**Phần 1: Con trỏ**

1. Giá trị ban đầu của một con trỏ không được khởi tạo là gì?
   * A. NULL
   * B. 0
   * C. Giá trị rác (garbage value)
   * D. Không có giá trị nào

**Đáp án**:

1. Địa chỉ nào sẽ được lưu trữ trong con trỏ p sau đoạn mã sau?

|  |
| --- |
| int x = 10;  int \*p = &x; |

* + A. Địa chỉ của biến x
  + B. Giá trị của biến x
  + C. NULL
  + D. Giá trị rác

**Đáp án**:

1. Con trỏ NULL là gì?
   * A. Một con trỏ trỏ đến địa chỉ 0
   * B. Một con trỏ có giá trị rác
   * C. Một con trỏ trỏ đến biến toàn cục
   * D. Một con trỏ không được khởi tạo

**Đáp án**:

1. Điều gì xảy ra nếu truy cập vào một con trỏ NULL?
   * A. Chương trình chạy bình thường
   * B. Truy cập vào một địa chỉ không xác định
   * C. Lỗi runtime (runtime error)
   * D. Lỗi biên dịch (compile-time error)

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

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

* + A. 10
  + B. 20
  + C. Lỗi biên dịch
  + D. Giá trị không xác định

**Đáp án**:

1. Một con trỏ void (void pointer) có thể:
   * A. Trỏ đến bất kỳ kiểu dữ liệu nào
   * B. Chỉ trỏ đến kiểu dữ liệu void
   * C. Không thể sử dụng trong toán tử dereference (\*)
   * D. Cả A và C

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

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

* + A. 5 10
  + B. 10 10
  + C. 5 5
  + D. Lỗi biên dịch

**Đáp án**:

1. Con trỏ hàm là gì?
   * A. Con trỏ trỏ đến địa chỉ của hàm
   * B. Con trỏ trỏ đến địa chỉ của biến
   * C. Con trỏ trỏ đến kiểu void
   * D. Cả A và B

**Đáp án**:

1. Đoạn mã sau đây sẽ in gì?

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

* + A. 10
  + B. 20
  + C. 30
  + D. Lỗi runtime

**Đáp án**:

1. Đâu là cú pháp đúng để cấp phát bộ nhớ động trong C?
   * A. malloc(sizeof(int))
   * B. malloc(int)
   * C. new int
   * D. alloc(sizeof(int))

**Đáp án**:

**Phần 2: Mảng**

1. Mảng trong C bắt đầu từ chỉ số nào?
   * A. -1
   * B. 0
   * C. 1
   * D. Không xác định

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

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

* + A. 1
  + B. 2
  + C. 4
  + D. 5

**Đáp án**:

1. Trong C, mảng có thể thay đổi kích thước sau khi khai báo không?
   * A. Có
   * B. Không
   * C. Tùy thuộc vào compiler
   * D. Chỉ với kiểu char

**Đáp án**:

1. Điều gì xảy ra khi truy cập phần tử ngoài phạm vi của mảng?
   * A. Lỗi runtime
   * B. Lỗi compile-time
   * C. Truy cập giá trị không xác định
   * D. Lỗi segmentation fault

**Đáp án**:

1. Địa chỉ của phần tử arr[0] và arr khác nhau như thế nào?
   * A. Giống nhau
   * B. Khác nhau
   * C. arr là địa chỉ của con trỏ, arr[0] là giá trị đầu tiên
   * D. Không xác định

**Đáp án**:

1. Kích thước của mảng int arr[10] là bao nhiêu?
   * A. 10 bytes
   * B. 20 bytes
   * C. 40 bytes (nếu int = 4 bytes)
   * D. Phụ thuộc vào hệ điều hành

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| int arr[3] = {10, 20};  printf("%d", arr[2]); |

* + A. 0
  + B. 10
  + C. 20
  + D. Giá trị không xác định

**Đáp án**:

1. Mảng một chiều int arr[5] có thể được duyệt bằng cách nào?
   * A. Vòng lặp for
   * B. Vòng lặp while
   * C. Sử dụng con trỏ
   * D. Tất cả các cách trên

**Đáp án**:

1. Đoạn mã sau đây sẽ in gì?

|  |
| --- |
| int arr[] = {10, 20, 30, 40};  printf("%d", sizeof(arr) / sizeof(arr[0])); |

* + A. 4
  + B. 10
  + C. 16
  + D. Lỗi compile-time

**Đáp án**:

1. Một mảng hai chiều có thể được khai báo như thế nào?
   * A. int arr[][];
   * B. int arr[5][5];
   * C. int arr[5][];
   * D. Không thể khai báo mảng hai chiều

**Đáp án**:

**Phần 3: Chuỗi**

1. Chuỗi trong C được kết thúc bằng ký tự nào?
   * A. \n
   * B. \0
   * C. NULL
   * D. Cả B và C

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

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

* + A. H
  + B. e
  + C. l
  + D. o

**Đáp án**:

1. Chuỗi có thể được gán trực tiếp như sau không?

|  |
| --- |
| char str[10];  str = "Hello"; |

* + A. Có
  + B. Không

**Đáp án**:

1. Hàm nào được sử dụng để nối hai chuỗi trong C?
   * A. strcopy
   * B. strcat
   * C. strlen
   * D. strncat

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| char str1[10] = "Hello";  char str2[10];  strcpy(str2, str1);  printf("%s", str2); |

* + A. Hello
  + B. Lỗi runtime
  + C. Lỗi compile-time
  + D. Giá trị không xác định

**Đáp án**:

1. Kích thước của chuỗi "Hello" là gì?
   * A. 5
   * B. 6
   * C. 7
   * D. 10

**Đáp án**:

1. Hàm nào trả về độ dài của chuỗi?
   * A. strlen
   * B. strcat
   * C. strcmp
   * D. strcpy

**Đáp án**:

1. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| char str[10] = "Hello";  str[0] = 'Y';  printf("%s", str); |

* + A. Hello
  + B. Yello
  + C. Y
  + D. Lỗi compile-time

**Đáp án**:

1. Chuỗi char str[5] = "Hello"; sẽ gây ra vấn đề gì?
   * A. Không có vấn đề
   * B. Lỗi runtime do thiếu không gian cho \0
   * C. Lỗi compile-time
   * D. Giá trị không xác định

**Đáp án**:

1. Hàm nào để so sánh hai chuỗi trong C?
   * A. strcmp
   * B. strcat
   * C. strlen
   * D. strcpy

**Đáp án**:

31. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  void test(int \*p) {  \*p = 20;  }  int main() {  int a = 10;  test(&a);  printf("%d", a);  return 0;  } |

* A. 10
* B. 20
* C. Lỗi biên dịch
* D. Lỗi runtime

**Đáp án**:

32. Con trỏ có thể được sử dụng để:

* Thay đổi giá trị của một biến thông qua địa chỉ của nó
* Duyệt qua các phần tử của mảng
* Tạo mối quan hệ giữa các hàm thông qua con trỏ hàm
* Tất cả các ý trên

**Đáp án**:

33. Đoạn mã sau in ra gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int a = 10;  int \*p = &a;  int \*\*pp = &p;  \*\*pp = 20;  printf("%d", a);  return 0;  } |

* 10
* 20
* Lỗi biên dịch
* Giá trị không xác định

**Đáp án**:

34. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int arr[] = {1, 2, 3, 4, 5};  int \*p = arr;  printf("%d", \*(p + 3));  return 0;  } |

* 1
* 2
* 4
* Lỗi runtime

**Đáp án**:

35. Giá trị của \*p sau đoạn mã này là gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int a = 10;  int \*p;  p = &a;  \*p = \*p + 5;  printf("%d", \*p);  return 0;  } |

* A. 10
* B. 15
* C. Giá trị không xác định
* D. Lỗi biên dịch

**Đáp án**:

36. Đâu là cách đúng để truy cập con trỏ đa cấp (multi-level pointer)?

* A. int \*\*\*p trỏ tới int \*\*p
* B. int \*p trỏ tới int
* C. int \*\*p trỏ tới int \*
* D. Cả A và C

**Đáp án**:

37. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  void test(int \*\*p) {  \*\*p = 30;  }  int main() {  int a = 10;  int \*ptr = &a;  test(&ptr);  printf("%d", a);  return 0;  } |

* A. 10
* B. 30
* C. Lỗi biên dịch
* D. Lỗi runtime

**Đáp án**:

38. Kết quả của đoạn mã này là gì?

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int main() {  int \*p = (int \*)malloc(sizeof(int) \* 3);  p[0] = 1; p[1] = 2; p[2] = 3;  printf("%d", \*(p + 1));  free(p);  return 0;  } |

* A. 1
* B. 2
* C. 3
* D. Lỗi runtime

**Đáp án**:

39. Con trỏ void có thể sử dụng toán tử dereference (\*) trực tiếp không?

* A. Có
* B. Không
* C. Chỉ trong một số trường hợp
* D. Không xác định

**Đáp án**:

40. Điều gì xảy ra khi giải phóng bộ nhớ đã được cấp phát động nhiều lần?

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  int main() {  int \*p = (int \*)malloc(sizeof(int));  free(p);  free(p);  return 0;  } |

* A. Lỗi biên dịch
* B. Lỗi runtime
* C. Không xảy ra lỗi (undefined behavior)
* D. Bộ nhớ bị giải phóng nhiều lần

**Đáp án**:

41. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int a = 5;  int \*p = &a;  int \*\*pp = &p;  printf("%d", \*\*pp);  return 0;  } |

* A. 5
* B. Địa chỉ của p
* C. Địa chỉ của a
* D. Giá trị không xác định

**Đáp án**:

42. Con trỏ hàm được khai báo đúng như thế nào?

* A. int (\*ptr)(int);
* B. int \*ptr(int);
* C. int \*(\*ptr)(int);
* D. int (ptr)(int);

**Đáp án**:

43. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  void swap(int \*a, int \*b) {  int temp = \*a;  \*a = \*b;  \*b = temp;  }  int main() {  int x = 5, y = 10;  swap(&x, &y);  printf("%d %d", x, y);  return 0;  } |

* A. 5 10
* B. 10 5
* C. Giá trị không xác định
* D. Lỗi biên dịch

**Đáp án**:

44. Con trỏ void có thể được ép kiểu như thế nào?

* A. int \*p = (int \*)void\_ptr;
* B. void \*p = (void \*)int\_ptr;
* C. Không cần ép kiểu
* D. Cả A và B

**Đáp án**:

45. Điều gì xảy ra khi cố truy cập vào con trỏ không được khởi tạo?

* A. Giá trị rác được trả về
* B. Lỗi runtime
* C. Undefined behavior
* D. Lỗi compile-time

**Đáp án**:

46. Kết quả của đoạn mã sau là gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int arr[3] = {1, 2, 3};  int \*p = arr + 2;  printf("%d", \*(p - 1));  return 0;  } |

* A. 1
* B. 2
* C. 3
* D. Lỗi runtime

**Đáp án**:

47. Địa chỉ được lưu trong con trỏ sau đoạn mã này là gì?

|  |
| --- |
| #include <stdio.h>  int main() {  int a = 10;  int \*p = &a;  p++;  printf("%p", (void \*)p);  return 0;  } |

* A. Địa chỉ của a
* B. Địa chỉ tăng thêm 4 bytes (hoặc kích thước của int)
* C. Giá trị không xác định
* D. Lỗi runtime

**Đáp án**:

48. Con trỏ hàm có thể được sử dụng như thế nào?

* A. Để gọi hàm thông qua con trỏ
* B. Để lưu địa chỉ của một hàm
* C. Cả A và B
* D. Không thể sử dụng

**Đáp án**:

49. Đoạn mã sau sẽ in gì?

|  |
| --- |
| #include <stdio.h>  void increment(int \*p) {  (\*p)++;  }  int main() {  int a = 5;  increment(&a);  printf("%d", a);  return 0;  } |

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

**Đáp án**:

50. Điều gì xảy ra khi con trỏ trỏ tới một vùng bộ nhớ đã bị giải phóng?

* A. Giá trị rác được trả về
* B. Undefined behavior
* C. Lỗi runtime
* D. Lỗi compile-time

**Đáp án**: