MULTIPLE CHOICE QUESTIONS

Sercan Külcü | 21.06.2023

Contents

Contents 1
1 Multiple choice questions 2

1 Multiple choice questions

What is the output of the following code snippet?

```
#include <stdio.h>

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

a) 11

b) 12

c) 13

d) Compiler error

Answer: c) 13

Which of the following statements is true about the "const" keyword in C?

a) It specifies that a variable cannot be modified.

b) It specifies that a function cannot be overridden.

c) It specifies that a variable must be initialized.

d) It specifies that a function cannot have variable arguments.

Answer: a) It specifies that a variable cannot be modified.

What does the "volatile" keyword indicate in C?

a) The variable is shared between multiple threads.

b) The variable's value can be changed by external factors.

c) The variable should not be optimized by the compiler.

d) The variable is declared in the global scope.

Answer: b) The variable's value can be changed by external factors.

What is the purpose of the "typedef" keyword in C?

a) It creates a new data type alias.

b) It specifies a function prototype.

c) It defines a macro.

d) It declares a variable with external linkage.

Answer: a) It creates a new data type alias.

What is the result of the following code snippet?

```
#include <stdio.h>

int main() {
  int arr[] = {1, 2, 3, 4, 5};
  int *ptr = (int*)((char*)arr + 2);
  printf("%d", *ptr);
  return 0;
}
```

a) 2

b) 3

c) 4

d) Compiler error

Answer: b) 3

Which of the following is true about function pointers in C?

a) They can only point to functions with the same return type.

b) They are used to call functions indirectly.

c) They can be assigned a NULL value but not a function address.

d) They can only be used with recursive functions.

Answer: b) They are used to call functions indirectly.

What is the output of the following code snippet?

```
#include <stdio.h>

int main() {
  char str[10] = "Hello";
  printf("%d", sizeof(str));
  return 0;
}
```

a) 6

b) 10

c) 11

d) 12

Answer: b) 10

Which of the following is true about the "restrict" keyword in C?

a) It restricts the usage of a variable to a specific block.

b) It specifies that a variable cannot be modified.

c) It indicates that a pointer is not aliased.

d) It limits the scope of a variable to a specific function.

Answer: c) It indicates that a pointer is not aliased.

What is the purpose of the "sizeof" operator in C?

a) It returns the memory size of a variable or data type.

b) It calculates the size of a string.

c) It returns the number of elements in an array.

d) It specifies the size of a function prototype.

Answer: a) It returns the memory size of a variable or data type.

Which of the following is true about the "malloc" function in C?

a) It allocates memory on the stack.

b) It is used to allocate memory for a single character.

c) It returns a pointer to a contiguous block of memory.

d) It automatically frees the allocated memory.

Answer: c) It returns a pointer to a contiguous block of memory.

What is the output of the following code snippet?

```
#include <stdio.h>

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

A) 30

B) 31

C) 32

D) 33

Answer: D) 33

What does the volatile keyword indicate in C?

A) It specifies that the variable cannot be modified

B) It specifies that the variable can only be accessed by a single thread

C) It requests the compiler to avoid optimizing the variable

D) It indicates that the variable is a constant

Answer: C) It requests the compiler to avoid optimizing the variable

Which of the following is an invalid declaration in C?

A) int arr[5];

B) int* ptr = NULL;

C) int matrix[3][3];

D) int array[];

Answer: D) int array[];

What is the size of the union in the following code snippet?

```
#include <stdio.h>

union Example {
  int x;
  char c;
  double d;
};

int main() {
  union Example e;
  printf("%lu\n", sizeof(e));
```

```
    return 0;
}
```

A) 4

B) 8

C) 16

D) It is implementation-dependent

Answer: C) 16

What does the static keyword signify when used with a global variable in C?

A) It restricts the variable's scope to the current source file

B) It specifies that the variable cannot be modified

C) It indicates that the variable is a constant

D) It requests the compiler to avoid optimizing the variable

Answer: A) It restricts the variable's scope to the current source file

Which of the following is NOT a valid way to initialize a string in C?

A) char str[10] = "Hello";

B) char str[] = "Hello";

C) char* str = "Hello";

D) char str[10]; strcpy(str, "Hello");

Answer: C) char* str = "Hello";

What is the output of the following code snippet?

```
#include <stdio.h>


int main() {
  int x = 10;
  int y = (x > 5) ? 1 : 0;
  printf("%d\n", y);
  return 0;
}
```

A) 1

B) 5

C) 0

D) 10

Answer: A) 1

In C, what is the result of dividing an integer by zero?

A) The program crashes with a runtime error

B) The result is implementation-dependent

C) The result is always zero

D) The result is undefined

Answer: D) The result is undefined

Which header file is required to use the malloc() function in C?

A) <stdlib.h>

B) <stdio.h>

C) <math.h>

D) <string.h>

Answer: A) <stdlib.h>

What is the scope of a label defined in C?

A) It can only be accessed within the current function

B) It can be accessed by any function within the same source file

C) It can be accessed by any function within the same source file and other source files

D) It can be accessed by any function within the same source file and other source files through a forward declaration

Answer: A) It can only be accessed within the current function

What is the output of the following code snippet?

```
#include <stdio.h>


int main() {

   int arr[] = {1, 2, 3, 4, 5};

   int *ptr = arr;

   printf("%d", *(ptr++));

   return 0;

}
```

A) 1

B) 2

C) 3

D) 4

Answer: B) 2

Which of the following is NOT a valid storage class specifier in C?

A) auto

B) static

C) register

D) virtual

Answer: D) virtual

What is the size of the following structure in C?

```
struct Student {
   int rollNo;
   char name[20];
};
```

A) 20

B) 24

C) 28

D) 32

Answer: C) 28

What is the value of x after the following code snippet?

```
#include <stdio.h>

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

A) 30

B) 31

C) 32

D) 33

Answer: B) 31

Which of the following is a correct way to declare a pointer to a function in C?

A) int *funcPtr();

B) int (*funcPtr)();

C) int *funcPtr[];

D) int *(funcPtr)();

Answer: B) int (*funcPtr)();

What is the output of the following code snippet?

```
#include <stdio.h>

int main() {
    int x = 10;
    int y = 20;
    int *ptr;
    ptr = &x;
    *ptr += y;
    printf("%d", x);
    return 0;
}
```

A) 10

B) 20

C) 30

D) 40

Answer: C) 30

Which of the following correctly defines a macro in C?

A) #define MAX_VALUE 100

B) #define MAX_VALUE = 100

C) #define MAX_VALUE (100)

D) #define MAX_VALUE == 100

Answer: A) #define MAX_VALUE 100

What is the scope of a global variable in C?

A) Local to the function where it is defined

B) Local to the block where it is defined

C) Accessible throughout the entire program

D) Accessible only within the file where it is defined

Answer: C) Accessible throughout the entire program

Which of the following is the correct syntax to allocate memory dynamically for a 2D array in C?

A) int **arr = malloc(sizeof(int) * rows * cols);

B) int arr[][] = (int **)malloc(sizeof(int *) * rows * cols);

C) int **arr = (int **)malloc(sizeof(int *) * rows);

D) int **arr = (int **)malloc(sizeof(int **) * rows);

Answer: C) int **arr = (int **)malloc(sizeof(int *) * rows);

What is the purpose of the volatile keyword in C?

A) To declare a variable as a constant

B) To declare a variable as a global

C) To specify that a variable can be accessed by multiple threads

D) To specify that a variable can change unexpectedly

Answer: D) To specify that a variable can change unexpectedly