;}
    int kq = 2 * a + 3 * b;
    scanf("%d%d", &a, &b);
}
```

```
printf("%d", kq);  
return 0;  
}
```

A.0

B.Lỗi chương trình

C.Tùy thuộc vào giá trị a, b nhập từ bàn phím

D.4

13.Kết quả của chương trình sau là gì ?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int a = 5;  
    {  
        int b = 10;  
        ++b;  
        a++;  
        {  
            int a = 20;  
            ++a;  
            a = ++b;  
        }  
        ++a;  
        ++b;  
        printf("%d %d ", a, b);  
    }  
    printf("%d", a);  
    return 0;  
}
```

A.7 13 7

B.13 13 5

C.13 13 13

D.Lỗi biên dịch

14.Kết quả chương trình dưới đây là gì ?

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

int a = 10;
int main(){
    int a = 20;
    {
        int a = 30;
        printf("%d ", a);
        ++a;
    }
    printf("%d", a);
    return 0;
}
```

A.30 20

B.30 21

C.30 11

D. Lỗi biên dịch do khai báo biến trùng tên

15. Kết quả chương trình dưới đây là gì ?

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

int main(){
    int a = 10, b = 20;
    if((a < b) && (a >= 10) && (a + b)){
        printf("X\n");
    }
    int x = 100000;
    else if(x * 100000 == 10000000000){
```



```
    printf("Y\n");
}
else{
    printf("Z");
}
return 0;
}
```

A.X

B.Y

C.Z

D.Chương trình gây lỗi biên dịch

16. Kết quả của chương trình sau là gì ?

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

int main(){
    int a = 100;
    if((100 == 100) || (a++)){
        printf("%d ", a);
    }
    if((100 > 50) && (a++ > 100)){
        ++a;
        printf("%d ", a);
    }
    if(++a < 102){
        printf("%d ", ++a);
    }
    return 0;
}
```

A.101 103

B. 100

C. 101 102

D. 100 102

17. Kết quả của chương trình sau là gì ?

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

int main(){
    char c = 65;
    ++c;
    printf("%c ", c);
    char d = c + 32;
    --d;
    printf("%c", (int)d);
    return 0;
}
```

A.65 96

B.A b

C.B a

D. Lỗi biên dịch

18. Kết quả của chương trình sau là gì ?

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

int main(){
    char x = '1', y = '2', z = '3';
    int tmp = x + y + (z - '0');
    printf("%d", tmp);
    return 0;
}
```

A.6

B.102

C.101

D.Lỗi biên dịch

19. Kết quả của chương trình sau là gì ?

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

int main(){
    int n = 3;
    switch(n == 3){
        case 0 :
            printf("C++");
        case 1:
            printf("C#");
            break;
        case 2:
            printf("Java");
            break;
        case 3:
            printf("C");
        default:
            printf("Default");
    }
    return 0;
}
```

A.C Default

B.C#

C.C++ C#

D.Lỗi biên dịch

20.Kết quả của chương trình sau là gì ?

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

```

int main(){
    int a = 1, b = 2;
    if(a < b){
        a = b;
        b = a;
    }
    printf("%d %d", a, b);
    return 0;
}

```

A.1 1

B.1 2

C.2 2

D.3 1

21. Kết quả của chương trình sau là gì ?

```

#include <stdio.h>
#include <math.h>

int main(){
    int can = sqrt(1e12);
    if(can * can == 1000000000000){
        printf("YES\n");
    }
    else{
        printf("NO\n");
    }
}

```

A.YES

B.NO

C.Lỗi biên dịch

D.Lỗi runtime

22. Kết quả của chương trình sau là gì ?

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

int main(){
    int x = 10;
    int a = !(10 < x++) ? ++x : x++;
    printf("%d %d", a, x);
    return 0;
}
```

A. 10 11

B. 11 11

C. 11 12

D. 12 12

23. Phát biểu nào sau đây về kiểu dữ liệu trong C là đúng?

A. Chỉ có 2 kiểu dữ liệu lưu số nguyên trong C là int và long long, muốn lưu số thực ta phải sử dụng float hoặc double

B. %u là đặc tả số nguyên trong C

C. Một biến int có thể lưu được giá trị của 1 số thuộc kiểu long long

D. Trong C không có kiểu dữ liệu số nguyên lớn

24. Phát biểu sau về phép nhân trong C là sai?

A. Khi nhân 1 số int với 1 số int sẽ cho kết quả ở số int

B. Khi nhân 1 số thực với 1 số nguyên trong C kết quả sẽ làm tròn sang số nguyên gần với kết quả nhất

C. Nhân 2 số long long sẽ cho kết quả là 1 số long long có kết quả có thể bị sai

D. Nhân 1 số int với 1 số int không thể bị tràn dữ liệu của kiểu long long

25. Hàm getchar() trong C được sử dụng làm gì?

A.Xóa bộ đệm

B.Đọc 1 ký tự từ bàn phím

C.Đọc 1 hoặc nhiều ký tự từ bàn phím cho tới khi gặp dấu xuống dòng

D.getchar() có kiểu trả về là void

26.Phát biểu hàm toán học nào sau đây là sai?

A.Hàm pow trả về số thực ở dạng double

B.Hàm cbrt() và sqrt() là hàm tính căn bậc 3 và căn bậc 2 của 1 số nguyên có dấu hoặc không dấu

C.Muốn sử dụng các hàm toán học ta cần thêm thư viện math.h

D.Hàm fabs() được dùng để tìm trị tuyệt đối của số thực

27.Phát biểu về Ngôn ngữ lập trình C sau đây là sai?

A.Ngôn ngữ lập trình C không có hướng đối tượng (OOP)

B.Ngôn ngữ C++ là mở rộng của C có bổ sung thêm OOP và STL

C.Ngôn ngữ lập trình C++ có thể sử dụng mã nguồn của C

D.Ngôn ngữ lập trình C rất mạnh trong phát triển website.

28.Phát biểu nào về hàm scanf() trong C là sai?

A.Dấu & trước tên biến trong hàm scanf() thể hiện địa chỉ của biến

B.Hàm scanf() không có giá trị trả về

C.1 câu lệnh scanf() có thể sử dụng để vừa nhập số vừa nhập chữ

D.Phần đặc tả của hàm scanf() được đặt trong dấu nháy kép

29.Phát biểu nào về hàm làm tròn trong C là đúng?

A.ceil làm tròn lên, floor làm tròn xuống, round làm tròn dựa vào phần thập phân, 3 hàm này đều trả về số thực.

B. ceiling làm tròn lên, floor làm tròn xuống, round làm tròn dựa vào phần thập phân, 3 hàm này đều trả về số nguyên

C. ceil làm tròn lên, floor làm tròn xuống, round làm tròn dựa vào phần thập phân, hàm ceil và floor trả về số thực, round trả về số nguyên

D. Hàm ceil và floor thuộc thư viện math.h còn hàm round thuộc thư viện stdlib.h

30. Kết quả của chương trình sau là gì?

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

int main(){
    printf("%d", pow(2, 10));
}
```

A. 1024

B. 1024.000000

C. 0

D. Lỗi biên dịch

## II. VÒNG LẶP

1. Phát biểu nào sau đây về vòng lặp for là đúng?

A. Vòng lặp for chỉ được sử dụng khi biết trước số lượng vòng lặp cần thực hiện

B. Vòng lặp for không thể bị lặp vĩnh viễn

C. Nếu vòng for không có điều kiện lặp nó sẽ không thể thực hiện

D. Vòng for có thể không thực hiện bất cứ vòng lặp nào

2. Output của chương trình sau là gì ?

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

int main(){
    for(int i = 1; i <= 5; i++){
        printf("%d ", i);
    }
    printf("%d ", i);
}
```

```
    return 0;
}
```

A. 1 2 3 4 5

B. 1 2 3 4 5 6

C. 1 2 3 4 5 sau đó hiển thị lỗi

D. Lỗi biên dịch

3. Output của chương trình sau là gì ?

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

int main(){
    int i = 0;
    for(int i = 1; i <= 10; i++){
        printf("%d ", i);
        ++i;
    }
    printf("%d ", i);
    return 0;
}
```

A. 1 3 5 7 9

B. 1 3 5 7 9 11

C. 1 3 5 7 9 0

D. Lỗi biên dịch

4. Output của chương trình sau là gì ?

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

int main(){
    int i = 0;
```



```
for(int i = 1; ; i++){  
    printf("%d ", i);  
}  
printf("%d ", i);  
return 0;  
}
```

A.0

B.Vô hạn các số tăng dần từ 1

C.Lỗi biên dịch

D.Vô hạn các số 1

5. Output của chương trình sau là gì ?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    for(int i = 1; i <= 10 ; i++){  
        printf("%d ", i);  
        if(i % 3 == 0)  
            i += 2;  
    }  
    return 0;  
}
```

A. 1 2 3 4 9 11

B. 1 2 3 6 7 9

C. 1 2 3 6 9

D. 1 2 3 5 7 9

6. Output của chương trình sau là gì ?

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

```
int main(){
    for(int i = 1; i <= 10 ; i++){
        printf("%d ", i);
        if(i % 3 == 0)
            continue;
        else
            ++i;
    }
    return 0;
}
```

A. 1 2 3 4 5 7 9 10

B. 1 3 4 6 7 9 10

C. 1 3 4 6 7 9

D. 1 2 3 4 5 6 7 8 9 10

7. Ý nghĩa của câu lệnh break trong vòng lặp?

A. Nếu gặp câu lệnh break, vòng lặp sẽ thực hiện nốt các câu lệnh còn lại trong vòng lặp hiện tại sau đó kết thúc

B. Nếu gặp câu lệnh break, vòng lặp sẽ kết thúc ngay lập tức mà không thực hiện thêm bất kỳ câu lệnh nào còn lại trong vòng lặp

C. Nếu gặp câu lệnh break, vòng lặp sẽ kết thúc ngay lập tức nếu trước đó đã thực hiện được ít nhất 1 vòng lặp

D. Nếu gặp câu lệnh break, vòng lặp sẽ bỏ qua các câu lệnh còn lại bên dưới trong vòng lặp và quay trở lại vòng lặp mới

8. Phát biểu đúng về câu lệnh continue?

A. Trong 1 vòng lặp chỉ được phép có 1 câu lệnh continue, không thể tồn tại cả break và continue hay nhiều hơn 2 câu lệnh continue

B. Khi câu lệnh continue được thực hiện, các câu lệnh bên dưới của continue sẽ bị bỏ qua và vòng lặp kết thúc ngay lập tức

C. Câu lệnh continue phải kết hợp với câu lệnh if

D. Khi câu lệnh continue được thực hiện, các câu lệnh bên dưới continue sẽ bị bỏ qua và vòng lặp tiếp tục thực hiện vòng lặp tiếp theo nếu điều kiện lặp còn đúng.

9. Output của chương trình sau là gì?

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

int main(){
    int n = 20;
    int tong = 0;
    for(int i = 1; i <= n; i *= 2){
        tong += i;
        if(i % 2 == 0){
            --tong;
        }
    }
    printf("%d", tong);
    return 0;
}
```

A.31

B.30

C.28

D.27

10. Output của chương trình sau là gì?

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

int main(){
    int n = 5;
    int tong = 0;
    for(int i = 1; ; i++){
        printf("%d ", i);
```

```

        if(i == n) break;
        else
            tong += i;
    }
    printf("%d", tong);
    return 0;
}

```

A.1 2 3 4 5 10

B.1 2 3 4 10

C.Lặp vô hạn

D.1 2 3 4 5 6 16

11.Vòng lặp sau lặp bao nhiêu lần?

```

for(int i = 1; i <= 10 && i % 2 == 1 ; i++){
}

```

A.Vô hạn lần

B.1

C.2

D.10

12. Output của chương trình sau là gì ?

```

#include <stdio.h>
#include <math.h>

int main(){
    int n = 10;
    int chan = 0, le = 0;
    for(int i = 1; i <= 10; i++){
        int chan = 0, le = 0;
        if(i % 2 == 0){
            chan += i;
        }
    }
}

```

```
    else{
        le += i;
    }
}
printf("%d %d", chan, le);
return 0;
}
```

A.0 0

B.30 25

C. Lỗi biên dịch

D.Vòng lặp vô hạn không có output

13.Vòng lặp sau lặp bao nhiêu lần?

```
for(int i = 1; i <= 7; i += 2){
    if(i % 2 == 0) ++i;
    else --i;
}
```

A.4

B.5

C.6

D.7

E.Lặp vô hạn lần

14. Vòng lặp sau lặp bao nhiêu lần?

```
int n = 1023456;
while(n >= 10){
    n /= 10;
}
```

A.5

B.6

C.7

D.4

15. Output của chương trình này là gì?

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

int main(){
    int n = 1023456;
    int tong = 0;
    while(n >= 10){
        n /= 10;
        tong += n % 10;
    }
    printf("%d", tong);
    return 0;
}
```

A.15

B.20

C.21

D.8

16. Output của chương trình sau là gì?

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

int main(){
    int n = 123456;
    int tong = 0;
    while(n){
        if(n % 2 == 0){
            tong += n % 10;
            n /= 10;
        }
    }
}
```

```
}  
printf("%d", tong);  
return 0;  
}
```

A.12

B.21

C.Lỗi biên dịch

D.Vòng lặp bị lặp vĩnh viễn

17. Output của chương trình sau là gì?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int n = 123456;  
    int rev = 0;  
    while(n){  
        if(n % 2 == 0){  
            rev = rev * 10 + n % 10;  
        }  
        n /= 10;  
    }  
    printf("%d", rev);  
    return 0;  
}
```

A.65432

B.642

C.Lỗi biên dịch do điều kiện lặp của while không đúng

D.654321

18.Vòng lặp while sau đây lặp bao nhiêu lần?

```
int n = 1234;
while(n > 0){
    if(n % 2 == 0){
        continue;
    }
    n /= 10;
}
```

A.4

B.3

C.Lặp vĩnh viễn

D.2

19.Output của chương trình sau là gì?

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

int main(){
    int n = 123456789;
    int sum = 0;
    while(n > 0){
        sum += n % 10;
        if(sum > 10) break;
        n /= 10;
        printf("%d ", n % 10);
    }
    printf("%d %d", sum, n);
    return 0;
}
```

A.9 17 12345678

B.9 17 1234567

C.8 17 12345678

D.8 17 1234567



20. Output của chương trình sau là gì?

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

int main(){
    int n = 123456789;
    while(n >= 10){
        int tong = 0;
        while(n > 0){
            tong += n % 10;
            n /= 10;
        }
        n = tong;
        printf("%d ", n);
    }
    return 0;
}
```

A.45 9 0

B.45 9

C. Lặp vô hạn

D. Lỗi biên dịch

21. Output của chương trình sau là gì?

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

int main(){
    int n = 5;
    int tong = 0;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < i; j++){
            tong++;
        }
    }
}
```

```
printf("%d", tong);  
return 0;  
}
```

A.10

B.11

C.12

D.9

22. Output của chương trình sau là gì?

```
#include <stdio.h>  
#include <math.h>  
  
int main(){  
    int n = 5, m = 3;  
    int tong = 0;  
    for(int i = 0; i < n; i++){  
        for(int j = 0; j < m; j++){  
            tong++;  
        }  
    }  
    printf("%d", tong);  
    return 0;  
}
```

A.25

B.9

C.15

D.26

23. Output của chương trình sau là gì?

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

```
#include <math.h>

int main(){
    int n = 36;
    for(int i = 2; i <= sqrt(n); i++){
        while(n % i == 0){
            printf("%d ", i);
            n /= i;
        }
    }
    if(n > 1) printf("%d ", n);
    return 0;
}
```

A. 2 2 3 3

B. 2 3

C. 2 2 3 3 36

D. 2 3 36

24. Output của chương trình sau là gì?

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

int main(){
    int a = 24, b = 18;
    while(a != b){
        if(a > b){
            a -= b;
        }
        else{
            b -= a;
        }
    }
    printf("%d", a);
    return 0;
}
```

A.6

B.18

C.3

D.Lặp vô hạn

25. Phát biểu về vòng lặp do-while là đúng?

A.Vòng lặp do-while có thể không thực hiện vòng lặp nào nếu điều kiện của vòng lặp sai

B.Vòng lặp do-while không thể bị lặp vĩnh viễn

C.Vòng lặp do-while có thể kết hợp với break và continue tương tự như for và while

D.Vòng lặp do-while chắc chắn sẽ thực hiện ít nhất 1 vòng lặp nếu điều kiện vòng lặp đúng

26.Output của chương trình sau là gì?

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

int main(){
    int dem = 0;
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < 3; j++){
            ++dem;
            if(j % 2 == 1){
                break;
            }
        }
    }
    printf("%d", dem);
    return 0;
}
```

A.12

B.8

C.9

D.10

27. Output của chương trình sau là gì?

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

int main(){
    int tong = 0, res = 0;
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < 3; j++){
            tong += j;
        }
        res += tong;
    }
    printf("%d", res);
    return 0;
}
```

A.30

B.32

C.45

D.12

28. Output của chương trình sau là gì?

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

int main(){
    int tong = 0, n = 28;
    for(int i = 1; i <= sqrt(n); i++){
        if(n % i == 0){
```

```

        tong += i;
        if(i != n / i){
            tong += n / i;
        }
    }
}
printf("%d", tong);
return 0;
}

```

A.56

B.28

C.7

D.14

29. Vòng lặp sau đây lặp tối đa và tối thiểu bao nhiêu lần?

```

for(int i = 2; i <= sqrt(n); i++){
    if(n % i == 0){
        break;
    }
}

```

A.  $\sqrt{n}$  và 1

B.  $\sqrt{n} - 1$  và 1

C.  $\sqrt{n} - 1$  và 0

D.  $\sqrt{n} - 2$  và 0

30. Phát biểu nào về vòng lặp dưới đây là đúng?

```

int tong = 0;
for(int i = 0; i <= n; i++){
    if(n % i == 0){
        tong += i;
    }
}

```

```
}
```

- A. Vòng lặp lặp tối đa  $n + 1$  vòng lặp
- B. Vòng lặp lặp tối thiểu 1 vòng lặp
- C. Biến tổng luôn có giá trị tối thiểu bằng  $n + 1$
- D. Vòng lặp có thể phát sinh lỗi

### III. HÀM

1. Kết quả của chương trình sau là gì?

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

int cal(int a, int b){
    int res = 2 * a + 3 * b;
    return 28;
    return res;
}

int main(){
    printf("%d", cal(10, 2));
    return 0;
}
```

- A. 26
  - B. 28
  - C. 28 26
  - D. Lỗi do hàm cal có 2 câu lệnh return
2. Kết quả của chương trình sau là gì?

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

int cal(int a, int b){
```

```
int res = 2 * a + 3 * b;  
return res;  
}  
  
int main(){  
    printf("%.2f", (float)cal(10.2, 2.1));  
    return 0;  
}
```

A.26.00

B.26.70

C.27.00

D.Chương trình báo lỗi biên dịch do truyền tham số chính thức không đúng kiểu dữ liệu với tham số hình thức

3.Phát biểu nào sau đây về hàm là sai?

A.Nếu hàm A gọi hàm B thì bắt buộc hàm B phải xây dựng trước hàm A

B.Khi lời gọi hàm được thực hiện, giá trị của đối số sẽ được gán cho tham số

C.Hàm có tham số là long long có thể áp dụng cho đối số là số float hoặc double

D.Mọi hàm kể cả hàm có kiểu trả về là void sẽ kết thúc ngay lập tức khi gặp câu lệnh return

4.Phát biểu nào về tham số và đối số là sai?

A.Tham số hay còn gọi là tham số hình thức, đối số là tham số chính thức

B.Khi lời gọi hàm xảy ra, đối số với tham số là 1

C.Số lượng đối số khi gọi hàm và số lượng tham số là giống nhau

D.Một hàm không giới hạn số lượng tham số

5. Kết quả của chương trình sau là gì?

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



```

long long cal(int a, int b){
    return a * b;
}

int main(){
    printf("%lld", cal(1000000.5, 10000000.2));
}

```

A.100000000000000

B.Lỗi biên dịch

C.Kết quả khác

D. 1.00000052e13

6. Kết quả của chương trình sau là gì?

```

#include <stdio.h>
#include <math.h>

void printOut(int a, int b, int c){
    printf("%d %d %d ", b, c, a);
    return;
    printf("%d %d %d", a, b, c);
}

int main(){
    printOut(10, 20, 30);
    return 0;
}

```

A.Lỗi biên dịch

B.20 30 10

C.20 30 10 10 20 30

D.10 20 30 10 20 30

7.Kết quả của chương trình sau là gì?

```

#include <stdio.h>
#include <math.h>

//my pow function
int pow(int a, int b){
    return a * b;
}

int main(){
    printf("%d", pow(2, 6));
    return 0;
}

```

A. Lỗi biên dịch do khai báo hàm trùng tên với hàm pow() trong thư viện math.h

B. 64

C. 12

D. 0

8. Kết quả của chương trình sau là gì?

```

#include <stdio.h>
#include <math.h>

//my pow function
long long factorial(int n){
    int res = 1;
    for(int i = 1; i <= n; i++){
        res *= i;
    }
    return res;
}

int main(){
    printf("%lld", factorial(15));
    return 0;
}

```

A. 1307674368000

B. 0

C. Lỗi biên dịch do trả về 1 số kiểu int trong khi hàm có kiểu trả về là long long

D. 2004310016

9. Kết quả của chương trình sau là gì?

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

void A(){
    printf("28tech\n");
    return;
    printf("C\n");
}

void B(){
    A();
    printf("Java\n");
    A();
}

int main(){
    B();
    return 0;
}
```

A.

28tech

Java

28tech

B.

28tech

C

Java

28tech

C

C.Lỗi biên dịch

D.

28tech

Java

10. Kết quả của chương trình sau là gì?

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

int tong(int n){
    int sum = 0;
    while(n){
        sum += n % 10;
        n /= 10;
    }
    return sum;
}

int tong2(int n){
    int ans = 0;
    while(n){
        ans += tong(n / 10);
        n /= 10;
    }
    return ans;
}

int main(){
    printf("%d", tong2(12345));
    return 0;
}
```

A.20

B.25

C.15

D.Không thể in ra kết quả vì vòng lặp bị lặp vô hạn

11. Kết quả của chương trình sau là gì?

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

int tong(int n){
    int sum = 0;
    while(n){
        sum += n % 10;
        n /= 10;
    }
    printf("%d", sum);
    return sum;
}

int main(){
    tong(12345);
    return 0;
}
```

A.15

B. 15 15

C. Lỗi biên dịch

D.15 0

12. Kết quả của chương trình sau là gì?

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

int tong(int n){
```

```

int sum = 0;
while(n){
    sum += n % 10;
    n /= 10;
}
printf("%d ", sum);
return sum;
printf("%d", n);
}

int main(){
    printf("%d", tong(12345));
    return 0;
}

```

A. 15 0 15

B. 15 15

C. 15

D. Lỗi biên dịch

13. Kết quả của chương trình sau là gì?

```

#include <stdio.h>
#include <math.h>

void pt(int n){
    for(int i = 2; i <= sqrt(n); i++){
        int dem = 0;
        while(n % i == 0){
            ++dem;
            n /= i;
        }
        if(dem) printf("%d ", i);
    }
    if(n > 1) printf("%d", n);
}

```

```
int main(){
    pt(1080);
    return 0;
}
```

A. 2 3 5

B. 2 3 4 5

C. 2 3 5 1

D. 2 2 2 3 3 3 5

14. Kết quả của chương trình sau là gì?

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

int dem = 0;

int nt(int n){
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0){
            return 0;
        }
        ++dem;
    }
    if(n > 1) printf("YES ");
    else printf("NO ");
    return n > 1;
}

int main(){
    if(nt(17)){
        printf("YES ");
    }
    else printf("NO ");
    printf("%d\n", dem);
    return 0;
}
```

A.YES 3

B.YES YES 3

C.NO YES 0

D.YES YES 0

15. Kết quả của chương trình sau là gì?

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

int dem = 0;

int check1(int n){
    ++dem;
    return (n % 2 == 0 ? 1 : 0);
}

int check2(int n){
    ++dem;
    int sum = 0;
    while(n){
        sum += n % 10;
        n /= 10;
    }
    return sum % 2 == 0;
}

int main(){
    for(int i = 1; i <= 20; i++){
        if(check2(i) && check1(i)){

        }
    }
    printf("%d\n", dem);
    return 0;
}
```



A.30

B.20

C.40

D.Lỗi biên dịch

16.Kết quả của chương trình sau là gì?

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

int findSum(int a, int b){
    return a + b;
}

int findSum(int c, int d){
    return c + d + 100;
}

int main(){
    printf("%d", findSum(10, 20));
    return 0;
}
```

A.30

B.130

C.Lỗi biên dịch

D.0

17. Chương trình sau có kết quả là gì?

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

int solve(int n){
    int tong = 0;
    while(n){
```

```

    tong += n % 10;
    n /= 10;
}
printf("%d ", tong);
return tong;
}

int main(){
    printf("%d", solve(1234));
    return 0;
}

```

A.10

B.10 10

C.Lỗi biên dịch

D.0

18.Phát biểu sau đây về hàm trong C là sai?

A.Bạn có thể xây dựng bao nhiêu hàm tùy ý

B.Bạn có thể xây dựng 2 hàm cùng tên

C.Bạn có thể xây dựng các hàm có cùng tên với các hàm có sẵn

D.Bạn có thể tự do đặt tên hàm theo ý của mình

19.Phát biểu về hàm có kiểu trả về là void nào sau đây là đúng?

A.Hàm void không thể có câu lệnh return

B.Hàm void thực thi chậm hơn so với các hàm trả về kiểu dữ liệu khác như int, float..

C.Hàm void sẽ kết thúc ngay khi gặp câu lệnh return

D.Hàm void có thể trả về các kiểu dữ liệu khác như int, float, char... nếu muốn

20.Phát biểu nào sau đây về hàm trong C là sai?

A.Sau khi hàm kết thúc thì đối số sẽ không bị thay đổi

B.Có thể thay đổi đối số thông qua hàm trong C

C. Giá trị của đối số bị thay đổi trong quá trình hàm thực hiện nhưng khi hàm kết thúc thì giá trị này được trả lại giá trị ban đầu

D. Tham số hình thức sẽ được thu hồi khi hàm kết thúc.

#### IV. ĐỘ PHỨC TẠP CỦA THUẬT TOÁN

1. Phát biểu nào sau đây là sai về độ phức tạp của thuật toán?

A. Thuật toán có độ phức tạp càng lớn chạy càng chậm

B. Độ phức tạp  $O(N \log N)$  tối ưu hơn độ phức tạp  $O(N^2)$

C. Độ phức tạp nhỏ nhất là  $O(1)$

D. Độ phức tạp lớn nhất là  $O(N!)$

2. Độ phức tạp của biểu thức :  $5 + 0.001n^3 + 0.025n$  là ?

A.  $O(N)$

B.  $O(N^3)$

C.  $O(0.001N^3)$

D.  $O(0.001N^3 + 0.025N)$

3. Độ phức tạp của biểu thức :  $500n + 100n^{1.5} + 50n \log_{10} n$  là ?

A.  $O(N)$

B.  $O(N^{1.5})$

C.  $O(N^2)$

D.  $O(N \log N)$

4. Độ phức tạp của chương trình sau là gì?

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

int tong(int n){
    int sum = 0;
    while(n){
        sum += n % 10;
```

```

        n /= 10;
    }
    return sum;
}

int main(){
    int n = 100;
    for(int i = 1; i <= n; i++){
        if(tong(i) % 5 == 0){
            printf("%d ", i);
        }
    }
}

```

A.  $O(N)$

B.  $O(N^2)$

C.  $O(N \log N)$

D.  $O(N/5)$

5. Ký hiệu big O mô tả trường hợp nào của thuật toán ?

A. Trường hợp tốt nhất

B. Trường hợp tệ nhất

C. Trường hợp trung bình

D. Trường hợp Tệ nhất và trung bình

6. Phát biểu sau đây về kí hiệu Omega, Theta và BigO là đúng?

A. Omega mô tả trường hợp tệ nhất, Theta mô tả trường hợp tốt nhất, BigO mô tả trường hợp trung bình

B. Omega mô tả trường hợp tốt nhất, Theta mô tả trường hợp tệ nhất, BigO mô tả trường hợp trung bình

C. Omega mô tả trường hợp tốt nhất, Theta mô tả trường hợp trung bình, BigO mô tả trường hợp tệ nhất

D. Omega mô tả trường hợp tệ nhất, Theta mô tả trường hợp trung bình, BigO mô tả trường hợp tốt nhất

7. Độ phức tạp của chương trình sau là gì?

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

int dem(int n){
    int ans = 0;
    while(n % 2 == 0){
        n /= 2;
        ++ans;
    }
    return ans;
}

int main(){
    int n = 100;
    for(int i = 1; i <= n; i++){
        printf("%d ", dem(i));
    }
}
```

A.  $O(N)$

B.  $O(N^2)$

C.  $O(N \log N)$

D.  $O(N\sqrt{N})$

8. Độ phức tạp của chương trình sau là gì?

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

int nt(int n){
    for(int i = 2; i <= sqrt(n); i++){
```

```

        if(n % i == 0) return false;
    }
    return n > 1;
}

int main(){
    int n = 100;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n + 5; j++){
            if(n % (i * j) == 0){
                printf("%d ", i * j);
            }
        }
    }
}

```

- A.  $O(N^2)$
- B.  $O(N^2 \log N)$
- C.  $O(N^2 \sqrt{N})$
- D.  $O(N^3)$

9. Vòng lặp sau thực hiện bao nhiêu câu lệnh code bên trong?

```

#include <stdio.h>
#include <math.h>

int main(){
    int n = 100;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            //code
        }
    }
}

```

- A. 5050

B.10000

C.200

D.100

10.Chương trình sau có độ phức tạp là bao nhiêu?

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

int main(){
    int n = 100;
    for(int i = 1; i <= n; i *= 2){
        printf("%d ", i);
    }
}
```

A.O(logN)

B.O(N)

C.O(N<sup>2</sup>)

D.O(N/2)

## V.ĐỆ QUY

1.Phát biểu nào sau về hàm đệ quy là sai?

A.Hàm đệ quy là hàm gọi lại chính nó 1 lần

B.Hàm đệ quy có thể gọi lại chính nó với đối số khác

C.Hàm đệ có thể là hàm có kiểu trả về là void

D.Hàm đệ quy có thể gây tràn bộ nhớ ngăn xếp

2.Hàm đệ quy sau đây được gọi bao nhiêu lần?

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

void re(int n){
    if(n > 0){
```

```

        if(n % 2 == 0){
            re(n / 2);
        }
        else{
            re(n / 10);1
        }
    }
}

int main(){
    re(1228);
}

```

A.6

B.7

C.1228

D.Gọi vô số lần cho tới khi tràn stack

3.Chương trình sau có kết quả là gì?

```

#include <stdio.h>
#include <math.h>

void dq(int n){
    if(n < 10){
        printf("%d ", n);
    }
    else{
        printf("%d ", n);
        dq(n / 10);
    }
}

int main(){
    dq(12345);
}

```



- A. 12345 1234 123 12 1
- B. 12345 1234 123 12
- C. 1 12 123 1234 12345
- D. 12 123 1234 12345

4. Chương trình sau có kết quả là gì?

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

int dq(int n){
    if(n < 10){
        if(n == 2 || n == 3 || n == 5 || n == 7){
            return n + 1;
        }
        else return 0;
    }
    else{
        int r = n % 10;
        if(r == 2 || r == 3 || r == 5 || r == 7){
            return r + 1 + dq(n / 10);
        }
        else{
            return dq(n / 10);
        }
    }
}

int main(){
    printf("%d", dq(1234567));
}
```

- A.17
- B.21
- C.18

D. Không có kết quả vì đệ quy không có điểm dừng

5. Chương trình sau có kết quả là gì?

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

int mypow(int x, int y){
    if(y == 0){
        return 1;
    }
    int tmp = mypow(x, y / 2);
    printf("%d ", tmp);
    if(y % 2 == 0){
        return tmp * tmp;
    }
    return tmp * tmp * x;
}

int main(){
    printf("%d", mypow(2, 9));
}
```

A. 1 2 4 16 512

B. 1 2 4 16

C. 1 2 4 8 16 32

D. 1 2 4 16 32 64 128 256 512

6. Chương trình sau có kết quả là gì?

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

int sum(int a[], int idx){
    if(idx >= 0){
        if(a[idx] % 2 == 0){
            return a[idx] + sum(a, --idx);
        }
    }
}
```

```

        else{
            return sum(a, --idx);
        }
    }
    else{
        return 0;
    }
}

int main(){
    int a[] = {1, 2, 3, 4, 5, 6};
    printf("%d", sum(a, 5));
}

```

A.12

B.10

C.21

D.Hàm đệ quy gây tràn bộ nhớ ngăn xếp

7. Chương trình sau có kết quả là gì?

```

#include <stdio.h>
#include <math.h>

void dq(int a[], int idx){
    if(idx >= 0){
        printf("%d ", a[idx / 2] );
        dq(a, idx - 1);
    }
}

int main(){
    int a[] = {1, 2, 3, 4, 5, 6};
    dq(a, 4);
}

```

A. 3 3 2 2 1 1

B. 3 2 2 1 1

C. 6 5 4 3 2 1

D. 3 2 1

8. Hàm đệ quy sau được gọi bao nhiêu lần?

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

int solve(int n){
    if(n == 1){
        return 0;
    }
    int x = 1e9, y = 1e9;
    if(n % 2 == 0){
        x = solve(n / 2) + 1;
    }
    if(n > 1){
        y = solve(n - 1) + 1;
    }
    return x < y ? x : y;
}

int main(){
    solve(6);
}
```

A.14

B.15

C.16

D.5

9. Hàm đệ quy sau được gọi bao nhiêu lần?

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

```

#include <math.h>

int T(int n){
    if(n == 0) return 0;
    if(n == 1) return 1;
    if(n == 2) return 1;
    return T(n - 1) + T(n - 2) + T(n - 3);
}

int main(){
    T(5);
}

```

A.12

B.13

C.27

D. Vô số lần cho tới khi tràn ngăn xếp

10. Hàm đệ quy sau được gọi bao nhiêu lần?

```

#include <stdio.h>
#include <math.h>

int F(int n){
    if(n == 0) return 0;
    if(n == 1) return 1;
    return F(n - 1) + F(n - 2);
}

int main(){
    F(5);
}

```

A.14

B.15

C.16

D.17

## VI. MẢNG

1. Phát biểu nào sau về mảng 1 chiều là sai?

A. Mảng 1 chiều không thể thêm và xóa phần tử khỏi mảng

B. Truy cập phần tử trong mảng thông qua chỉ số mất  $O(1)$

C. Các phần tử trong mảng nằm cạnh nhau trong bộ nhớ

D. Mảng 1 chiều không hỗ trợ truy cập vào chỉ số âm

2. Cho mảng 1 chiều A có 50 phần tử thuộc kiểu int, phát biểu nào sau đây là sai?

A. Mảng A chiếm 200 byte trong bộ nhớ

B. Chỉ số hợp lệ của mảng A là từ 0 tới 49

C. Mảng A không thể dùng để lưu số long long

D. Giá trị của A chính là một địa chỉ trong bộ nhớ

3. Phát biểu nào sau là đúng ?

A. Nếu 1 hàm có tham số là mảng 1 chiều, bạn phải chỉ rõ số lượng phần tử của mảng đó

B. Nếu 1 hàm có tham số là mảng 1 chiều, những thay đổi với phần tử của mảng trong hàm sẽ được giữ nguyên khi hàm kết thúc

C. Trong C bạn có thể khai báo mảng int có số lượng phần tử tùy ý

D. Mảng 1 chiều khi khai báo trong hàm main sẽ được cấp phát tại vùng nhớ ngăn xếp

4. Cho mảng A là mảng int gồm 10 phần tử, phần tử A[3] có địa chỉ là 1024. Phát biểu nào sau đây là đúng?

A. Phần tử A[4] có địa chỉ là 1028

B. Phần tử A[4] có địa chỉ là 1025

C. Phần tử A[4] có địa chỉ là 1032

D. Không thể xác định được địa chỉ của A[4]

5. Kết quả của chương trình sau là gì?

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

int main(){
    int a[5] = {1, 2, 3};
    printf("%d", a[4]);
    return 0;
}
```

A. Lỗi biên dịch

B. 0

C. Giá trị rác

D. -1

6. Cách khai báo mảng nào sau đây không hợp lệ?

A. int a[100];

B. int a[];

C. int[] a = {1, 2, 3};

D. int a[5] = {1, 2, 3, 4};

7. Kết quả của chương trình sau là gì?

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

int Find(int a[], int n, int x){
    for(int i = 0; i < n; i++){
        if(a[i] == x){
            return i;
        }
    }
    return -1;
}
```

```
}  
  
int main(){  
    int a[5] = {1, 2, 3, 8, 4};  
    printf("%d", Find(a, 5, 8));  
    return 0;  
}
```

A.-1

B.3

C.Lỗi biên dịch

D.Giá trị rác

8.Phát biểu nào về thuật toán tìm kiếm tuyến tính là sai?

A.Độ phức tạp của thuật toán tìm kiếm tuyến tính là  $O(N)$

B.Thuật toán tìm kiếm tuyến tính có thể áp dụng cho cả mảng không có thứ tự và mảng đã được sắp xếp

C.Thuật toán tìm kiếm tuyến tính có thể đạt được độ phức tạp  $O(1)$

D.Thuật toán tìm kiếm tuyến tính tìm kiếm nhanh hơn so với thuật toán tìm kiếm nhị phân

9.Phát biểu nào về thuật toán tìm kiếm nhị phân là sai?

A.Tìm kiếm nhị phân có độ phức tạp là  $O(\log N)$

B.Tìm kiếm nhị phân không thể áp dụng với mảng giảm dần

C.Tìm kiếm nhị phân có độ phức tạp tốt hơn tìm kiếm tuyến tính nhưng yêu cầu mảng đầu vào cần được sắp xếp

D.Tìm kiếm nhị phân thường được sử dụng khi cần kiểm tra sự tồn tại của phần tử trong nhiều trường hợp

10. Phát biểu nào về thuật toán tìm kiếm nhị phân là sai?

A.Đối với mảng chưa được sắp xếp thì thuật toán tìm kiếm nhị phân vẫn cho kết quả đúng nhưng với số bước thực hiện nhiều hơn



- B. Tìm kiếm nhị phân vẫn có thể áp dụng với mảng được sắp xếp giảm dần
- C. Tìm kiếm nhị phân có độ phức tạp chính xác là  $O(\log_2 N)$
- D. Tìm kiếm nhị phân có thể áp dụng với mảng số, mảng ký tự hoặc mảng struct với điều kiện mảng đã được sắp xếp
11. Phát biểu nào sau đây về thuật toán sắp xếp chọn là sai?
- A. Thuật toán sắp xếp chọn (Selection sort) có độ phức tạp là  $O(N^2)$
- B. Thuật toán sắp xếp chọn (Selection sort) có thể sắp mảng tăng hoặc giảm dần
- C. Thuật toán sắp xếp chọn (Selection sort) có tính ổn định (stable)
- D. Thuật toán sắp xếp chọn (Selection sort) có thể không cần thực hiện phép hoán đổi 2 phần tử nào.
12. Phát biểu nào sau đây về thuật toán sắp xếp nổi bọt là sai?
- A. Thuật toán sắp xếp nổi bọt (Bubble sort) có độ phức tạp là  $O(N^2)$
- B. Thuật toán sắp xếp nổi bọt (Bubble sort) có tính ổn định (stable)
- C. Thuật toán sắp xếp nổi bọt (Bubble sort) có độ phức tạp tốt nhất khi mảng đã sắp xếp tăng dần.
- D. Thuật toán sắp xếp nổi bọt (Bubble sort) có thể không sắp xếp được mảng giảm dần.
13. Phát biểu nào sau đây về thuật toán sắp xếp đếm phân phối là sai?
- A. Thuật toán đếm phân phối (Counting sort) có thể áp dụng với mảng số int
- B. Thuật toán đếm phân phối (Counting sort) dựa vào chỉ số của mảng đếm để đếm số lần xuất hiện của phần tử trong mảng ban đầu
- C. Thuật toán đếm phân phối (Counting sort) có độ phức tạp là  $O(N + K)$  trong đó  $N$  là số lượng phần tử trong mảng và  $K$  là phần tử có giá trị lớn nhất
- D. Thuật toán đếm phân phối (Counting sort) có tính ổn định (stable)
14. Phát biểu nào sau đây về thuật toán sắp xếp nhanh là sai?
- A. Thuật toán sắp xếp nhanh (Quick sort) có độ phức tạp tệ nhất là  $O(N^2)$

- B. Thuật toán sắp xếp nhanh (Quick sort) có tính ổn định (stable)
- C. Thuật toán sắp xếp nhanh (Quick sort) có thể sử dụng một trong 2 phân hoạch là Lomuto và phân hoạch Hoare
- D. Thuật toán sắp xếp nhanh (Quick sort) được sử dụng rất rộng rãi trong thực tế
15. Phát biểu nào sau đây về thuật toán sắp xếp trộn là sai?
- A. Thuật toán sắp xếp trộn (Merge sort) có độ phức tạp là  $O(N \log N)$
- B. Thuật toán sắp xếp trộn (Merge sort) có tính ổn định (stable)
- C. Thuật toán sắp xếp trộn (Merge sort) là một giải thuật chia và trị
- D. Thuật toán sắp xếp trộn (Merge sort) có độ phức tạp tệ nhất là  $O(N^2)$
16. Để tìm kiếm phần tử X trong mảng A[] thì cách nào sau đây sẽ hợp lý nhất?
- A. Áp dụng thuật toán tìm kiếm nhị phân
- B. Áp dụng thuật toán tìm kiếm tuyến tính
- C. Áp dụng thuật toán sắp xếp sau đó áp dụng thuật toán tìm kiếm nhị phân
- D. Áp dụng thuật toán sắp xếp nhanh sau đó áp dụng thuật toán tìm kiếm nhị phân
17. Phát biểu nào sau đây là đúng?
- A. Để kiểm tra 1 mảng số nguyên tăng dần chỉ cần 1 vòng lặp là đủ
- B. Đếm số lần xuất hiện của 1 phần tử trong mảng số nguyên cần độ phức tạp tối thiểu là  $O(N^2)$
- C. Để tìm phần tử nhỏ nhất và lớn nhất trong mảng thì cách tốt nhất là sắp xếp mảng tăng dần sau đó in ra phần tử đầu tiên và cuối cùng
- D. Mảng tăng dần không thể đối xứng
18. Phần tử trung vị (median) của mảng là?
- A. Phần tử có giá trị là trung bình cộng của các số trong mảng
- B. Phần tử nằm chính giữa của mảng khi mảng được sắp xếp tăng dần
- C. Phần tử nằm chính giữa của mảng ban đầu
- D. Phần tử là bội chung nhỏ nhất của tất cả các phần tử trong mảng

19. Mảng đánh dấu có thể áp dụng với bài toán nào?

A. Mảng số int không vượt quá  $10^7$

B. Mảng số int không âm

C. Mảng số int không âm và không vượt quá  $10^7$

D. Mảng số int và float không âm

20. Khai báo mảng 2 chiều nào sau đây là hợp lệ?

A. `int a[2][3] = {{1, 2}, {4, 5}, {2, 1}};`

B. `int a[2][3] = {{1, 2}, {4, 5, 6}};`

C. `int a[3][2] = {{1, 2, 3}, {2, 2, 1}};`

D. `int a[2][2] = {{1, 2, 3, 4}};`

## VII. CON TRỎ

1. Phát biểu nào sau đây về con trỏ là sai?

A. Biến con trỏ có thể lưu địa chỉ của 1 biến khác

B. Nhiều con trỏ có thể cùng lưu địa chỉ của 1 biến

C. Con trỏ có thể dùng để lưu giá trị địa chỉ của 1 con trỏ khác

D. Thông qua con trỏ có thể truy cập giá trị của biến mà nó lưu địa chỉ.

2. Khi khai báo `int *ptr, n;` thì

A. Cả `ptr` và `n` đều là biến con trỏ

B. `ptr` là biến con trỏ kiểu `int`, `n` là biến `int`

C. `ptr` là biến con trỏ kiểu `int`, `n` có thể là biến con trỏ hoặc biến `int`

D. Lỗi biên dịch

3. Cho địa chỉ của biến `n` trong bộ nhớ là 2048, chương trình sau có kết quả là gì?

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

```
int main(){  
    int n = 100;
```

```
int *ptr1 = &n;
printf("%d ", ptr1); // cout << ptr1 << endl;
printf("%d\n", *ptr1); // cout << *ptr1 << endl;
}
```

- A.100 100
- B.2048 100
- C.2048 2048
- D.Lỗi biên dịch

4.Kết quả của chương trình sau là gì?

```
#include <stdio.h>

int main(){
    int n = 100;
    int *ptr1 = &n;
    int *ptr2 = ptr1;
    *ptr2 = 200;
    *ptr1 = 500;
    printf("%d %d %d", n, *ptr1, *ptr2);
}
```

- A.100 100 200
- B.500 500 500
- C.200 200 200
- D.100 100 100

5.Kết quả của chương trình sau là gì, biết rằng địa chỉ của biến n là 2048?

```
#include <stdio.h>

void change(int *x){
    printf("%d ", x);
    *x = 20;
}
```

```

}

int main(){
    int n = 100;
    change(&n);
    printf("%d ", n);
}

```

- A.100 20
- B.2048 20
- C.2048 100
- D.100 100

6.Kết quả của chương trình sau là gì, biết rằng địa chỉ của n trong bộ nhớ là 2048?

```

#include <stdio.h>

int* change(int *x){
    printf("%d ", x);
    *x = 20;
    return x;
}

int main(){
    int n = 100;
    int *ptr = &n;
    ptr = change(ptr);
    printf("%d %d", ptr, n);
}

```

- A.2048 2048 100
- B.2048 2048 20
- C.100 100 100
- D.Lỗi biên dịch

7. Kết quả của chương trình sau là gì?

```
#include <stdio.h>

int main(){
    int n = 100;
    int *ptr1 = &n;
    int *ptr2 = *ptr1;
    int *ptr3 = *ptr2;
    *ptr3 = 20;
    printf("%d", n);
}
```

A.20

B.100

C.0

D.Lỗi biên dịch

8. Câu lệnh printf trong chương trình in ra giá trị gì?

```
#include <stdio.h>

int main(){
    int n = 100;
    int *ptr1 = &n;
    int **ptr2 = &ptr1;
    printf("%d %d %d", *ptr1, *ptr2, **ptr2);
}
```

A. Giá trị của n, giá trị của con trỏ ptr1, giá trị của n

B. Giá trị của n, địa chỉ của ptr1, địa chỉ của n

C. Giá trị của n, giá trị của n, giá trị của n

D. Giá trị của n, địa chỉ của ptr2, giá trị của biến mà ptr1 đang trỏ tới

9. Kết quả của chương trình sau là gì, biết rằng input là “28tech.com.vn”?

```
#include <stdio.h>

int main(){
    char *c;
    gets(c);
    printf("%s", c);
}
```

A.28tech.com.vn

B.2

C.Lỗi thực thi

D.Lỗi biên dịch

10.Kết quả của chương trình sau là gì?

```
#include <stdio.h>

void update(int *n, int *m){
    *n += 20;
    *m += 30;
}

int main(){
    int n = 100;
    int *ptr = &n;
    update(&n, ptr);
    printf("%d", n);
}
```

A.120

B.130

C.100

D.150

11.Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <string.h>

struct SV{
    char ten[100];
    char *lop;
};

int main(){
    SV x;
    strcpy(x.ten, "28tech");
    strcpy(x.lop, "CNTT");
    SV *ptr = &x;
    printf("%s %s", ptr->ten, ptr->lop);
}

```

A.28tech CNTT

B.28tech 28tech

C.null null

D.Lỗi thực thi

12.Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <string.h>

void change(int **n, int *m){
    *(*n) *= 2;
    *m += 10;
}

int main(){
    int a = 10;
    int *ptr1 = &a;
    int **ptr2 = &ptr1;
    change(ptr2, ptr1);
    printf("%d", a);
}

```



```
}
```

A.10

B.20

C.30

D.Lỗi biên dịch

13. Câu lệnh printf trong chương trình sau in giá trị gì ?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    printf("%d %d", a, (a + 4));
}
```

A.a[0] và a[4]

B.Địa chỉ của a[0] và địa chỉ của a[4]

C.Giá trị rác và giá trị của a[4]

D.Chương trình sẽ phát sinh lỗi biên dịch

14.Chương trình sau cho kết quả là gì?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    int *ptr = &a[2];
    ptr += a[1];
    printf("%d", *(ptr + 1));
}
```

A.1

B.5

C.3

D.Lỗi biên dịch

15.Cho chương trình sau, phát biểu nào là đúng?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    int *ptr = a;
    ptr += *(a + 2);
}
```

A.ptr đang trỏ tới phần tử đầu tiên trong mảng a

B.ptr đang quản lý a[3]

C.\*ptr có giá trị 3

D. Lỗi biên dịch

16.Chương trình sau cho kết quả là gì?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    int *ptr = a + 4;
    ++a;
    printf("%d %d", *a, *ptr);
}
```

A.1 5

B.2 5

C.2 4

#### D. Lỗi biên dịch

17. Chương trình sau cho kết quả là gì?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    int *ptr1 = a;
    int *ptr2 = &a[2];
    --ptr2; ptr2--;
    *ptr2 = 20;
    *ptr1 += 10;
    printf("%d %d", *ptr1, *ptr2);
}
```

A. 30 30

B. 30 11

C. 11 20

D. Lỗi biên dịch

18. Chương trình sau cho kết quả là gì?

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

int main(){
    int a[] = {1, 2, 3, 4, 5, 1};
    int *ptr1 = (a + 3);
    int **ptr2 = &ptr1;
    (*ptr2) = a + 4;
    *ptr1 += 10;
    *(*ptr2) += 20;
    printf("%d %d", *ptr1, **ptr2);
}
```

A.30 35

B.35 35

C.15 20

C.Lỗi biên dịch

19.Chương trình sau cho kết quả là gì?

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

int main(){
    char a[] = "28tech.com.vn";
    char *ptr = &a[2];
    printf("%s", ptr);
}
```

A.t

B.tech.com.vn

C.28tech.com.vn

D.Lỗi biên dịch

20.Chương trình sau cho kết quả là gì?

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

int main(){
    char a[] = "28tech.com.vn";
    char *ptr = &a[2];
    char *ptr2 = ptr;
    *ptr2 = 'T';
    printf("%s", (a + 2));
}
```

A.tech.com.vn

B.Tech.com.vn

C.28Tech.com.vn

D.Lỗi biên dịch

21.Chương trình sau cho kết quả là gì?

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

int main(){
    char a[] = "28tech.com.vn";
    char *ptr = a;
    char *ptr2 = ptr;
    *(ptr2 + 6) = '\0';
    printf("%s", a);
}
```

A.28tech

B.28tech.com.vn

C. .com.vn

D.Lỗi biên dịch

22. Chương trình sau cho kết quả là gì?

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

int main(){
    char *c = (char*)malloc(20 * sizeof(int));
    strcpy(c, "28tech.com.vn");
    char *ptr1 = (c + 0);
    char *ptr2 = (c + 1);
    int x = (*ptr2 - *ptr1);
    printf("%s", (ptr1 + x));
    free(c);
}
```

A.28tech.com.vn

B. .com.vn

C.28tech

D.Lỗi biên dịch

23.Chương trình sau có kết quả là gì?

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

int main(){
    int n = 100;
    int *ptr1 = &n;
    int **ptr2 = &ptr1;
    int ***ptr3 = &ptr2;
    printf("%d %d %d %d", sizeof(n), sizeof(ptr1), sizeof(ptr2), sizeof(ptr3));
}
```

A.4 4 4 4

B.4 8 8 8

C.4 8 12 16

D.4 8 16 32

24.Kết quả của chương trình sau là gì ?

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

int main(){
    int a[2][3] = {
        {1, 2, 3},
        {4, 5, 6}
    }
```

```
};  
int *ptr = &a[0][0];  
printf("%d", *(ptr + 4));  
}
```

A.4

B.5

C.Giá trị rác

D.Lỗi biên dịch

25. Cho mã nguồn sau :

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

Cho biết địa chỉ của phần tử  $a[0][0]$  là 1024, vậy câu lệnh printf trong hàm main in ra giá trị bao nhiêu?

A.1028

B.1044

C.1048

D.Lỗi biên dịch

26.Cho mảng 2 chiều  $a[][]$  có n hàng, m cột, phát biểu nào sau đây là đúng?

A. $(a + 1)$  là địa chỉ của phần tử  $a[1][1]$  trong mảng a

B.  $*(a + 1)$  là địa chỉ của phần tử  $a[1][0]$

C. Muốn truy cập phần tử  $a[i][j]$  có thể sử dụng  $*(a + m * i + j)$

D. Muốn truy cập phần tử  $a[i][j]$  có thể sử dụng  $*(a + n * i + j)$

27. Chương trình sau cho kết quả là gì?

```
#include <stdio.h>

int main(){
    int a[4][5] = {
        {1, 2, 3, 4, 5},
        {6, 7, 8, 9, 10},
        {1, -2, 3, 0, 6},
        {1, 0, 0, 9, 1}
    };
    int *ptr = &a[0][0];
    *(ptr + 13) = 28;
    ++ptr;
    printf("%d ", *ptr);
    printf("%d", (*(a + 2) + 3));
}
```

A. 2 28

B. 0 28

C. 2 2

D. Lỗi biên dịch

28. Cho mảng 2 chiều  $a[][]$  có  $n$  hàng,  $m$  cột, phát biểu nào sau đây là đúng?

A.  $*(a + 2) + 3$  là giá trị của phần tử  $a[2][3]$  trong mảng

B.  $*(a) + 1$  là con trỏ trỏ tới phần tử  $a[0][1]$

C.  $*(a + 2) + 4$  và  $a[2][4]$  là 2 giá trị khác nhau

D. Để truy cập vào phần tử  $a[i][j]$  ta có thể thông qua biểu thức  $*(a + m * i + j)$

29. Chương trình sau có kết quả là gì?



```
#include <stdio.h>
int main(){
    int a[4][5] = {
        {1, 2, 3, 4, 5},
        {6, 7, 8, 9, 10},
        {1, -2, 3, 0, 6},
        {1, 0, 0, 9, 1}
    };
    int (*p)[5];
    p = a;
    printf("%d", *((p + 2) + 3));
}
```

A.0

B.5

C.-2

D.Lỗi biên dịch

30.Chương trình sau có kết quả là gì?

```
#include <stdio.h>

int main(){
    int n = 891;
    int *ptr1 = &n;
    char *ptr2 = (char*)ptr1;
    *ptr2 = 0;
    printf("%d", n);
}
```

A.768

B.891

C.0

D. Lỗi biên dịch

## VIII. DANH SÁCH LIÊN KẾT

1. Phát biểu nào sau đây là sai về DSLK đơn?

- A. Thao tác chèn một node vào DSLK đơn có độ phức tạp là  $O(N)$
- B. Chèn một node vào đầu DSLK đơn có độ phức tạp là  $O(1)$
- C. Truy xuất phần tử ở vị trí bất kỳ trong DSLK đơn có độ phức tạp là  $O(1)$
- D. Xóa phần tử khỏi DSLK đơn ở vị trí đầu tiên có độ phức tạp là  $O(1)$

2. Thông tin nào được lưu trong một node của DSLK đôi

- A. Giá trị của node
- B. Địa chỉ của node kế tiếp
- C. Địa chỉ của node trước
- D. Cả 3 đáp án trên

3. Độ phức tạp của thao tác đếm số node trong DSLK đơn là bao nhiêu?

- A.  $O(N)$
- B.  $O(1)$
- C.  $O(\log N)$
- D. Không đáp án nào trong 3 đáp án trên

4. Ứng dụng của DSLK ?

- A. Cài đặt hệ thống file
- B. Bảng băm
- C. Cài đặt cây nhị phân
- D. Cả 3 phương án trên

5. Thêm một node vào giữa trong DSLK cần thay đổi bao nhiêu con trỏ?

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

D.4

6. Thêm một node vào cuối trong DSLK cần thay đổi bao nhiêu con trỏ

A.1

B.2

C.0

D.3

7. DSLK có thể sử dụng để ?

A. Cài đặt ngăn xếp, hàng đợi

B. Cài đặt cây nhị phân

C. Cài đặt các cấu trúc dữ liệu tương tự như một mảng động

D. Cả 3 phương án trên

8. Thêm 1 node vào đầu DSLK đơn được quản lý bởi node đầu tiên có độ phức tạp là bao nhiêu?

A.  $O(1)$

B.  $O(\log N)$

C.  $O(N)$

D. Đáp án khác

9. Để xóa các phần tử trùng nhau trong 1 DSLK các phần tử đã được sắp xếp tăng dần có độ phức tạp là?

A.  $O(N)$

B.  $O(N^2)$

C.  $O(\log N)$

D. Đáp án khác

10. DSLK đơn được quản lý bởi node đầu tiên là head, vậy head lưu giá trị gì?

A. Giá trị của nó và địa chỉ của node tiếp theo

B. Head có giá trị là một địa chỉ

C.Head lưu địa chỉ của node tiếp theo trong DSLK

D.Head không có giá trị xác định

Cấu trúc node cho các câu hỏi về DSLK đơn phía dưới

```
struct node{
    int data;
    node *next;
};

typedef struct node node;
```

11.Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, output của chương trình là gì khi gọi hàm printOut?

```
struct node{
    int data;
    node *next;
};

typedef struct node node;

void printOut(node *head){
    node *tmp = head->next;
    cout << tmp->next->next->data << endl;
}
```

A.5

B.3

C.4

5. Không xác định được giá trị

12. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau với con trỏ head là gì?

```
head->next->next->next = NULL;
```

A.1->2->3

B.1->2->3->4

C.1->2->3->4->5

D.4->5

13. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau với con trỏ head là gì?

```
head->next->next->data = 100;
```

A.1->2->100->3->4->5

B.1->2->100->3->4

C.1->2->100->4->5

D.1->2->100

14. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```
node *newnode = (node*)malloc(sizeof(node));  
newnode->data = 28;  
newnode->next = NULL;  
node *tmp = head;  
tmp->next->next = newnode;
```

A.1->2->3->28->4->5

B.1->2->28

C.1->2->28->4->5

D.1->2->3->28->5

15. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```
node *newnode = (node*)malloc(sizeof(node));
```

```

newnode->data = 28;
newnode->next = NULL;
node *tmp = head;
while(tmp != NULL){
    tmp = tmp->next;
}
tmp->next = newnode;

```

A.1->2->3->4->5->28

B.1->2->3->4->28

C.1->2->3->5->28

D.Chương trình phát sinh lỗi thực thi

16. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```

node *tmp = head;
while(tmp->next->next != NULL){
    tmp = tmp->next;
}
tmp->data = 100;
tmp->next->data = 200;

```

A.1->2->3->4->100->200

B.1->2->3->100->200

C.1->2->3->4->200

D.1->2->3->4->5->100->200

17. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```

node *newnode = (node*)malloc(sizeof(node));
newnode->data = 28;
newnode->next = NULL;
node *tmp = head;

```

```
while(tmp->next != NULL){  
    tmp = tmp->next;  
}  
tmp->next = newnode;
```

A.1->2->3->4->5->28

B.1->2->3->4->28

C.1->2->3->28->4->5

D.1->2->3->4->28->5

18. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```
node *tmp = head;  
while(tmp->next != NULL){  
    if(tmp->data % 2 == 1){  
        tmp->data = 28;  
    }  
    tmp = tmp->next;  
}
```

A.28->2->28->4->28

B.28->2->28->4->5

C.1->2->28->4->5

D.1->2->3->4->28

19. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, output của chương trình sau là gì?

```
int ans = 0;  
node *tmp = head;  
while(tmp != NULL){  
    if(tmp->data % 2 == 0){  
        ++ans;  
    }  
}
```

```
tmp = tmp->next->next;
}
printf("%d", ans);
```

A.2

B.0

C.5

D.Lỗi thực thi

20. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện câu lệnh sau là gì?

```
node *newNode = (node*)malloc(sizeof(node));
newNode->data = 27;
newNode->next = NULL;
node *tmp = head;
while(tmp != NULL){
    if(tmp->data % 2 == 0){
        tmp->next = newNode;
    }
    tmp = tmp->next;
}
```

A.1->2->27->3->4->27->5

B.1->2->27

C.1->27->3->27->5

D.Lỗi biên dịch

21. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện hàm sau với node head là gì?

```
void change(node *head){
```



```

node *newNode = (node*)malloc(sizeof(node));
newNode->data = 28;
newNode->next = NULL;
node *tmp = head;
while(tmp->next != NULL){
    tmp = tmp->next;
}
tmp->next = newNode;
}

```

A.1->2->3->4->5->28

B.1->2->3->4->28

C.1->2->3->4->5

D.Rỗng

22. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện các câu lệnh sau là gì?

```

node *newNode = (node*)malloc(sizeof(node));
newNode->data = 28;
newNode->next = NULL;
node *tmp = head;
for(int i = 1; i <= 2; i++){
    tmp = tmp->next;
}
newNode->next = tmp->next;
tmp->next = newNode;

```

A.1->2->3->28->4->5

B.1->2->28->3->4->5

C.1->2->28->4->5

D.Rỗng

23. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện các câu lệnh sau là gì?

```

node *newNode = (node*)malloc(sizeof(node));
newNode->data = 28;
newNode->next = NULL;
node *tmp = head;
for(int i = 1; i <= 2; i++){
    tmp = tmp->next;
}
tmp=tmp->next->next;

```

A.1->2->3->5

B.1->2->4->5

C.1->2->5

D.1->2->3

24. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện các câu lệnh sau là gì?

```

node *newNode = (node*)malloc(sizeof(node));
newNode->data = 28;
newNode->next = NULL;
node *tmp = head;
for(int i = 1; i <= 4; i++){
    tmp = tmp->next;
}
tmp=tmp->next->next;

```

A.1->2->3->4

B.1->2->3->5

C.1->2->3

D.Lỗi thực thi

25. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5 và mã nguồn sau, DSLK sau khi thực hiện các câu lệnh sau là gì?

```
while(head->next != NULL){  
    head = head->next;  
}  
head->data = 20;
```

A.1->2->3->4->20

B.1->2->3->4->5->20

C.20

D.Rỗng

26. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, phát biểu nào sau đây là đúng?

A.head->data đang lưu địa chỉ của node có giá trị 1 trong DSLK

B.head->next đang lưu địa chỉ của node có giá trị 2 trong DSLK

C.head đang lưu giá trị node 1

D.head->next->next->data lưu địa chỉ của node 3 trong DSLK

27. Cho DSLK được quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, địa chỉ của các node lần lượt là 10, 20, 30, 40, 50 phát biểu nào sau đây là đúng?

A. head->next->next->data là 30

B.head->next->next là 30

C.head->next là 2

D.head->next->next->next->data là 4

Cấu trúc node sau dành cho các câu hỏi về DSLK đôi bên dưới

```
struct node{  
    int data;  
    node *next;  
    node *prev;
```

```
};
```

```
typedef struct node node;
```

28. Cho DSLK đôi quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, câu lệnh sau cho kết quả là gì?

```
printf("%d", head->next->next->prev->data);
```

A.2

B.3

C.1

D. Giá trị rác

29. Cho DSLK đôi quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, phát biểu nào sau đây là sai?

A. head->next là địa chỉ của node có giá trị 2 trong DSLK

B. head->next->prev là địa chỉ của node đầu tiên trong DSLK

C. head->next->prev->data có giá trị là 2

D. head->next->next->prev->next lưu địa chỉ của node thứ 3 trong DSLK

30. Cho DSLK đôi quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, câu lệnh sau cho kết quả là gì?

```
printf("%d", head->prev->next->next->data);
```

A.2

B.3

C.1

D. Lỗi thực thi

31. Cho DSLK đôi quản lý bởi con trỏ head có các phần tử 1->2->3->4->5, DSLK sau khi thực hiện các câu lệnh sau là gì?

```
node *tmp = head;
while(tmp != NULL){
    if(tmp->data == 3){
        break;
    }
    tmp = tmp->next;
}
tmp->prev->data = 28;
```

A. 1->2->28->4->5

B. 1->28->3->4->5

C. 1->2->28->3->4->5

D. 1->2->3->4->5

32. Chương trình sau cho kết quả là gì?

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

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

void add(node *head, int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    head->next = newNode;
}
```

```
int main(){
    node *head = NULL;
    add(head, 100);
    printf("%d", head->data);
}
```

- A.100
- B.Không có output
- C.Lỗi thực thi
- D.100 100

33.Chương trình sau có kết quả là gì?

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

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

int main(){
    node *head = NULL;
    head = taoNode(1);
```

```

head->next = taoNode(2);
head->next->next = taoNode(3);
head->next->data = 4;
while(head != NULL){
    printf("%d ", head->data);
    head = head->next;
}
}

```

A.1 2 3 4

B.1 4 3

C.1 2 3

D. 1 2 4 3

34.Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

int main(){

```

```

node *head = NULL;
node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
head = x; head->next = y;
y->prev = head;
y->next = z; z->prev = y;
z->next = t; t->prev = z;
head->next->next->prev->data = 5;
while(head != NULL){
    printf("%d ", head->data);
    head = head->next;
}
}

```

A.1 2 3 4 5

B.1 5 3 4

C.1 2 5 3 4

D.1 2 5 4

35. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

```



```

int main(){
    node *head = NULL;
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    head = taoNode(10);
    while(head != NULL){
        printf("%d ", head->data);
        head = head->next;
    }
}

```

- A.1 2 3 4 10
- B.10 1 2 3 4
- C.10
- D.Lỗi thực thi

36. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));

```

```

newNode->data = x;
newNode->next = newNode->prev = NULL;
return newNode;
}

void themcuoi(node *head, int x){
    node *newNode = taoNode(x);
    node *tmp = head;
    while(tmp->next != NULL){
        tmp = tmp->next;
    }
    newNode->prev = tmp;
    tmp->next = newNode;
}

int main(){
    node *head = NULL;
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    themcuoi(head, 5);
    while(head != NULL){
        printf("%d ", head->data);
        head = head->next;
    }
}

```

A.1 2 3 4

B.1 2 3 4 5

C.5

D.Output rỗng

37.Chương trình sau có kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

void themcuoi(node *head, int x){
    node *newNode = taoNode(x);
    node *tmp = head;
    while(tmp->next != NULL){
        tmp = tmp->next;
    }
    newNode->prev = tmp;
    tmp->next = newNode;
}

int main(){
    node *head = NULL;
    themcuoi(head, 0);
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    themcuoi(head, 5);
    while(head != NULL){

```

```
    printf("%d ", head->data);  
    head = head->next;  
}  
}
```

A.0 1 2 3 4 5

B.1 2 3 4 5

C.0

D.Lỗi thực thi

38. Chương trình sau có kết quả là gì?

```
#include <stdio.h>  
#include <stdlib.h>  
  
struct node{  
    int data;  
    node *next;  
    node *prev;  
};  
  
typedef struct node node;  
  
node *taoNode(int x){  
    node *newNode = (node*)malloc(sizeof(node));  
    newNode->data = x;  
    newNode->next = newNode->prev = NULL;  
    return newNode;  
}  
  
void themcuoi(node *head, int x){  
    node *newNode = taoNode(x);  
    while(head->next != NULL){  
        head = head->next;  
    }  
}
```

```

newNode->prev = head;
head->next = newNode;
}

int main(){
    node *head = NULL;
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    themcuoi(head, 5);
    while(head != NULL){
        printf("%d ", head->data);
        head = head->next;
    }
}

```

A.1 2 3 4 5

B.1 2 3 4

C.5

D.Lỗi thực thi

39. Chương trình sau có kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

```

```

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

void themcuoi(node **head, int x){
    node *newNode = taoNode(x);
    while((*head)->next != NULL){
        (*head) = (*head)->next;
    }
    newNode->prev = *head;
    (*head)->next = newNode;
}

int main(){
    node *head = NULL;
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    themcuoi(&head, 5);
    while(head != NULL){
        printf("%d ", head->data);
        head = head->next;
    }
}

```

A.1 2 3 4 5

B.4 5

C.5

D.Lỗi thực thi

40. Chương trình sau có kết quả là gì?

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

struct node{
    int data;
    node *next;
    node *prev;
};

typedef struct node node;

node *taoNode(int x){
    node *newNode = (node*)malloc(sizeof(node));
    newNode->data = x;
    newNode->next = newNode->prev = NULL;
    return newNode;
}

node* themcuoi(node *head, int x){
    node *newNode = taoNode(x);
    node *tmp = head;
    while(head->next != NULL){
        head = head->next;
    }
    newNode->prev = head;
    head->next = newNode;
    return tmp;
}

int main(){
    node *head = NULL;
    node *x = taoNode(1), *y = taoNode(2), *z = taoNode(3), *t = taoNode(4);
    head = x; head->next = y;
    y->prev = head;
    y->next = z; z->prev = y;
    z->next = t; t->prev = z;
    head = themcuoi(head, 5);
```

```

while(head != NULL){
    printf("%d ", head->data);
    head = head->next;
}
}

```

A.1 2 3 4 5

B.4 5

C.5

D.Lỗi thực thi

## IX. CÂY NHỊ PHÂN

1. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    root->left = x; x->left = y;

```



```

y->right = z; y->left = t;
while(root != NULL){
    printf("%d ", root->data);
    root = root->left;
}
}

```

A.1 2 3 4

B.1 2 3 4 5

C.1 2 3 5

D.Lỗi thực thi

2. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
}

```

```

x->left = z; x->right = t;
y->left = u; y->right = v;
int tong = 0;
while(root != NULL){
    tong += root->data;
    root = root->left;
}
printf("%d", tong);
}

```

A.28

B.7

C.6

D.Lỗi thực thi

3. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
}

```

```

node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
node *u = taoNode(6), *v = taoNode(7);
root->left = x; root->right = y;
x->left = z; x->right = t;
y->left = u; y->right = v;
while(root != NULL){
    printf("%d ", root->data);
    root = root->right;
}
}

```

A.1 3 7

B.1 2 5

C.1 3 6 7

D.1 2 3 4 5 6 7

4. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){

```

```

node *root = taoNode(1);
node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
node *u = taoNode(6), *v = taoNode(7);
root->left = x; root->right = y;
x->left = z; x->right = t;
y->left = u; y->right = v;
int ans = 0;
while(root != NULL){
    if(root->data % 2 == 1){
        ans += root->data;
        root = root->right;
    }
}
printf("%d", ans);
}

```

A.11

B.10

C.4

D.Không có output vì vòng lặp bị lặp vĩnh viễn

5. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
}

```

```

    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    node *tmp = root;
    while(tmp != NULL){
        if(tmp->data % 2 == 1){
            tmp->data = 28;
        }
        tmp = tmp->right;
    }
    printf("%d", root->right->data + root->left->data);
}

```

A.30

B.28

C.4

D.56

6. Cho mã nguồn sau, câu lệnh printf trong chương trình in ra giá trị gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

```

```

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    printf("%d", root->left->right);
}

```

A.Số 5

B.Địa chỉ của node z

C.Địa chỉ của node t

D.Không thể xác định giá trị

7. Cho mã nguồn sau, câu lệnh printf trong chương trình in ra giá trị gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

```

```

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    t->left = taoNode(9);
    printf("%d", root->left->right->left->data);
}

```

A.Số 5

B.Số 9

C.Địa chỉ của node t trong cây nhị phân

D.Câu lệnh printf gây lỗi thực thi nên không có output

8. Cho mã nguồn sau, tổng số node trên cây NP sau khi các câu lệnh trong main được thực hiện xong là bao nhiêu?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;
    struct node *right;
};

typedef struct node node;

```

```

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    node *tmp = root;
    while(tmp != NULL){
        if(tmp->data % 2 == 0){
            tmp->left = NULL;
            tmp->right = NULL;
        }
        tmp = tmp->left;
    }
}

```

A.4

B.5

C.6

D.7

9. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *left;

```



```

    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    while(root != NULL){
        root = root->left;
    }
    printf("%d", root->data);
}

```

A.1

B.4

C.5

D.Lỗi thực thi

10. Chương trình sau cho kết quả là gì?

```

#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;

```

```

    struct node *left;
    struct node *right;
};

typedef struct node node;

node *taoNode(int x){
    node *res = (node*)malloc(sizeof(node));
    res->data = x;
    res->left = res->right = NULL;
    return res;
}

int main(){
    node *root = taoNode(1);
    node *x = taoNode(2), *y = taoNode(3), *z = taoNode(4), *t = taoNode(5);
    node *u = taoNode(6), *v = taoNode(7);
    root->left = x; root->right = y;
    x->left = z; x->right = t;
    y->left = u; y->right = v;
    while(root != NULL){
        if(root->data > 2) break;
        root = root->right;
    }
    printf("%d", root->data);
}

```

A.5

B.3

C.4

D.Lỗi thực thi