Explain the reason for your answers.

1.#include<stdio.h> #define FALSE -1 #define NULL 0 #define TRUE 1

int main(){ if(NULL) printf("NULL"); else if(FALSE) printf("TRUE"); else printf("FALSE"); return 0; }

A. FALSE

B. NULL

C. TRUE

D. Compilation Error

**ANSWER: C. TRUE**

**EXPLANATION: conditional statements execute only for true values (non-zero values). NULL is defined as 0, so if statement is not executed. FALSE is defined as -1 which is a non-zero value. So the output is printed as TRUE**

2.#include<stdio.h> int main(){ int i = 0, j = 0; if(i++ == j++) printf("%d %d", i--, j--); else printf("%d %d", i, j); return 0; }

A. 0 0

B. 0 1

C. 1 0

D. 1 1

**ANSWER: D.1 1**

**EXPLANATION: if(i++ == j++) after this line, i=1, j=1**

**printf("%d %d", i--, j--) this line has post increment, so the value of i and j while printing remains 1, 1**

3.#include<stdio.h> int main(){ int i = 0, j = 1, k = 0; if(++k, j, i++) printf("%d %d %d", i, j, k); return 0; }

A. Prints Nothing

B. 1 1 0

C. 0 1 0

D. Compilation Error

**ANSWER: A. Prints Nothing**

**EXPLANATION: if(++k, j, i++) it becomes if(1,1,0) the last value is false so the statement is not executed and nothing is printed.**

4.#include<stdio.h> int main() { char str[8] = "if block"; if(str == "if block") printf("if block executed"); else printf("else block executed"); return 0; }

A. if block executed

B. else block executed

C. Compilation Error

D. None of the above

**ANSWER: B. else block executed**

**EXPLANATION: only characters can be compared using if statement, strings cant be compared**

5.#include<stdio.h> int main() { char str[] = "\0"; if(printf("%s",str)) printf("inside if block"); else printf("inside else block"); return 0; }

A. inside else block

B. inside if block

C. Compilation Error

D. Runtime Error

**ANSWER: A. inside else block**

**EXPLANATION: if(printf("%s",str)) the printf statement prints nothing, which is a false value so if is not executed and the else block is executed**

6.#include<stdio.h> int main() { int i = 5, j = 5; if(i == j); printf("Equal"); else printf("Not Equal"); return 0; } A. Compilation Error

B. Runtime Error

C. Equal

D. Not Equal

**ANSWER: A. Compilation Error**

**EXPLANATION: the if statement is terminated with a semicolon. The else statement is present without an if statement which leads to compilation error**

7.#include<stdio.h> int main(){ float me = 5.25; double you = 5.25; if(me == you) printf("Hello\n"); else break; return 0; }

A. Prints Nothing

B. Hello

C. Runtime Error

D. Compilation Error

**ANSWER: D. Compilation Error**

**EXPLANATION: break statement must be present within loop or switch statement only**

8.#include<stdio.h> int main() { printf("\nab"); printf("\bsi"); printf("\rha"); return 0; }

A. asiha

B. abha

C. aha

D. hai

**ANSWER: D. hai**

**EXPLANATION: first printf statement prints “ab” in second printf \b backspace one character(b) and prints (si), so now the output is “asi”, \r used to position the cursor to the beginning of the current line, so “as” is replaced by “ha” and output becomes “hai”**

9.#include<stdio.h> int main(){ if("May I Get in") printf("yes, Get in"); else printf("No"); return 0; }

A. Compilation Error

B. No

C. yes, Get in

D. None of above

**ANSWER: C. yes, Get in**

**EXPLANATION: the expression inside if is true, so the statement “yes,Get in” is printed**

10.#include<stdio.h> int main(){ int i = 5; if(i == 3, 4) printf("Hai"); else printf("No Hai"); return 0; }

A. Hai

B. No Hai

C. Compilation Error

D. None of the Above

**ANSWER: A. Hai**

**EXPLANATION:** if(i == 3, 4) the first expression is false, so 0, but second expression is true(if(4)) so if block is executed.

11.#include<stdio.h> int main(){ char \*str = {"2braces"}; char \*str1 = {"2braces"}; if(\*str == \*str1) printf("inside if block"); else printf("inside else block"); return 0; }

A. Runtime Error

B. Compilation Error

C. inside if block

D. inside else block

**ANSWER: C. inside if block**

**EXPLANATION: while using pointer character variable, the complete memory location is used to compare the strings**

12.#include<stdio.h> int main() { int i; if(scanf("%d",&i)) //if we give input as 0 printf("inside if block"); else printf("inside else block"); return 0; }

A. Runtime Error

B. Compilation Error

C. inside else block

D. inside if block

**ANSWER: D. inside if block**

**EXPLANATION: scanf value will return as true , so if block is executed**

13.#include<stdio.h> int main() { if(sizeof('\0')) printf("Hai"); else printf("Bye"); return 0; }

A. Bye

B. Hai

C. Compilation Error

D. None

**ANSWER: B. Hai**

**EXPLANATION: ‘\0’ is a character constant, its size is 1. So if block is executed and Hai is printed**

14.#include <stdio.h> int main() { int a=1; printf("%d %d %d\n",++a,a,a++); return 0; }

A. 2,2,2

B. 2,2,3

C. 1,2,3

D. 3,3,1

**ANSWER: D. 3,3,1**

**EXPLANATION: the compiler executes from right to left .**

15.#include<stdio.h> int main() { int i = 1; for(i = 0; i<10; i+3) switch(i) {

case 3:

printf("Hai. This is case 3");

break;

case 6:

printf("Hai. This is case 6");

break;

break;

default:

printf("Hai. This is default");

break;

}

return 0;

}

A. Hai. This is case 3

B. Hai. This is case 6

C. Infinite Execution

D. Hai. This is default

**ANSWER: C. Infinite Execution**

**EXPLANATION: for i=0, the default statement will be executed, in the increment part i+3 is given instead of i=i+3, so the i value is never incremented and the default block executes infinite times.**