1) What is the output of the following code snippet?

#include <stdio.h>

struct test {

int x;

char y;

float z;

};

int main() {

struct test t = {10, 'A', 3.14};

struct test \*ptr = &t;

printf("%d", sizeof(\*ptr));

return 0;

}

**a) 4**

b) 8

c) 9

d) Compiler-dependent

2) In C, a nested structure is a structure \_\_\_\_\_\_.

**a) Defined inside another structure**

b) Defined outside the main function

c) Defined inside the standard library

d) Defined inside a loop

3) Which of the following is true regarding nested structures in C?

a) Nested structures cannot be passed as function arguments

b) Nested structures can only contain basic data types

c) Nested structures can only be defined inside the main function

**d) Nested structures can be defined within other nested structures**

4) Consider the following code snippet:

struct outer {

int x;

struct inner {

int y;

} in;

};

Which of the following statements correctly accesses the variable y?

a) outer.y

**b) outer.in.y**

c) in.y

d) y

5) Which of the following statements is true regarding structure pointers in C?

a) Structure pointers cannot be assigned to structures

b) Structure pointers cannot be used with the arrow operator (->)

**c) Structure pointers are used to access members of a structure**

d) Structure pointers can only point to the first member of a structure

6) Consider the following code snippet:

struct person {

char name[20];

int age;

};

struct person \*ptr;

Which of the following statements correctly assigns a value to the name member using the structure pointer ptr?

a) ptr->name = "John";

b) \*ptr.name = "John";

c) ptr.name = "John";

**d) strcpy(ptr->name, "John");**

7) Consider the following code snippet:

struct student {

int roll;

struct {

char name[20];

int age;

} info;

};

struct student s = {1, {"John", 20}};

Which of the following statements correctly accesses the age member of the info structure?

**a) s.info.age**

b) s->info.age

c) s.info->age

d) s->info->age

8) What is the output of the following code snippet?

#include <stdio.h>

struct date {

int day;

int month;

int year;

};

int main() {

struct date \*ptr;

struct date d = {10, 6, 2022};

ptr = &d;

printf("%d/%d/%d", (\*ptr).day, (\*ptr).month, (\*ptr).year);

return 0;

}

**a) 10/6/2022**

b) 6/10/2022

c) 2022/6/10

d) Compiler Error

9) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

void modify(struct point \*ptr) {

ptr->x = 5;

ptr->y = 10;

}

int main() {

struct point p = {0, 0};

modify(&p);

printf("%d %d", p.x, p.y);

return 0;

}

a) 0 0

**b) 5 10**

c) 0 10

d) 5 0

10) What is the output of the following code snippet?

#include <stdio.h>

struct rectangle {

int length;

int width;

};

int main() {

struct rectangle r = {5, 10};

struct rectangle \*ptr = &r;

printf("%d", ptr->length \* ptr->width);

return 0;

}

**a) 50**

b) 15

c) 0

d) Compiler Error

11) What is the output of the following code snippet?

#include <stdio.h>

struct student {

char name[20];

int age;

};

void printInfo(struct student \*ptr) {

printf("%s %d", ptr->name, ptr->age);

}

int main() {

struct student s = {"Alice", 25};

printInfo(&s);

return 0;

}

**a) Alice 25**

b) Random Value Random Value

c) Compiler Error

d) None of the above

12) What is the output of the following code snippet?

#include <stdio.h>

struct person {

char name[20];

int age;

};

void changeAge(struct person \*ptr) {

ptr->age = 30;

}

int main() {

struct person p = {"John", 25};

changeAge(&p);

printf("%d", p.age);

return 0;

}

a) 25

**b) 30**

c) Compiler Error

d) None of the above

13) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

typedef struct point Point;

int main() {

Point p = {5, 10};

printf("%d %d", p.x, p.y);

return 0;

}

**a) 5 10**

b) 10 5

c) Compiler Error

d) None of the above

14) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

typedef struct point Point;

int main() {

Point \*ptr;

ptr->x = 5;

ptr->y = 10;

printf("%d %d", ptr->x, ptr->y);

return 0;

}

a) 5 10

b) 10 5

**c) Compiler Error**

d) None of the above

15) What is the output of the following code snippet?

#include <stdio.h>

struct employee {

char name[20];

int age;

};

void printName(struct employee \*ptr) {

printf("%s", ptr->name);

}

int main() {

struct employee e = {"John Doe", 30};

printName(&e);

return 0;

}

**a) John Doe**

b) Random Value

c) Compiler Error

d) None of the above

16) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

int main() {

struct point \*ptr;

struct point p = {5, 10};

ptr = &p;

printf("%d %d", ptr->x, ptr->y);

return 0;

}

**a) 5 10**

b) 10 5

c) Compiler Error

d) None of the above

17) What is the output of the following code snippet?

#include <stdio.h>

struct person {

char name[20];

int age;

};

int main() {

struct person p = {"Alice", 25};

printf("%s", p.name);

return 0;

}

**a) Alice**

b) Random Value

c) Compiler Error

d) None of the above

18) What is the output of the following code snippet?

#include <stdio.h>

struct student {

int roll;

char name[20];

};

int main() {

struct student s = {1, "John"};

struct student \*ptr = &s;

printf("%s", ptr->name);

return 0;

}

**a) John**

b) Random Value

c) Compiler Error

d) None of the above

19) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

void update(struct point \*ptr) {

ptr->x = 10;

ptr->y = 20;

}

int main() {

struct point p = {5, 5};

update(&p);

printf("%d %d", p.x, p.y);

return 0;

}

a) 5 5

**b) 10 20**

c) 20 10

d) Compiler Error

20) What is the output of the following code snippet?

#include <stdio.h>

struct point {

int x;

int y;

};

void change(struct point p) {

p.x = 10;

p.y = 20;

}

int main() {

struct point p = {5, 5};

change(p);

printf("%d %d", p.x, p.y);

return 0;

}

**a) 5 5**

b) 10 20

c) 20 10

d) Compiler Error

21) What is the output of the following code snippet?

#include <stdio.h>

struct person {

char name[20];

int age;

};

int main() {

struct person \*ptr;

struct person p = {"John Doe", 30};

ptr = &p;

printf("%s", (\*ptr).name);

return 0;

}

**a) John Doe**

b) Random Value

c) Compiler Error

d) None of the above

22) What is the correct way to access a member of a structure using a structure pointer in C?

a) structureName.memberName

**b) pointerName->memberName**

c) pointerName.memberName

d) structureName->memberName

23) What is the purpose of using structure pointers in C?

a) To dynamically allocate memory for structures

b) To efficiently access and modify structure members

c) To pass structures as function arguments by reference

**d) All of the above**

24) Which of the following statements is true about structure pointers in C?

a) Structure pointers cannot be dereferenced

**b) Structure pointers always point to valid memory locations**

c) Structure pointers can only be used with arrays of structures

d) Structure pointers are automatically initialized to NULL

25) Which of the following is the correct way to allocate memory for a structure pointer in C?

a) structName \*pointerName = malloc(sizeof(structName));

**b) structName \*pointerName = (structName \*)malloc(sizeof(structName));**

c) structName \*pointerName = new structName;

d) structName \*pointerName = allocate(structName);

26) What does the following declaration represent?

struct student \*ptr;

a) Declaration of a structure variable

**b) Declaration of a pointer to a structure**

c) Declaration of a structure with a pointer member

d) Declaration of a structure type

27) How do you assign a value to a member of a structure using a pointer?

a) \*ptr.member = value;

**b) ptr->member = value;**

c) ptr.member = value;

d) ptr::member = value;

28) How do you pass a structure pointer to a function in C?

a) By value

**b) By reference**

c) By using the \* (asterisk) operator

d) By using the & (address-of) operator

29) Which of the following is true about the relationship between structures and structure pointers?

a) Structures and structure pointers are completely unrelated concepts.

b) Structures are used to define data types, while structure pointers are used to allocate memory.

**c) Structures can be assigned directly to structure pointers.**

d) Structure pointers can only be used to access structure members, but not modify them.

30) Guess the output:

#include <stdio.h>

struct point {

int x;

int y;

};

int main() {

struct point p = {5, 10};

struct point \*ptr = &p;

printf("%d %d", (\*ptr).y, (\*ptr).x);

return 0;

}

a) 5 10

**b) 10 5**

c) Compiler Error

d) None of the above