C Preprocessor (Questions)

Sayak Haldar
IIEST, Shibpur

1. What is the sequence for preprocessor to look for the file within <> ?

a) The predefined location then the current directory
b) The current directory then the predefined location
c) The predefined location only
d) The current directory location

2. Which directory the compiler first looks for the file when using #include

a) Current directory where program is saved
b) C:COMPILERINCLUDE
c) S:SOURCEHEADERS
d) Both (b) and (c) simultaneously

3. What would happen if you create a file stdio.h and use #include "stdio.h" ?

a) The predefined library file will be selected
b) The user-defined library file will be selected
c) Both the files will be included
d) The compiler won't accept the program

4. How is search done in #include<somelibrary.h> and #include "somelibrary.h" normally or conventionally?

a) When former is used, current directory is searched and when latter is used, standard directory is searched
b) When former is used, predefined directory is searched and when latter is used, current directory is searched and then predefined directories are searched
c) When former is used, search is done in implementation defined manner and latter is used to search current directory
d) For both, search for somelibrary is done in implementation-defined manner

5. Can function definition be present in header files?

a) Yes
b) No
c) Depends on the compiler
d) Depends on the standard

6. Comment on the output of this C code?

```
#include <stdio.h>
```

```
#include "test.h"
#include "test.h"
int main()
{
    //some code
}
```

a) true
b) Compile time error
c) false
d) Depends on the compiler

7. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) m ## n
void myfunc();
int main()
{
    myfunc();
    return 0;
}
void myfunc()
{
    printf("%d\n", foo(2, 3));
}
```

a) 23
b) 2 3
c) Compile time error
d) Undefined behaviour

8. If the file name is enclosed in double quotation marks

a) The preprocessor treats it as a user-defined file
b) The preprocessor treats it as a system-defined file
c) Both a & b
d) None of the mentioned

9. If the file name is enclosed in angle brackets

a) The preprocessor treats it as a user-defined file
b) The preprocessor treats it as a system-defined file
c) Both a & b
d) None of the mentioned

10. What is the output of this C code?

```
#include <stdio.h>

int main()

{

    printf("hello");

    return 0;

}
```

a) hello
b) Nothing
c) compile time error
d) Depends on compiler

11. The below two lines are equivalent to

#define C_IO_HEADER
#include C_IO_HEADER
a) #include
b) #include”printf”
c) #include”C_IO_HEADER”
d) #include

12. What is the output of this C code?

```
#include <stdio.h>

#include "printf"
```

```
int main()
{
    printf("hello");
    return 0;
}
```

a) hello
b) Error
c) Depends on compiler
d) Varies

13. Property which allows to produce different executable for different platforms in C is called?

a) File inclusion
b) Selective inclusion
c) Conditional compilation
d) Recursive macros

14. #include

a) Preprocessor directive
b) Inclusion directive
c) File inclusion directive
d) None of the mentioned

15. C preprocessors can have compiler specific features.

a) true
b) false
c) Depends on the standard
d) Depends on the platform

16. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) m * n = 10
int main()
```

```
{
    printf("in main\n");
    return 0;
}
```

a) In main
b) Compilation error as lvalue is required for the expression m*n=10
c) Preprocessor error as lvalue is required for the expression m*n=10
d) None of the mentioned

17. C preprocessor is conceptually the first step during compilation

a) true
b) false
c) Depends on the compiler
d) Depends on the standard

18. Preprocessor feature that supply line numbers and filenames to compiler is called?

a) Selective inclusion
b) macro substitution
c) Concatenation
d) Line control

19. #include

a) Library, Library
b) Library, user-created header
c) User-created header, library
d) They can include all types of file

20. A preprocessor is a program

a) That processes its input data to produce output that is used as input to another program
b) That is nothing but a loader
c) That links various source files
d) All of the mentioned

21. Which of the following are C preprocessors?

a) #ifdef
b) #define
c) #endif
d) All of the mentioned

22. #include<stdio.h> statement must be written

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

23. #pragma exit is primarily used for?

a) Checking memory leaks after exiting the program
b) Informing Operating System that program has terminated
c) Running a function at exiting the program
d) No such preprocessor exist

24. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int one = 1, two = 2;
    #ifdef next
    one = 2;
    two = 1;
    #endif
    printf("%d, %d", one, two);
    return 0;
}
```

a) 1, 1
b) 1, 2
c) 2, 1
d) 2, 2

25. The C-preprocessors are specified with _________symbol.

a) #
b) $
c) ” ”

d) None of the mentioned

26. The #include directive

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

27. 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

28. #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

29. What is the output of this C code?

```
#include <stdio.h>
#define foo(m, n) m ## n
int main()
{
    printf("%s\n", foo(k, l));
    return 0;
}
```

a) k l
b) kl
c) Compile time error
d) Undefined behaviour

30. What is the output of this C code?

```
#include <stdio.h>
```

```
#define foo(m, n) " m ## n "
int main()
{
    printf("%s\n", foo(k, l));
}
```

a) k l
b) kl
c) Compile time error
d) m ## n

31. What is the output of this C code?

```
#include <stdio.h>
#define foo(x, y) #x #y
int main()
{
    printf("%s\n", foo(k, l));
    return 0;
}
```

a) kl
b) k l
c) xy
d) Compile time error

32. What is the output of this C code?

```
#include <stdio.h>
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d\n",foo(i + j, 3));
    return 0;
}
```

a) Divided by zero exception
b) Compile time error
c) -8
d) -4

33. What is the output of this C code?

```
#include <stdio.h>
void f();
int main()
{
    #define foo(x, y) x / y + x
    f();
}
void f()
{
    printf("%d\n", foo(-3, 3));
}
```

a) -8
b) -4
c) Compile time error
d) Undefined behaviour

34. What is the output of this C code?

```
#include <stdio.h>
void f();
int main()
```

```
{
    #define max 10
    f();
    return 0;
}
void f()
{
    printf("%d\n", max * 10);
}
```

a) 100
b) Compile time error since #define cannot be inside functions
c) Compile time error since max is not visible in f()
d) Undefined behaviour

35. What is the output of this C code?

```
#include <stdio.h>
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d ", foo(i + j, 3));
    printf("%d\n", foo(-3, 3));
    return 0;
}
```

a) -8 -4
b) -4 divided by zero exception
c) -4 -4
d) Divided by zero exception

36. What is the output of this C code?

```
#include <stdio.h>
int foo(int, int);
#define foo(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d ",foo(i + j, 3));
    #undef foo
    printf("%d\n",foo(i + j, 3));
}
int foo(int x, int y)
{
    return x / y + x;
}
```

a) -8 -4
b) Compile time error
c) -8 -8
d) Undefined behaviour

37. What is the advantage of #define over const?

a) Data type is flexible
b) Can have a pointer
c) Reduction in the size of the program
d) Both (a) and (c)

38. What is the output of this C code?

```
#include <stdio.h>
int main()
```

```
{
    #define max 37;
    printf("%d", max);
    return 0;
}
```

a) 37
b) Compile time error
c) Varies
d) Depends on compiler

39. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    #define max 37
    printf("%d", max);
}
```

a) 37
b) Run time error
c) Varies
d) Depends on compiler

40. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    #define const int
    const max = 32;
    printf("%d", max);
}
```

a) Run time error
b) 32
c) int
d) const

41. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    #define max 45
    max = 32;
    printf("%d", max);
}
```

a) 32
b) 45
c) Compile time error
d) Varies

42. What is the output of this C code?

```
#include <stdio.h>
# define max
void m()
{
    printf("hi");
}
void main()
{
    max;
    m();
}
```

a) Run time error
b) hi hi
c) Nothing
d) hi

43. What is the output of this C code?

```
#include <stdio.h>
#define A 1 + 2
#define B 3 + 4
int main()
{
    int var = A * B;
    printf("%d\n", var);
}
```

a) 9
b) 11
c) 12
d) 21

44. Which of the following Macro substitution are accepted in C?

a) #define A #define
   A VAR 20
b) #define A define
   #A VAR 20
c) #define #A #define
  #A VAR 20
d) None of the mentioned

45. Comment on the following code?

```
#include <stdio.h>
#define var 20);
```

```
int main()
{
    printf("%d\n", var
    return 0;
}
```

a) No errors, it will show the output 20
b) Compile time error, the printf braces aren’t closed
c) Compile time error, there are no open braces in #define
d) Both (b) and (c).

46. Which of the following properties of #define not true?

a) You can use a pointer to #define
b) #define can be made externally available
c) They obey scope rules
d) All of the mentioned

47. What is the output of this C code?

```
#include <stdio.h>
#define SYSTEM 20
int main()
{
    int a = 20;
    #if SYSTEM == a
    printf("HELLO ");
    #endif
    #if SYSTEM == 20
    printf("WORLD\n");
    #endif
}
```

a) HELLO
b) WORLD
c) HELLO WORLD
d) No Output

48. Comment on the following code?

```
#include <stdio.h>
#define Cprog
int main()
{
    int a = 2;
    #ifdef Cprog
    a = 1;
    printf("%d", Cprog);
}
```

a) No output on execution
b) Output as 1
c) Output as 2
d) Compile time error

49. The “else if” in conditional inclusion is written by?

a) #else if
b) #elseif
c) #elsif
d) #elif

50. What is the output of this C code?

```
#include <stdio.h>
#define COLD
int main()
{
    #ifdef COLD
    printf("COLD\t");
    #undef COLD
    #endif
    #ifdef COLD
    printf("HOT\t");
    #endif
}
```

a) HOT
b) COLD
c) COLD HOT
d) No Output

51. Which of the following sequences are unaccepted in C language?

a) #if
 #else
 #endif
b) #if
 #elif
 #endif
c) #if
 #if
 #endif
d) #if
 #undef
 #endif

52. In a conditional inclusion, if the condition that comes after the if holds.

a) Then the code up to the following #else or #elif or #endif is compiled
b) Then the code up to the following #endif is compiled even if #else or #elif is present
c) Both a & b
d) None of the mentioned

53. Conditional inclusion can be used for

a) Preventing multiple declarations of a variable
b) Check for existence of a variable and doing something if it exists
c) Preventing multiple declarations of same function

d) All of the mentioned

54. The #elif directive cannot appear after the preprocessor #else directive.

a) true
b) false
c) None of the mentioned
d) Varies

55. For each #if, #ifdef, and #ifndef directive.

a) There are zero or more #elif directives
b) Zero or one #else directive
c) One matching #endif directive
d) All of the mentioned

The #else directive is used for
a) Conditionally include source text if the previous #if, #ifdef, #ifndef, or #elif test fails.
b) Conditionally include source text if a macro name is not defined
c) Conditionally include source text if a macro name is defined
d) Ending conditional text

56. What is the output of this C code?

```
#include <stdio.h>
#define MIN 0
#if MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

a) 10 0
b) Compile time error
c) Undefined behaviour

d) None of the mentioned

57. What is the output of this C code?

```
#include <stdio.h>
#define MIN 0
#ifdef MIN
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

a) 10 0
b) Compile time error
c) Undefined behaviour
d) None of the mentioned

58. What is the output of this C code?

```
#include <stdio.h>
#define MIN 0
#if defined(MIN) + defined(MAX)
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

a) 10 0
b) Compile time error
c) Undefined behaviour
d) Somegarbagevalue 0

59. What is the output of this C code?

```
#include <stdio.h>
#define MIN 0
#if defined(MIN) - (!defined(MAX))
#define MAX 10
#endif
int main()
{
    printf("%d %d\n", MAX, MIN);
    return 0;
}
```

a) 10 0
b) Compile time error
c) Undefined behaviour
d) Somegarbagevalue 0

60. What is the output of code given below?

```
#include <stdio.h>
#define MIN 0);
#ifdef MIN
#define MAX 10
#endif
int main()
```

```
{
    printf("%d %d\n", MAX, MIN
    return 0;
}
```

a) 10 0
b) Compile time error due to illegal syntax for printf
c) Undefined behaviour
d) Compile time error due to illegal MIN value