**CON TRỎ**

**Câu 1:** Biến int \*p là gì?  
A. Con trỏ trỏ tới kiểu char  
B. Con trỏ trỏ tới những biến thuộc kiểu int  
C. Biến lưu giá trị kiểu int  
D. Không hợp lệ  
**Đáp án:** B - Con trỏ p có kiểu dữ liệu là int. Con trỏ p này nó chỉ Trỏ được tới những biến có cùng kiểu dữ liệu (int)

**Giải thích:**

<Kiểu dữ liệu>: int

Dấu \*

Tên con trỏ: p

Làm thế nào để biết mình đang con khai báo 1 con trỏ: Là có dấu \*, và có kiểu dữ liệu.

Nhưng khi dấu \* đi với kiểu dữ liệu thì nó chỉ có tác dụng là Đang khai báo 1 con trỏ.

Tụi em đang hiểu sai:

Khi mà không có Kiểu dữ liệu. thì dấu \* sẽ là Lấy giá trị mà con trỏ p đang trỏ đến.

\*p

**Câu 2:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 10, \*p = &a;  printf("%d", \*p); // Lấy giá trị của a |

1. 10  
   B. Địa chỉ của a  
   C. Lỗi biên dịch  
   D. Không xác định  
   **Đáp án: A**

**Lưu ý**: Khi khai báo 1 con trỏ thì chúng ta phải ***ngay lập tức*** gán giá trị cho nó.

Mà giá trị của nó là: Địa chỉ của biến khác.

**Câu 3:** Tên của một mảng trong C là:  
A. Giá trị đầu tiên của mảng  
B. Kích thước mảng  
C. Địa chỉ của phần tử đầu tiên  
D. Không xác định

**Đáp án:** C

**Câu 4:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[3] = {1, 2, 3};  int \*p = arr; // Mà arr là **&**arr[0]  printf("%d", \*(**&**arr[0] + 1)); // **&**arr[1] |

A. 1  
B. 2  
C. 3  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 5:** Tại sao cần sử dụng con trỏ NULL?  
A. Để chỉ ra rằng con trỏ không trỏ tới đâu  
B. Để tăng hiệu suất chương trình  
C. Để tối ưu hóa bộ nhớ  
D. Để tránh lỗi logic  
**Đáp án:** A

**Câu 6:** Đoạn mã sau in ra gì?

|  |
| --- |
| int x = 5;  int \*ptr = &x;  \*ptr = 10;  printf("%d", x); |

A. 5  
B. 10  
C. Địa chỉ của x  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 7:** Phép toán nào cho phép lấy địa chỉ của một biến?  
A. \* : Nhân, dấu hiệu nhânj biết 1 con trỏ khi báo báo,   
B. & : Lấy địa chỉ  
C. sizeof : Lấy kích thước   
D. Không phép toán nào đúng  
**Đáp án:** B

**Câu 8:** Con trỏ không thể được sử dụng để:  
A. Truy cập địa chỉ của một biến  
B. Truy cập giá trị tại địa chỉ  
C. Thay đổi địa chỉ bộ nhớ vật lý  
D. Truy cập giá trị trong mảng  
**Đáp án:**

**Câu 9:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 5, b = 10;  int \*p = &a;  p: Địa chỉ của a  p = &b;  p: Địa chỉ của b  printf("%d", \*p); // Lấy giá trị của b |

A. 5  
B. 10  
C. Địa chỉ của b  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 10:** Đoạn mã sau có lỗi gì?

|  |
| --- |
| int \*p;  // Nó chưa được gán địa chỉ  printf("%d", \*p); |

A. Lỗi biên dịch  
B. Không lỗi  
C. Truy cập vào con trỏ không được khởi tạo  
D. Lỗi runtime

**Đáp án:** Lỗi logic.

**Câu 11:** Con trỏ là gì trong ngôn ngữ C?  
A. Một biến lưu giá trị của một biến khác  
B. Một biến lưu địa chỉ bộ nhớ của một biến khác  
C. Một biến không có giá trị  
D. Một kiểu dữ liệu đặc biệt  
**Đáp án:** B

**Câu 12:** Phép toán nào không thể thực hiện trên con trỏ?  
A. Phép cộng ( + )  
B. Phép trừ ( - )  
C. Phép chia  
D. Phép so sánh

**Đáp án:** C

**Câu 13:** Câu lệnh nào khai báo đúng một con trỏ trỏ đến một số nguyên?  
A. int \*ptr;  
B. int ptr\*;  
C. int &ptr;  
D. int ptr;  
**Đáp án:** A

**Câu 14:** Điều gì xảy ra khi bạn cố gắng truy cập một con trỏ chưa được khởi tạo?  
A. Lỗi biên dịch  
B. Lỗi runtime  
C. Giá trị không xác định  
D. Trả về giá trị 0

**Đáp án:**

**Câu 15:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 10, \*ptr = &x;  printf("%d", \*ptr); |

A. 10  
B. Địa chỉ của x  
C. Lỗi biên dịch  
D. Không xác định  
**Đáp án:** A

**Câu 16:** Con trỏ NULL trong C là gì?  
A. Con trỏ lưu giá trị bằng 0  
B. Con trỏ không trỏ đến bất kỳ địa chỉ nào  
C. Con trỏ trỏ đến địa chỉ của NULL  
D. Con trỏ không hợp lệ  
**Đáp án:**

**Câu 17:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int \*ptr = NULL;  printf("%d", \*ptr); |

A. 0  
B. NULL  
C. Lỗi runtime  
D. Giá trị không xác định  
**Đáp án:**

**Câu 18:** Con trỏ mảng được định nghĩa như thế nào?  
A. int \*arr[10];  
B. int (\*arr)[10];  
C. int arr[10];  
D. int arr\*;  
**Đáp án:**

**Câu 19:** Đoạn mã sau in ra giá trị gì?

|  |
| --- |
| int x = 5, y = 10;  int \*p = &x;  p = &y;  printf("%d", \*p); |

A. 5  
B. 10  
C. Địa chỉ của y  
D. Lỗi biên dịch  
**Đáp án:** B

**Câu 20:** Để truy cập phần tử thứ i của một mảng thông qua con trỏ, câu lệnh đúng là:  
A. \*(ptr + i)  
B. \*(i + ptr)  
C. ptr[i]  
D. Tất cả các đáp án trên đều đúng  
**Đáp án:** C

**Câu 21:** Đoạn mã sau in ra giá trị gì?

|  |
| --- |
| int arr[5] = {10, 20, 30, 40, 50};  int \*ptr = arr + 3; // arr[3]  printf("%d", \*(ptr - 1)); |

A. 20  
B. 30  
C. 40  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 22:** Con trỏ hàm được khai báo như thế nào?  
A. int (\*func)();  
B. int func\*();  
C. int \*func();  
D. int func();  
**Đáp án:**

**Câu 23:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| void increment(int \*ptr) {  (\*ptr)++;  }  int main() {  int x = 5;  increment(&x);  printf("%d", x);  } |

A. 5  
B. 6  
C. Lỗi biên dịch  
D. Giá trị không xác định  
**Đáp án:** B

**Câu 24:** Để giải phóng bộ nhớ cấp phát động, hàm nào được sử dụng?  
A. malloc()  
B. calloc()  
C. free()  
D. realloc()  
**Đáp án:**

**Câu 25:** Địa chỉ bộ nhớ trong C được biểu diễn bằng kiểu dữ liệu nào?  
A. int  
B. char  
C. void \*  
D. float  
**Đáp án:**

**Câu 26:** Con trỏ là gì trong ngôn ngữ C?  
A. Một biến lưu giá trị của một biến khác  
B. Một biến lưu địa chỉ bộ nhớ của một biến khác  
C. Một biến không có giá trị  
D. Một kiểu dữ liệu đặc biệt  
**Đáp án:**

**Câu 27:** Phép toán nào không thể thực hiện trên con trỏ?  
A. Phép cộng  
B. Phép trừ  
C. Phép chia  
D. Phép so sánh  
**Đáp án:**

**Câu 28:** Câu lệnh nào khai báo đúng một con trỏ trỏ đến một số nguyên?  
A. int \*ptr;  
B. int ptr\*;  
C. int &ptr;  
D. int ptr;  
**Đáp án:**

**Câu 29:** Điều gì xảy ra khi bạn cố gắng truy cập một con trỏ chưa được khởi tạo?  
A. Lỗi biên dịch  
B. Lỗi runtime  
C. Giá trị không xác định  
D. Trả về giá trị 0  
**Đáp án:**

**Câu 30:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 10, \*ptr = &x;  printf("%d", \*ptr); |

A. 10  
B. Địa chỉ của x  
C. Lỗi biên dịch  
D. Không xác định  
**Đáp án:**

**Câu 31:** Con trỏ NULL trong C là gì?  
A. Con trỏ lưu giá trị bằng 0  
B. Con trỏ không trỏ đến bất kỳ địa chỉ nào  
C. Con trỏ trỏ đến địa chỉ của NULL  
D. Con trỏ không hợp lệ  
**Đáp án:**

**Câu 32:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int \*ptr = NULL;  printf("%d", \*ptr); |

A. 0  
B. NULL  
C. Lỗi runtime  
D. Giá trị không xác định  
**Đáp án:**

**Câu 33:** Con trỏ mảng được định nghĩa như thế nào?  
A. int \*arr[10];  
B. int (\*arr)[10];  
C. int arr[10];  
D. int arr\*;  
**Đáp án:**

**Câu 34:** Đoạn mã sau in ra giá trị gì?

|  |
| --- |
| int x = 5, y = 10;  int \*p = &x;  p = &y;  printf("%d", \*p); |

A. 5  
B. 10  
C. Địa chỉ của y  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 35:** Để truy cập phần tử thứ i của một mảng thông qua con trỏ, câu lệnh đúng là:  
A. \*(ptr + i)  
B. \*(i + ptr)  
C. ptr[i]  
D. Tất cả các đáp án trên đều đúng  
**Đáp án:**

**Câu 36:** Lệnh nào cấp phát động bộ nhớ trong C?  
A. malloc()  
B. calloc()  
C. realloc()  
D. Tất cả các đáp án trên  
**Đáp án:**

**Câu 37:** Đoạn mã sau in ra giá trị gì?

|  |
| --- |
| int a = 10, \*ptr = &a;  printf("%p", ptr); |

A. Địa chỉ của a  
B. Giá trị của a  
C. Lỗi biên dịch  
D. Không xác định  
**Đáp án:**

**Câu 38:** Hàm free() có tác dụng gì?  
A. Giải phóng con trỏ  
B. Giải phóng bộ nhớ đã được cấp phát động  
C. Xóa giá trị của con trỏ  
D. Xóa tất cả các biến  
**Đáp án:**

**Câu 39:** Để thay đổi nội dung của một biến trong hàm bằng cách truyền tham chiếu, cần sử dụng:  
A. Địa chỉ biến  
B. Giá trị biến  
C. Con trỏ  
D. Cả A và C đều đúng  
**Đáp án:**

**Câu 40:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {10, 20, 30};  int \*p = arr;  printf("%d", \*(p + 2)); |

A. 10  
B. 20  
C. 30  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 41:** Đoạn mã sau in ra giá trị gì?

|  |
| --- |
| int arr[5] = {10, 20, 30, 40, 50};  int \*ptr = arr + 3;  printf("%d", \*(ptr - 1)); |

A. 20  
B. 30  
C. 40  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 42:** Con trỏ hàm được khai báo như thế nào?  
A. int (\*func)();  
B. int func\*();  
C. int \*func();  
D. int func();  
**Đáp án:**

**Câu 43:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| void increment(int \*ptr) {  (\*ptr)++;  }  int main() {  int x = 5;  increment(&x);  printf("%d", x);  } |

A. 5  
B. 6  
C. Lỗi biên dịch  
D. Giá trị không xác định  
**Đáp án:**

**Câu 44:** Để giải phóng bộ nhớ cấp phát động, hàm nào được sử dụng?  
A. malloc()  
B. calloc()  
C. free()  
D. realloc()  
**Đáp án:**

**Câu 45:** Địa chỉ bộ nhớ trong C được biểu diễn bằng kiểu dữ liệu nào?  
A. int  
B. char  
C. void \*  
D. float  
**Đáp án:**

**Câu 46:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 5, b = 10;  int \*p1 = &a, \*p2 = &b;  printf("%d", \*p1 + \*p2); |

A. 10  
B. 15  
C. 20  
D. Lỗi runtime  
**Đáp án:**

**Câu 47:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {1, 2, 3, 4, 5};  int \*ptr = arr + 4;  printf("%d", \*ptr - \*(ptr - 2)); |

A. 1  
B. 2  
C. 3  
D. Lỗi runtime  
**Đáp án:**

**Câu 48:** Điều gì xảy ra khi bạn gọi free() trên một con trỏ NULL?  
A. Không làm gì cả  
B. Gây lỗi runtime  
C. Làm rỗng giá trị con trỏ  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 49:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| char str[] = "Hello";  char \*ptr = str;  printf("%c", \*(ptr + 1)); |

A. H  
B. e  
C. l  
D. o  
**Đáp án:**

**Câu 50:** Nếu một con trỏ có giá trị là 0x100, sau khi thực hiện ptr = ptr + 1, địa chỉ của ptr là gì?  
(giả sử ptr là kiểu int \* và sizeof(int) = 4)  
A. 0x101  
B. 0x104  
C. 0x108  
D. 0x100  
**Đáp án:**

**Câu 51:** Đoạn mã sau in ra gì?

|  |
| --- |
| int x = 10, y = 20;  int \*ptr = &x;  \*ptr = \*ptr + y;  printf("%d", x); |

A. 10  
B. 20  
C. 30  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 52:** Một con trỏ void (void \*) có thể được sử dụng như thế nào?  
A. Trỏ đến bất kỳ kiểu dữ liệu nào  
B. Thực hiện các phép toán trực tiếp như cộng, trừ  
C. Truy cập trực tiếp giá trị mà nó trỏ đến  
D. Cả A và B  
**Đáp án:**

**Câu 53:** Để truyền một mảng cho hàm, câu lệnh nào sau đây đúng?  
A. func(int arr[]);  
B. func(int \*arr);  
C. func(int arr[10]);  
D. Tất cả các đáp án trên đều đúng  
**Đáp án:**

**Câu 54:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 5;  int \*ptr1 = &a, \*\*ptr2 = &ptr1;  \*\*ptr2 = 10;  printf("%d", a); |

A. 5  
B. 10  
C. Địa chỉ của a  
D. Lỗi runtime  
**Đáp án:**

**Câu 55:** Đoạn mã sau in ra gì?

|  |
| --- |
| int arr[3] = {10, 20, 30};  int \*p = arr;  printf("%d %d", \*p++, \*p); |

A. 10 20  
B. 20 20  
C. 20 30  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 56:** Một con trỏ được trả về bởi malloc() sẽ có giá trị nào nếu không cấp phát thành công?  
A. 0  
B. NULL  
C. -1  
D. Giá trị không xác định  
**Đáp án:**

**Câu 57:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a[2][3] = {{1, 2, 3}, {4, 5, 6}};  int (\*ptr)[3] = a;  printf("%d", \*(\*(ptr + 1) + 2)); |

A. 3  
B. 4  
C. 5  
D. 6  
**Đáp án:**

**Câu 58:** Đoạn mã sau sẽ in ra gì?

|  |
| --- |
| int x = 5, y = 10;  int \*p1 = &x, \*p2 = &y;  printf("%d", (\*p1 > \*p2) ? \*p1 : \*p2); |

A. 5  
B. 10  
C. Địa chỉ của x  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 59:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 10, \*ptr = &x;  ptr = NULL;  printf("%p", ptr); |

A. Địa chỉ của x  
B. 10  
C. 0 (NULL)  
D. Lỗi runtime  
**Đáp án:**

**Câu 60:** Con trỏ const là gì?  
A. Con trỏ trỏ đến một giá trị không đổi  
B. Con trỏ có giá trị không đổi  
C. Con trỏ không thể thay đổi nơi nó trỏ đến  
D. Con trỏ không tồn tại  
**Đáp án:**

**Câu 61:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {10, 20, 30};  int \*ptr = arr;  \*(ptr + 1) = 50;  printf("%d", arr[1]); |

A. 20  
B. 30  
C. 50  
D. Lỗi runtime  
**Đáp án:**

**Câu 62:** Cách khai báo đúng một con trỏ hằng là gì?  
A. int const \*ptr;  
B. int \*const ptr;  
C. const int \*ptr;  
D. Tất cả các đáp án trên đều đúng  
**Đáp án:**

**Câu 63:** Khi một con trỏ được truyền vào hàm, điều gì sẽ xảy ra?  
A. Giá trị con trỏ bị sao chép  
B. Địa chỉ mà con trỏ trỏ đến bị thay đổi  
C. Cả A và B đều đúng tùy trường hợp  
D. Không có gì xảy ra  
**Đáp án:**

**Câu 64:** Một con trỏ trỏ đến một con trỏ khác được gọi là gì?  
A. Con trỏ cấp hai  
B. Con trỏ hàm  
C. Con trỏ đa cấp  
D. Cả A và C  
**Đáp án:**

**Câu 65:** Điều gì xảy ra nếu bạn giải phóng cùng một con trỏ hai lần?  
A. Không có gì xảy ra  
B. Lỗi runtime (undefined behavior)  
C. Con trỏ sẽ trở thành NULL  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 66:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| int x = 10;  int \*p1 = &x, \*\*p2 = &p1;  printf("%d", \*\*p2); |

A. 10  
B. Địa chỉ của x  
C. Địa chỉ của p1  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 67:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[5] = {1, 2, 3, 4, 5};  int \*p = arr + 2;  printf("%d", \*p); |

A. 1  
B. 2  
C. 3  
D. Lỗi runtime  
**Đáp án:**

**Câu 68:** Con trỏ void \* có thể:  
A. Trỏ tới mọi kiểu dữ liệu  
B. Chỉ trỏ tới kiểu void  
C. Không sử dụng được  
D. Lưu giá trị kiểu int  
**Đáp án:**

**Câu 69:** Đoạn mã sau in ra gì?

|  |
| --- |
| int arr[3] = {1, 2, 3};  int \*p = arr;  printf("%p", p + 1); |

A. Địa chỉ của phần tử thứ nhất  
B. Địa chỉ của phần tử thứ hai  
C. Giá trị của phần tử thứ hai  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 70:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 5;  int \*p = &a;  int \*\*pp = &p;  printf("%d", \*(\*pp)); |

A. Địa chỉ của a  
B. 5  
C. Địa chỉ của p  
D. Lỗi runtime  
**Đáp án:**

**Câu 71:** Đoạn mã sau có lỗi gì?

|  |
| --- |
| int \*p = (int \*)malloc(sizeof(int));  \*p = 10;  free(p);  printf("%d", \*p); |

A. Lỗi biên dịch  
B. Không lỗi, kết quả là 10  
C. Lỗi runtime do sử dụng con trỏ bị giải phóng  
D. Lỗi do không ép kiểu khi dùng malloc  
**Đáp án:**

**Câu 72:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| char str[] = "Hello";  char \*p = str;  printf("%c", \*(p + 1)); |

A. H  
B. e  
C. l  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 73:** Địa chỉ mà một con trỏ NULL trỏ tới là:  
A. Địa chỉ hợp lệ trong bộ nhớ  
B. Địa chỉ 0  
C. Địa chỉ không xác định  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 74:** Đoạn mã sau in ra gì?

|  |
| --- |
| int arr[5] = {10, 20, 30, 40, 50};  int \*p = arr;  p += 3;  printf("%d", \*p); |

A. 10  
B. 30  
C. 40  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 75:** Khi nào cần sử dụng free trong lập trình con trỏ?  
A. Khi sử dụng malloc để cấp phát bộ nhớ  
B. Khi sử dụng con trỏ tới mảng  
C. Khi con trỏ trỏ tới biến toàn cục  
D. Khi sử dụng con trỏ NULL  
**Đáp án:**

**Câu 76:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {1, 2, 3};  int \*p1 = arr, \*p2 = arr + 2;  printf("%ld", p2 - p1); |

A. 2  
B. 1  
C. 0  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 77:** Con trỏ hàm dùng để:  
A. Truyền tham số kiểu hàm vào hàm khác  
B. Lưu địa chỉ của một hàm  
C. Gọi hàm thông qua con trỏ  
D. Tất cả các ý trên  
**Đáp án:**

**Câu 78:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 100;  int \*p = &x;  \*p = \*p + 1;  printf("%d", x); |

A. 100  
B. 101  
C. Địa chỉ của x  
D. Lỗi runtime  
**Đáp án:**

**Câu 79:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| void func(int \*\*p) {  \*\*p = 20;  }  int main() {  int a = 10;  int \*p = &a;  func(&p);  printf("%d", a);  return 0;  } |

A. 10  
B. 20  
C. Địa chỉ của a  
D. Lỗi runtime  
**Đáp án:**

**Câu 80:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 10;  int \*p1 = &x, \*p2 = p1;  \*p2 = 20;  printf("%d", x); |

A. 10  
B. 20  
C. Địa chỉ của x  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 81:** Đoạn mã sau in ra gì?

|  |
| --- |
| int a = 5, b = 15;  int \*p = &a;  \*p = b;  printf("%d %d", a, b); |

A. 5 15  
B. 15 15  
C. 15 5  
D. Lỗi runtime  
**Đáp án:**

**Câu 82:** Khi truyền một con trỏ vào hàm, giá trị mà con trỏ trỏ tới có thể:  
A. Thay đổi bên trong hàm  
B. Không thay đổi  
C. Phụ thuộc vào giá trị của tham số  
D. Không xác định  
**Đáp án:**

**Câu 83:** Điều gì xảy ra khi không giải phóng bộ nhớ cấp phát bằng malloc?  
A. Bộ nhớ bị rò rỉ (memory leak)  
B. Lỗi biên dịch  
C. Lỗi runtime  
D. Bộ nhớ tự động giải phóng  
**Đáp án:**

**Câu 84:** Con trỏ void \* cần phải làm gì trước khi sử dụng?  
A. Ép kiểu về kiểu dữ liệu cụ thể  
B. Sử dụng trực tiếp  
C. Không thể sử dụng  
D. Chỉ sử dụng trong malloc  
**Đáp án:**

**Câu 85:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {1, 2, 3, 4};  int \*p = arr + 1;  printf("%d", \*(p - 1)); |

A. 1  
B. 2  
C. 0  
D. Lỗi runtime  
**Đáp án:**

**Câu 86:** Đoạn mã sau có lỗi gì?

|  |
| --- |
| int \*p = NULL;  \*p = 5; |

A. Lỗi biên dịch  
B. Không lỗi  
C. Lỗi runtime do truy cập con trỏ NULL  
D. Không xác định  
**Đáp án:**

**Câu 87:** Đoạn mã sau in ra gì?

|  |
| --- |
| char \*str = "Hello";  printf("%c", \*(str + 4)); |

A. H  
B. o  
C. e  
D. Lỗi runtime  
**Đáp án:**

**Câu 88:** Đoạn mã sau có lỗi gì?

|  |
| --- |
| int arr[5] = {1, 2, 3, 4, 5};  int \*p = arr + 5;  printf("%d", \*p); |

A. Lỗi biên dịch  
B. Không lỗi, nhưng hành vi không xác định  
C. Lỗi runtime do truy cập ngoài mảng  
D. Không lỗi, in ra 0  
**Đáp án:**

**Câu 89:** Con trỏ hàm được khai báo như thế nào?  
A. int (\*fptr)(int, int);  
B. int \*fptr(int, int);  
C. int fptr(\*int, \*int);  
D. void \*fptr();  
**Đáp án:**

**Câu 90:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 10;  int \*p = &x;  int \*\*pp = &p;  \*\*pp = 15;  printf("%d", x); |

A. 10  
B. 15  
C. Địa chỉ của x  
D. Lỗi runtime  
**Đáp án:**

**Câu 91:** Điều gì xảy ra nếu bạn sử dụng một con trỏ không hợp lệ để truy cập dữ liệu?  
A. Lỗi biên dịch  
B. Lỗi runtime hoặc hành vi không xác định (undefined behavior)  
C. Giá trị mặc định là 0  
D. Lỗi logic  
**Đáp án:**

**Câu 92:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[] = {10, 20, 30, 40};  int \*ptr = arr + 1;  printf("%d", \*(ptr++)); |

A. 10  
B. 20  
C. 30  
D. Lỗi runtime  
**Đáp án:**

**Câu 93:** Đoạn mã sau in ra gì?

|  |
| --- |
| int x = 5, y = 10, \*ptr1 = &x, \*ptr2 = &y;  \*ptr1 = \*ptr2;  printf("%d %d", x, y); |

A. 5 10  
B. 10 10  
C. 5 5  
D. Lỗi runtime  
**Đáp án:**

**Câu 94:** Để khai báo một con trỏ hàm nhận vào hai tham số kiểu int và trả về kiểu float, khai báo nào sau đây là đúng?  
A. float \*func(int, int);  
B. float (\*func)(int, int);  
C. float \*(func)(int, int);  
D. int (\*func)(float, float);  
**Đáp án:**

**Câu 95:** Điều gì xảy ra nếu bạn truy cập ptr[-1] trong trường hợp ptr trỏ đến phần tử đầu tiên của mảng?  
A. Giá trị không xác định (undefined behavior)  
B. Truy cập phần tử cuối cùng của mảng  
C. Lỗi biên dịch  
D. Trả về giá trị 0  
**Đáp án:**

**Câu 96:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int a = 5;  int \*ptr = &a;  int \*\*pptr = &ptr;  \*\*pptr += 10;  printf("%d", a); |

A. 5  
B. 10  
C. 15  
D. Lỗi runtime  
**Đáp án:**

**Câu 97:** Đoạn mã sau sẽ in ra gì?

|  |
| --- |
| int a = 10, b = 20;  int \*p1 = &a, \*p2 = &b;  p1 = p2;  printf("%d", \*p1); |

A. 10  
B. 20  
C. Địa chỉ của b  
D. Lỗi biên dịch  
**Đáp án:**

**Câu 98:** Đoạn mã sau in ra gì?

|  |
| --- |
| int arr[5] = {10, 20, 30, 40, 50};  int \*p = arr;  printf("%d", \*(p + 3)); |

A. 10  
B. 20  
C. 40  
D. Lỗi runtime  
**Đáp án:**

**Câu 99:** Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int x = 5, y = 10, z = 15;  int \*arr[3] = {&x, &y, &z};  printf("%d", \*arr[1]); |

A. 5  
B. 10  
C. 15  
D. Lỗi runtime  
**Đáp án:**

**Câu 100:** Đoạn mã sau có kết quả gì?

|  |
| --- |
| int x = 5;  int \*p = &x;  free(p);  printf("%d", x); |

A. 5  
B. Giá trị không xác định (undefined behavior)  
C. 0  
D. Lỗi runtime  
**Đáp án:**