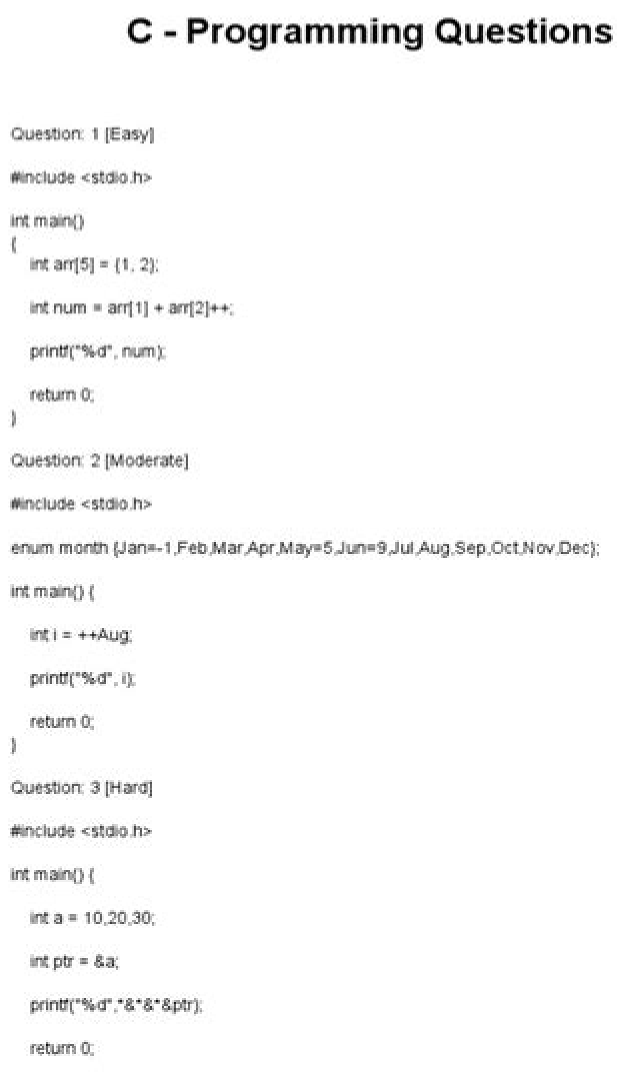
C Programming Questions With Solutions



C - Programming Questions

Question: 1 [Easy]

```
#include <stdio.h>

int main()
{
    int arr[5] = {1, 2};

    int num = arr[1] + arr[2]++;

    printf("%d", num);

    return 0;
}
```

Question: 2 [Moderate]

```
#include <stdio.h>

enum month {Jan=-1,Feb,Mar,Apr,May=5,Jun=9,Jul,Aug,Sep,Oct,Nov,Dec};

int main() {

    int i = ++Aug;

    printf("%d", i);

    return 0;
}
```

Question: 3 [Hard]

```
#include <stdio.h>

int main() {

    int a = 10,20,30;

    int ptr = &a;

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

    return 0;
```

C programming questions with solutions are essential for anyone looking to strengthen their understanding of the C programming language. C is one of the most widely used programming languages, known for its efficiency and versatility. Whether you are preparing for interviews, exams, or simply improving your coding skills, practicing C programming questions can be incredibly beneficial. This article will explore various categories of C programming questions, provide detailed solutions, and offer insights into best practices.

Types of C Programming Questions

C programming questions can be categorized into several types based on their complexity and the concepts they cover. Below are some common categories:

1. Basic Syntax and Data Types

These questions focus on the fundamental aspects of C, including syntax, data types, and operators.

2. Control Structures

Control structures include conditional statements (if, switch) and loops (for, while, do-while). Questions in this category assess your understanding of flow control in C.

3. Functions and Recursion

These questions test your ability to create and use functions, including recursive functions.

4. Arrays and Strings

This section covers questions related to manipulating arrays and strings, which are crucial data structures in C.

5. Pointers and Dynamic Memory Allocation

Pointers are a powerful feature of C. Questions may involve pointer arithmetic, pointer to functions, and dynamic memory allocation using malloc and free.

6. Structures and Unions

These questions assess your understanding of user-defined data types in C.

7. File Handling

This category includes questions related to reading from and writing to files.

8. Algorithms and Data Structures

Questions in this section may involve implementing common algorithms and data structures like sorting, searching, linked lists, and trees.

Sample Questions and Solutions

Now, let’s delve into some sample C programming questions along with their solutions.

1. Basic Syntax and Data Types

Question: Write a C program to find the sum of two integers.

Solution:
```c
include

int main() {
int a, b, sum;

printf("Enter two integers: ");
scanf("%d %d", &a, &b);

sum = a + b;

printf("Sum = %d\n", sum);
return 0;
}
```

Explanation:
- The program begins by including the standard input-output library.
- It declares three integer variables: `a`, `b`, and `sum`.
- The user is prompted to enter two integers, which are read using `scanf`.
- The sum is calculated and printed.

2. Control Structures

Question: Write a C program that checks whether a number is even or odd.

Solution:
```c
include

int main() {
int number;

printf("Enter an integer: ");
scanf("%d", &number);

if (number % 2 == 0) {
printf("%d is even.\n", number);
} else {
printf("%d is odd.\n", number);
}

return 0;
}
```

Explanation:
- This program takes an integer input from the user.
- It uses the modulus operator to determine if the number is even or odd, printing the appropriate message.

3. Functions and Recursion

Question: Write a recursive function to calculate the factorial of a number.

Solution:
```c
include

int factorial(int n) {
if (n == 0)
return 1;
else
return n factorial(n - 1);
}

int main() {
int num;

printf("Enter a positive integer: ");
scanf("%d", &num);

printf("Factorial of %d = %d\n", num, factorial(num));
return 0;
}
```

Explanation:
- The `factorial` function calls itself recursively until it reaches the base case (`n == 0`).
- The `main` function handles user input and displays the result.

4. Arrays and Strings

Question: Write a C program to reverse a string.

Solution:
```c
include
include
```

```
int main() {
char str[100], reversed[100];
int len, i, j = 0;

printf("Enter a string: ");
fgets(str, sizeof(str), stdin);

len = strlen(str);

for (i = len - 1; i >= 0; i--) {
reversed[j++] = str[i];
}
reversed[j] = '\0'; // Null-terminate the reversed string

printf("Reversed string: %s\n", reversed);
return 0;
}
```

Explanation:
- This program uses `fgets` to read a string, which allows spaces.
- The string is reversed using a loop, and the reversed string is printed.

5. Pointers and Dynamic Memory Allocation

Question: Write a C program that dynamically allocates memory for an array and calculates the average.

Solution:
```c
include
include

int main() {
int n, i;
float sum = 0.0, average;

printf("Enter the number of elements: ");
scanf("%d", &n);

float arr = (float)malloc(n sizeof(float));

if (arr == NULL) {
printf("Memory allocation failed!\n");
return 1;
}

printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
```

```
scanf("%f", &arr[i]);
sum += arr[i];
}

average = sum / n;
printf("Average = %.2f\n", average);

free(arr); // Free allocated memory
return 0;
}
```

Explanation:
- The program allocates memory for an array of floats based on user input.
- It calculates the sum and average, then frees the allocated memory.

6. Structures and Unions

Question: Create a structure to store student information and print it.

Solution:
```c
include

struct Student {
char name[50];
int age;
float grade;
};

int main() {
struct Student student;

printf("Enter student name: ");
fgets(student.name, sizeof(student.name), stdin);
printf("Enter age: ");
scanf("%d", &student.age);
printf("Enter grade: ");
scanf("%f", &student.grade);

printf("\nStudent Information:\n");
printf("Name: %s", student.name);
printf("Age: %d\n", student.age);
printf("Grade: %.2f\n", student.grade);

return 0;
}
```

Explanation:
- A `Student` structure is defined with fields for name, age, and grade.
- The program collects and displays the data.

7. File Handling

Question: Write a C program to read from a file and display its content.

Solution:
```c
include

int main() {
FILE file;
char ch;

file = fopen("example.txt", "r");
if (file == NULL) {
printf("Could not open file.\n");
return 1;
}

while ((ch = fgetc(file)) != EOF) {
putchar(ch);
}

fclose(file);
return 0;
}
```

Explanation:
- The program opens a file for reading, checks if it was opened successfully, and reads the content character by character until EOF.

8. Algorithms and Data Structures

Question: Implement a bubble sort algorithm.

Solution:
```c
include

void bubbleSort(int arr[], int n) {
int i, j, temp;
for (i = 0; i < n - 1; i++) {
for (j = 0; j < n - i - 1; j++) {
```

```
if (arr[j] > arr[j + 1]) {
temp = arr[j];
arr[j] = arr[j + 1];
arr[j + 1] = temp;
}
}
}
}

int main() {
int arr[] = {64, 34, 25, 12, 22, 11, 90};
int n = sizeof(arr) / sizeof(arr[0]);

bubbleSort(arr, n);

printf("Sorted array: \n");
for (int i = 0; i < n; i++)
printf("%d ", arr[i]);

return 0;
}
```

Explanation:
- The `bubbleSort` function sorts the array in ascending order using the bubble sort algorithm.
- The `main` function initializes an array and calls the sorting function.

Conclusion

In conclusion, practicing C programming questions with solutions is an excellent way to enhance your programming skills. The examples provided in this article cover a range of topics from basic

Frequently Asked Questions

What is the difference between '==' and '===' in C?

'==' is used for equality comparison of values, while '===' is not a valid operator in C. C does not have a strict equality operator like JavaScript.

How can I dynamically allocate memory in C?

You can use functions like malloc(), calloc(), or realloc() from the stdlib.h library to dynamically allocate memory. For example: 'int arr = (int)malloc(n sizeof(int));' allocates memory for an array of n integers.

What is a segmentation fault in C?

A segmentation fault occurs when a program tries to access a memory location that it's not allowed to access. This often happens due to dereferencing a null or uninitialized pointer.

How do I swap two numbers without using a third variable in C?

You can swap two numbers using arithmetic operations: 'a = a + b; b = a - b; a = a - b;' or by using bitwise XOR: 'a = a ^ b; b = a ^ b; a = a ^ b;'.

What is the purpose of the 'static' keyword in C?

The 'static' keyword limits the visibility of a variable or function to the file in which it is declared, and for variables, it retains their value between function calls.

How do I read and write to a file in C?

You can use 'fopen()' to open a file, 'fprintf()' or 'fwrite()' to write to it, 'fscanf()' or 'fread()' to read from it, and 'fclose()' to close the file. Example: 'FILE fp = fopen("file.txt", "r");'.

What is a pointer and how is it used in C?

A pointer is a variable that stores the address of another variable. It is used for dynamic memory allocation, arrays, and to pass variables by reference. Example: 'int ptr = &var;' assigns the address of 'var' to 'ptr'.

C Programming Questions With Solutions

c□□□□□□□□□□□? - □□
□□□□□□□ □□□□□□ □□□□□□□□□C□□□□□□□□□□□□□□□□□□□G□□□□ 1□□□C□□□□ 2□□□C□ ...

□□C□□□□CMD□□□□□□15□□□□CMD□□□□□□□□□□ ...
Nov 16, 2024 · □□C□□□□CMD□□□□□□15□□□□CMD□□□□□□□□□□□C□□□□□□□□Windows□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ...

C□APPData□□□□□□□□□□□□□□□□G□ - □□
C□□□□□□□□□□ □□□□□□□□□□C□□□□□□□□Users□□□□□□□Windows□□□□□□□66.7%□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ °C□□□□□ - □□□□□
□□□□□□ °C□□□□□□□□□□□□□□□□□□□□°C□□□□□□□□□□□□□□□□□□□□“C”□□□□□□□□□□□□□□□□□□ (C)□□□□□□□□□□□□□°C□□□□□□□□

[illegible] …

C [illegible] *C++* [illegible] - [illegible]

C[illegible]C++[illegible]C++[illegible]C[illegible] [illegible]C++[illegible]C[illegible] [illegible]C[illegible]C++[illegible]C[illegible]C++[illegible] [illegible] …

C:\users[illegible]_[illegible]

C:\users[illegible]1[illegible]users[illegible]“[illegible]”[illegible]2[illegible]C[illegible]3[illegible]C[illegible]“[illegible]”[illegible]users[illegible] …

[illegible]*C/D*[illegible]*C*[illegible]*D*[illegible] - [illegible]

[illegible]G[illegible] [illegible] [illegible]C[illegible]D[illegible] [illegible] …

[illegible]*A*[illegible]*B*[illegible]*C*[illegible]*D*[illegible]*E*[illegible]*F*[illegible]*G*[illegible] …

1[illegible]A[illegible] 2[illegible]B[illegible] 3[illegible]C[illegible] 4[illegible]D[illegible] 5[illegible]E[illegible] 6[illegible]F[illegible] 7[illegible]G[illegible] [illegible] 1[illegible]“[illegible]S ”[illegible] …

bigbang[illegible] [illegible] …

Aug 15, 2014 · bigbang[illegible] [illegible]BigBang [illegible] [illegible] [illegible] Ye the finally I realize that I'm nothing without you I was so wrong forgive …

[illegible] - [illegible]

[illegible] qBittorrent [illegible] Windows[illegible]Mac [illegible] Linux[illegible] [illegible] BT [illegible] [illegible]——[illegible] …

c[illegible]*?* - [illegible]

[illegible] [illegible] [illegible]C[illegible]G[illegible] 1[illegible]C[illegible] 2[illegible]C[illegible] …

[illegible]*C*[illegible]*CMD*[illegible]*15*[illegible]*CMD*[illegible]*C*[illegible]_[illegible] …

Nov 16, 2024 · [illegible]C[illegible]CMD[illegible]15[illegible]CMD[illegible]C[illegible]Windows[illegible]C[illegible] …

C[illegible]APPData[illegible]G[illegible] - [illegible]

C[illegible] [illegible]C[illegible]Users[illegible]Windows[illegible]66.7%[illegible]C[illegible]

[illegible] °C[illegible] - [illegible]

[illegible] °C[illegible]℃[illegible]“C”[illegible] (C)[illegible]℃[illegible] …

C [illegible] C++ [illegible] - [illegible]

C[illegible]C++[illegible]C++[illegible]C[illegible] [illegible]C++[illegible]C[illegible] [illegible]C[illegible]C++[illegible]C[illegible]C++[illegible] [illegible] …

Explore essential C programming questions with solutions to enhance your coding skills. Get practical insights and examples. Learn more to master C programming today!

Back to Home