

Session 7 exercise

Question 1

```
#include <stdio.h>
#define PRINT(i, limit) do \
{ \
    if (i++ < limit) \
    { \
        printf("GeeksQuiz\n"); \
        continue; \
    } \
}while(1)
```

```
int main()
{
    PRINT(0, 3);
    return 0;
}
```

How many times **GeeksQuiz** is printed in the above program?

- 1) 1
- 2) 2
- 3) 3
- 4) Compile-time error

Question 2

```
#include <stdio.h>
#if X == 3
    #define Y 3
#else
    #define Y 5
#endif
```

```
int main()
{
    printf("%d", Y);
    return 0;
}
```

What is the output of the above program?

- 1) 3
- 2) 5
- 3) 3 or 5 depending on value of X
- 4) Compile time error

Question 3

What is the output of following program?

```
#include <stdio.h>
#define macro(n, a, i, m) m##a##i##n
#define MAIN macro(n, a, i, m)
```

```
int MAIN()
{
    printf("GeeksQuiz");
    return 0;
}
```

- 1) Compiler Error
- 2) GeeksQuiz
- 3) MAIN
- 4) main

Question 4

```
#include <stdio.h>
#define X 3
#if !X
    printf("Geeks");
#else
    printf("Quiz");
#endif
```

```
#endif
int main()
{
    return 0;
}
```

- ☒ 1) Geeks
- 2) Quiz
- 3) Compiler Error
- 4) Runtime Error

Question 5

```
#include <stdio.h>
#define ISEQUAL(X, Y) X == Y
int main()
{
    #if ISEQUAL(X, 0)
        printf("Geeks");
    #else
        printf("Quiz");
    #endif
    return 0;
}
```

Output of the above program?

- 1) Geeks
- 2) Quiz
- 3) Any of Geeks or Quiz
- 4) Compile time error

Question 6

```
#include <stdio.h>
#define square(x) x*x
int main()
{
    int x;
    x = 36/square(6);
    printf("%d", x);
    return 0;
}
```

- 1) 1
- 2) 36
- 3) 0
- 4) Compiler Error

Question 7

Output?

```
# include <stdio.h>
# define scanf "%s Geeks Quiz "
int main()
{
    printf(scanf, scanf);
    return 0;
}
```

- 1) Compiler Error
- 2) %s Geeks Quiz
- 3) Geeks Quiz
- 4) %s Geeks Quiz Geeks Quiz

Question 8

```
#include <stdio.h>
#define a 10
int main()
{
    printf("%d ", a);

    #define a 50

    printf("%d ", a);
    return 0;
}
```

- 1) Compiler Error
- 2) 10 50
- 3) 50 50
- 4) 10 10

Question 9

Output?

```
#include<stdio.h>
#define f(g,g2) g##g2
int main()
{
    int var12 = 100;
    printf("%d", f(var,12));
    return 0;
}
```

- 1) 100
- 2) CompilerError
- 3) 0
- 1

Question 10

Which file is generated after pre-processing of a C program?

- 1) .p
- 2) .i
- 3) .o
- 4) .m

Question for Smart Mind

Write a program without main function?

Answers

1- Compile-time error

Question 1 Explanation:

The **PRINT** macro gets expanded at the pre-processor time i.e. before the compilation time. After the macro expansion, the if expression becomes: **if (0++ < 3)**. Since **0** is a constant figure and represents only r-value, applying increment operator gives compile-time error: lvalue required. lvalue means a memory location with some address.

2- 5

Question 2 Explanation:

In the first look, the output seems to be compile-time error because macro X has not been defined. In C, if a macro is not defined, the pre-processor assigns 0 to it by default. Hence, the control goes to the conditional else part and 5 is printed. See the next question for better understanding.

3- GeeksQuiz

Question 3 Explanation:

The program has a preprocessor that replaces "MAIN" with "macro(n, a, i, m)". The line "macro(n, a, i, m)" is again replaced by main. The key thing to note is **token pasting operator ##** which concatenates parameters to macro.

4- Compiler Error

Question 4 Explanation:

A program is converted to executable using following steps 1) Preprocessing 2) C code to object code conversion 3) Linking The first step processes macros. So the code is converted to following after the preprocessing step.

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

The above code produces error because printf() is called outside main. The following program works fine and prints "Quiz"

```
#include
#define X 3

int main()
{
    #if !X
        printf("Geeks");
    #else
        printf("Quiz");
    #endif

    return 0;
}
```

5- Geeks

Question 5 Explanation:

The conditional macro `#if ISEQUAL(X, 0)` is expanded to `#if X == 0`. After the pre-processing is over, all the undefined macros are initialized with default value 0. Since macro X has not been defined, it is initialized with 0. So, **Geeks** is printed.

6- 36

Question 6 Explanation:

Preprocessor replaces `square(6)` by `6*6` and the expression becomes `x = 36/6*6` and value of x is calculated as 36. Note that the macro will also fail for expressions `"x = square(6-2)"` If we want correct behavior from macro `square(x)`, we should declare the macro as

```
#define square(x) ((x)*(x))
```

7- %s Geeks Quiz Geeks Quiz

Question 7 Explanation:

After pre-processing phase of compilation, `printf` statement will become. `printf("%s Geeks Quiz ", "%s Geeks Quiz ");` Now you can easily guess why output is `"%s Geeks Quiz Geeks Quiz"`.

8- 10 50

Question 8 Explanation:

Preprocessor doesn't give any error if we redefine a preprocessor directive. It may give warning though. Preprocessor takes the most recent value before use of and put it in place of a.

9- 100

Question 9 Explanation:

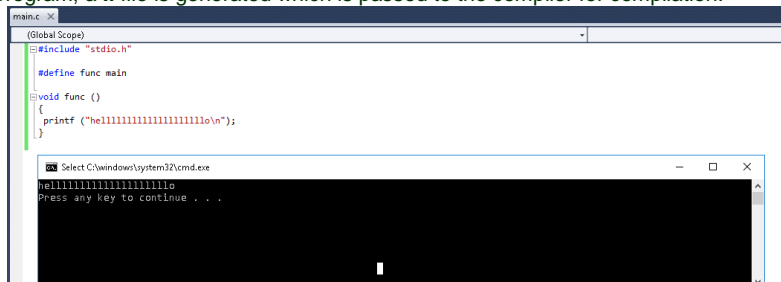
The operator `##` is called "Token-Pasting" or "Merge" Operator. It merges two tokens into one token. So, after preprocessing, the main function becomes as follows, and prints 100.

```
int main()
{
    int var12 = 100;
    printf("%d", var12);
    return 0;
}
```

10- .i

Question 10 Explanation:

After the pre-processing of a C program, a `.i` file is generated which is passed to the compiler for compilation.



Question 11