

NAME: OGUNLEYE MOFEYISOPE SIMISOLA

MATRIC NUMBER: 19/sci01/099

DEPARTMENT: COMPUTER SCIENCE 200level

15) Identifying the errors in the code. NOTE: The corrections are the bold codes.

```
#include(stdio.h) - #include<stdio.h>
```

```
Int main()
```

```
{
```

```
int counter, N; i, isPrime; - int counter, N, i, isPrime;
```

```
printf("Enter a number");
```

```
scanf("%d", &N);
```

```
Printf("List of prime factors of %d", N); - printf("List of prime factors of %d", N);
```

```
/*Check for every number between 1 to N, whether it divides N*/
```

```
for(counter = 2; conter <= N; counter++); - for(counter = 2; counter <= N;  
counter++)
```

```
{
```

```
/*If counter completely divides N, *then it is a factor of N - /*If counter  
completely divides N, then it is a factor of N*/
```

```
if(N%counter= 0) – if(N%counter == 0)
```

```
{
```

```
/*Check if counter is also a prime number*/
```

```
isPrime == 1; - isPrime = 1;
```

```
for(i = 2; i <= (counter/2); i++)
```

```
{
```

```
if(counter%i == 0)
```

```

{
isPrime = 0;
break – break;
}
}
if(isPrime == 1)
    - {
printf(“%d”, counter);
}
}
- }

return 0;
}

//program to find the prime factor of a number

```

16) Operators in C are the symbols which are used to perform logical and mathematical operations in a C program. Below are the types of operators:

- **Arithmetic operators:** These are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus. Some mathematical operators are +, -, *, /, ++, %, -- and they all have their various functions.
- **Relational operators:** These operators are used to compare the value of two variables. Some relational operators are:

== : this checks if the values of two operands are equal or not. If yes, then the condition becomes true.

!= : this checks if the values of two operands are equal or not.

> : this checks if the values of the left operand is greater than the value of right operand.

< : this checks if the values of the left operand is less than the value of right operand.

>= : this checks if the value of the left operand is greater than or equal to the value of the right operand.

<= : this checks if the value of the left operand is less than or equal to the value of the right operand.

- Logical operators: These operators are used to perform logical operations on the given two variables.

&&: this is called logical AND operator.

||: this is called logical OR operator.

!: this is called logical NOT operator.

- Assignment operators: These are used to assign the values for the variables in C program. Some assignment operators are:

= : Simple assignment operator. Assigns values from right side to left side operand.

+= :Add AND assignment operator. It adds the right operand to the left operand and assigns the result to the left operand.

-=: Subtract AND assignment operator. It subtracts the right operand to the left operand and assigns the result to the left operand.

*= : multiply AND assignment operator. It multiplies the right operand to the left operand and assigns the result to the left operand.

- Bitwise operator: These operators are used to perform bit operations on given two variables. They work on bits and perform bit-by-bit operation. Some bitwise operators are:

& : Binary AND operator copies a bit to the result if it exists in both operands.

| : Binary OR operator copies a bit to the result if it exists in either operands.

^ : Binary XOR operator copies the bit if it is set in one operand but not both.

-

- Conditional operator: This returns one value if condition is true and returns another value if condition is false.
- Special operator: These are &, *, sizeof() operators.

sizeof() : returns the size of the variable.

* : pointer to a variable.

& : returns the address of a variable

- Increment and decrement operator: These are used to either increase or decrease the value of the variable by 1. Increment operator is ++, it means add 1 to the value of the variable. Decrement operator is --, it means remove 1 from the value of the variable. There are 2 types of increment or decrement which are pre increment(++n)/pre decrement(--n) and post increment(n++)/post decrement(n--).

17) #include<stdio.h>

int main()

{

float Interest, principle, time, rate;

printf("Enter the principle amount: ");

scanf("%f", &principle);

printf("Enter time range in years: ");

scanf("%f", &time);

printf("Enter interest rate: ");

scanf("%f", &rate);

Interest = (principle * time * rate)/100;

printf("Interest is %f", Interest);

```
return 0;
```

```
}
```

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

```
int main()
```

```
{
```

```
int age,card, nationality, accredited;
```

```
printf("Enter your age: ");
```

```
scanf("%d", &age);
```

```
printf("Choose an option from the two below to answer the next question\n");
```

```
printf("1. YES\n");
```

```
printf("2. NO\n");
```

```
printf("Do you have voter's card: ");
```

```
scanf("%d", &card);
```

```
printf("Choose an option from the two below to answer the next question\n");
```

```
printf("1. YES\n");
```

```
printf("2. NO\n");
```

```
printf("Are you accredited: ");
```

```
scanf("%d", &accredited);
```

```
printf("Choose an option from the two below to answer the next question\n");
```

```
printf("1. Nigerian\n");
```

```
printf("2. Other\n");
```

```
printf("What is your nationality: ");
```

```
scanf("%d", &nationality);
```

```

if(age >= 18 && card == 1 && accredited == 1 && nationality == 1)
{
    printf("\nYou are eligible to vote");
}
else if(age <= 17 && card == 2 && accredited == 2 && nationality == 2)
{
    printf("\nSorry, You are not eligible to vote");
}
return 0;
}

```

19) Using if statements

```

#include<stdio.h>
int main()
{
    float temp1=50, temp;

    printf("Enter patients temperature: ");
    scanf("%f", &temp);
    if(temp > temp1)
    {
        printf("This patient is sick");
    }
    if(temp == temp1)

```

```
{  
  
    printf("This patient is healthy");  
}  
if(temp < temp1)  
{  
    printf("This patient is sick");  
}  
return 0;
```

ii) **Using if....else statement**

```
#include<stdio.h>  
int main()  
{  
    float temp1=50, temp;  
  
    printf("Enter patients temperature: ");  
    scanf("%f", &temp);  
    if(temp > temp1)  
    {  
        printf("This patient is sick");  
    }  
    else  
    {  
        printf("This patient is healthy");  
    }  
}
```

```
}  
    return 0;
```

```
}
```

20)

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

```
#include<math.h>
```

```
#define p 1000530.00
```

```
#define r 3
```

```
int main()
```

```
{
```

```
    int year;
```

```
    double amount;
```

```
    printf("%4s%21s\n", "Year", "Amount on deposit");
```

```
    for(year = 1; year <= 5; year++)
```

```
    {
```

```
        amount = p * pow(1 + r,year);
```

```
        printf("%4d%21.2f\n", year, amount);
```

```
    }
```

```
    return 0;
```

```
}
```

```
22) 1. #include<stdio.h>
```

```
int main()
```

```
{
```

```
    int A = 30, B = 10, C;
```



```
C = (A < B ? 1 : 2);  
{  
    printf("The answer is %d", C);  
}  
return 0;  
}
```

2. #include<stdio.h>

```
#define True 0
```

```
#define False 1
```

```
int main()
```

```
{
```

```
    int A = 30, B = 10, D;
```

```
    D = (A == B ? True : False);
```

```
{
```

```
    printf("The answer is %d", D);
```

```
}
```

```
return 0;
```

```
}
```

3. #include<stdio.h>

```
int main()
```

```
{
```

```
int A = 30, B = 10, C;  
C = (B < A ? 5 : 14);  
{  
    printf("The answer is %d", C);  
}  
return 0;  
}
```

4. #include<stdio.h>

```
int main()  
{  
  
    int A = 30, B = 10, D;  
D = (A >= B ? 0 : 1);  
{  
    printf("The answer is %d", D);  
}  
    return 0;  
}
```

23) #include<stdio.h>

#include<math.h>

```
int main()
```

```
{
```

```
int a, b, c, d;
double root1, root2;

printf("Enter A, B, and C where a*x*x + b*x + c = 0\n");
scanf("%d%d%d", &a, &b, &c);

d = b*b - 4*a*c;

if(d < 0)
{
    printf("First root = %.2lf + i%.2lf\n", - b/(double)(2*a), sqrt(-d)/(2*a));
    printf("Second root = %.2lf - i%.2lf\n", - b/(double)(2*a), sqrt(-d)/(2*a));
}
else{
    root1 = (-b + sqrt(d))/(2 * a);
    root2 = (-b - sqrt(d))/(2 * a);

    printf("First root = %.2lf\n", root1);
    printf("Second root = %.2lf\n", root2);

}

return 0;
}

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

```
int main()
{
    int sum, product, diff, integer1, integer2;
    float division, remainder;

    integer1 = 80;
    integer2 = 50;

    sum = integer1 + integer2;
    printf("The sum of %d and %d is: %d.\n", integer1 , integer2, sum);

    diff = integer1 - integer2;
    printf("The difference of %d and %d is: %d.\n", integer1 , integer2, diff);

    product = integer1 * integer2;
    printf("The product of %d and %d is: %d.\n", integer1 , integer2, product);

    division = integer1/(float)integer2;
    printf("The division of %d and %d is: %f.\n", integer1 , integer2, division);

    remainder = integer1%integer2;
    printf("The remainder of %d and %d is: %f.\n", integer1 , integer2, remainder);

    return 0;
}
```

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

```
int main()
```

```
{
```

```
    printf("***** ** * ** *****");
```

```
}
```

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

```
int main()
```

```
{
```

```
    printf("***** ");
```

```
    printf("*** ");
```

```
    printf("*** ");
```

```
    printf("** ");
```

```
    printf("*** ");
```

```
    printf("** ");
```

```
    printf("* ");
```

```
    printf("*****");
```

```
}
```

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

```
int main()
```

```
{
```

```
    int num1;
```

```
    float num2;
```

```
double num3;
printf("Enter an integer value: ");
scanf("%d", &num1);
printf("The integer number is %d\n", num1);

printf("Enter a float value: ");
scanf("%f", &num2);
printf("The float number is %f\n", num2);

printf("Enter a double value: ");
scanf("%lf", &num3);
printf("The double number is %lf", num3);

return 0;

}
```