C Preprocessor Job Interview Questions And Answers

Interview Questions Answers

https://interviewquestionsanswers.org/

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on C Preprocessor will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit C Preprocessor Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in C Preprocessor category. To ensure quality, each submission is checked by our team, before it becomes live. This C Preprocessor Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

C Preprocessor Interview Questions And Answers Guide.

Question - 1:

What is include directive in C?

Ans:

The include directive is used to include files like as we include header files in the beginning of the program using #include directive like
* #include<stdio.h>
* #include<conio.h></conio.h></stdio.h>

View All Answers

Question - 2:

What is define directive?

Ans:

It is used to assign names to different constants or statements which are to be used repeatedly in a program. These defined values or statement can be used by main or in the user defined functions as well.

View All Answers

Question - 3:

What are # Preprocessor operator in C?

Ans:

is called stringize opertor and turns the argument it precede into a quoted string. Use of # is shown in the C Source code given below and should be properly studied.

View All Answers

Question - 4:

Can a file other than a .h file be included with #include?

Ans:

The preprocessor will include whatever file you specify in your #include statement. Therefore, if you have the line
#include <macros.inc>
in your program, the file macros.inc will be included in your precompiled program. It is, however, unusual programming practice to put any file that does not have a .h or .hpp extension in an #include statement. You should always put a .h extension on any of your C files you are going to include. This method makes it easier for you and others to identify which files are being used for preprocessing purposes.</macros.inc>

View All Answers

Question - 5:

What are the advantages of using macro?

Ans:

In modular programming, using functions is advisable when a certain code is repeated several times in a program. However, everytime a function is called the control gets transferred to that function and then back to the calling function. This consumes a lot of execution time. One way to save this time is by using macros. Macros substitute a function call by the definition of that function. This saves execution time to a great extent.

View All Answers

Question - 6:

If you know then define #pragma?

Ans:

The #pragma Directives are used to turn ON or OFF certain features. They vary from compiler to compiler.

View All Answers

Question - 7:

What is a macro in C Preprocessor?

Ans:

A macro is a preprocessor directive that provides a mechanism for token replacement in your source code. Macros are created by using the #define statement. Here is an example of a macro:
#define VERSION_STAMP "1.02"

View All Answers

Question - 8:

What is typedf?

Ans:

The typedef clause can be used in a similar manner.
typedef long int int32; /* 32 bit signed integer */
The typedef is preferred over the #define because is better integrated into the C language, and it can create more kinds of variable types than a mere define.

View All Answers

Question - 9:

What is the mean of function?

Ans:

Functions allow for modular programming. You must remember that all parameters passed into function in C are passed by value!

View All Answers

Question - 10:

What is #define?

Ans:

The #define directive can be used to define types, such as:
#define INT32 long int /* 32 bit signed integer type */

View All Answers

Question - 11:

What is ## Preprocessor operator in C?

Ans:

is called the pasting opertor which is used to concates two tokens. Use of ## is shown in the source code.

View All Answers

Question - 12:

What is the general form of #line preprocessor?

Ans:

General form of #line preprocessor is #line number "filename"
Here the file name is optional. Filename string replaces the string value of _ _FILE_ _ while the number changes the value of _ _LINE_ _.
The major use of #line is in debugging and rare programming situation.
Following C Source code shows the #line preprocessor in action -

```
#include <stdio.h>
int main ()
{
printf ("n%d", __LINE__); //Prints 6
#line 100;
printf ("n%d",__LINE__); // Prints 101
printf ("n%d", __FILE__);// Prints original source file name
#line 103 "Super C"
printf ("n%d", __FILE__); //Prints Super C
return 0;
}</stdio.h>
```

View All Answers

Question - 13:

What is file in C Preprocessor?

Ans:

This macro stores the file name of the source file being compiled

View All Answers

Question - 14:

What is line in C Preprocessor?

Ans:

This a dynamic macro which stores the line number of the line presently being compiled. Value is constantly updated as compiler moves forward.

View All Answers

Question - 15:

What is #line?

Ans:

#line preprocessor is used to change the values of 2 MACROS , _ _LINE _ _ and _ _ FILE _ _.

View All Answers

Question - 16:

What is #error and use of it?

Ans:

The use of this preprocessor is in debugging. Whenever this preprocessor is encountered during compilation the compiler would stop compilation and display the error message associated with the preprocessor in complation Output/Result Window. Depending upon compiler any other debugging information will also accompany your error message.
General form of #error is -
#error message
Also note that message doesnot need to be in double quotes.
Following source code display the use of #error preprocessor directive in C -

```
#include <stdio.h>
int main ()
{
#error I am Error and while i am here i will not let this program compile :-) .
return 0;
}</stdio.h>
```

View All Answers

Question - 17:

Where define directive used?

Ans:

* Defining a constant
* Defining a statement
* Defining a mathematical expression

For example
* #define PI 3.141593
* #define TRUE 1
* #define floatingpointno float

View All Answers

Question - 18:

What are the preprocessor categories?

Ans:

* Macro substitution division
* File inclusion division
* Compiler control division

View All Answers

Question - 19:

What are types of Preprocessor in C?

Ans:

* #define and #undef - Used for defining and undefining MACROS.
* #error - Used for Debugging
* #include - Used for combining source code files
* #if, #else, #elseif, and #endif - Used for conditional compilation similar to if - else statment.
* #ifdef and #ifndef - Conditional Compilation on basis of #define and #undef
* #line - Controls the program line and file macros.
* #pragma - Used for giving compiler instruction. Highly specific to the compiler being used.
* # and ## - Operators used for stringize and concating operation respectively.

View All Answers

Question - 20:

What are the advantages of C Preprocessor?

Ans:

* Programs easier to develop,
* Easier to read,
* Easier to modify
* C code more transportable between different machine architectures.

View All Answers

Question - 21:

What is C Preprocessor mean?

Ans:

The C preprocessor is a tool which filters your source code before it is compiled. The preprocessor allows constants to be named using the #define notation.It is particularly useful for selecting machine dependent pieces of code for different computer types, allowing a single program to be compiled and run on several different computers.

View All Answers

Question - 22:

What is cpp?

Ans:

The preprocessor is called cpp, however it is called automatically by the compiler so you will not need to call it while programming in C.

View All Answers

Question - 23:

If #include is used with file name in angular brackets
a) The file is searched for in the standard compiler include paths
b) The search path is expanded to include the current source directory
c) Both a & b
d) None of the mentioned

Ans:

a
(The file is searched for in the standard compiler include paths)
Explanation:With the #include, if the filename is enclosed within angle brackets, the file is searched for in the standard compiler include paths.

View All Answers

Question - 24:

The preprocessor provides the ability for _____.
a) The inclusion of header files
b) The inclusion of macro expansions
c) Conditional compilation and line control.
d) All of the mentioned

Ans:

d
(All of the mentioned)
Explanation:The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

View All Answers

Question - 25:

The #include directive
a) Tells the preprocessor to grab the text of a file and place it directly into the current file
b) Statements are typically placed at the top of a program
c) both a & b
d) None of a & b

Ans:

c
(both a & b)
Explantion:The #include directive tells the preprocessor to grab the text of a file and place it directly into the current file and are statements are typically placed at the top of a program.

View All Answers

Question - 26:

The C-preprocessors are specified with _____symbol.
a) #
b) $
c) " "
d) None of the mentioned

Ans:

a
(#)
Explanation:The C-preprocessors are specified with # symbol.

View All Answers

Question - 27:

#pragma exit is primarily used for?
a) Checking memory leaks after exitting the program
b) Informing Operating System that program has terminated
c) Running a function at exitting the program

d) No such preprocessor exist

Ans:

c
(Running a function at exitting the program)
Explanation:It is primarily used for running a function upon exitting the program.

View All Answers

Question - 28:

#include statement must be written
a) Before main()
b) Before any scanf/printf
c) After main()
d) It can be written anywhere

Ans:

b
(Before any scanf/printf)
Explanation:Using these directives before main() improves readability.

View All Answers

Question - 29:

1. Which of the following are C preprocessors?
a) #ifdef
b) #define
c) #endif
d) All of the mentioned

Ans:

d
(All of the mentioned)

View All Answers

Question - 30:

Do you have any idea about the use of "auto" keyword?

Ans:

When a certain variable is declared with the keyword â€˜auto' and initialized to a certain value, then within the scope of the variable, it is reinitialized upon being called repeatedly.

View All Answers

Question - 31:

Can you please explain the scope of static variables?

Ans:

Static variables in C have the scopes;
1. Static global variables declared at the top level of the C source file have the scope that they can not be visible external to the source file. The scope is limited to that file.
2. Static local variables declared within a function or a block, also known as local static variables, have the scope that, they are visible only within the block or function like local variables. The values assigned by the functions into static local variables during the first call of the function will persist / present / available until the function is invoked again.
The static variables are available to the program, not only for the function / block. It has the scope within the current compile. When static variable is declared in a function, the value of the variable is preserved , so that successive calls to that function can use the latest updated value. The static variables are initialized at compile time and kept in the executable file itself. The life time extends across the complete run of the program.
Static local variables have local scope. The difference is, storage duration. The values put into the local static variables, will still be present, when the function is invoked next time.

View All Answers

Question - 32:

What do you know about the use of bit field?

Ans:

Packing of data in a structured format is allowed by using bit fields. When the memory is a premium, bit fields are extremely useful. For example:
- Picking multiple objects into a machine word : 1 bit flags can be compacted
- Reading external file formats : non-standard file formats could be read in, like 9 bit integers
This type of operations is supported in C language. This is achieved by putting :'bit length' after the variable. Example:

```
struct packed_struct {
      unsigned int var1:1;
      unsigned int var2:1;
      unsigned int var3:1;
      unsigned int var4:1;
      unsigned int var5:4;
      unsigned int funny_int:9;
} pack;
```

packed-struct has 6 members: four of 1 bit flags each, and 1 4 bit type and 1 9 bit funny_int.
C packs the bit fields in the structure automatically, as compactly as possible, which provides the maximum length of the field is less than or equal to the integer word

length the computer system.
The following points need to be noted while working with bit fields:
- The conversion of bit fields is always integer type for computation
- Normal types and bit fields could be mixed / combined
- Unsigned definitions are important.

View All Answers

Question - 33:

Tell us bitwise shift operators?

Ans:

The bitwise operators are used for shifting the bits of the first operand left or right. The number of shifts is specified by the second operator.
Expression << or >> number of shifts
Ex:
number<<3;/* number is an operand - shifts 3 bits towards left*/
number>>2; /* number is an operand â€" shifts 2 bits towards right*/
The variable number must be an integer value.
For leftward shifts, the right bits those are vacated are set to 0. For rightward shifts, the left bits those are vacated are filled with 0's based on the type of the first operand after conversion.
If the value of â€˜number' is 5, the first statement in the above example results 40 and stored in the variable â€˜number'.
If the value of â€˜number' is 5, the second statement in the above example results 0 (zero) and stored in the variable â€˜number'.

View All Answers

C Programming Most Popular & Related Interview Guides

1 : **C Functions Interview Questions and Answers.**

2 : C Pointers Interview Questions and Answers.

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG