

question 1.

sol:-

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

```
#include<conio.h>
```

```
int main() {  
    float price,tax;  
    printf("enter the price");  
    scanf("%f",&price);  
    printf("enter the tax");  
    scanf("%f",&tax);  
    float FP=price+(price*tax/100);  
    printf("%f is the final price of the product",FP);  
  
    return 0;  
}
```

question 2,

sol:-

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

```
int main() {  
    double wagesPerHour, weeklyWages;  
    int hoursWorked;  
  
    printf("Enter wages per hour: ");  
    scanf("%lf", &wagesPerHour);
```

```

printf("Enter hours worked: ");
scanf("%d", &hoursWorked);

if (hoursWorked <= 30) {
    weeklyWages = wagesPerHour * hoursWorked;
} else {
    int regularHours = 30;
    int overtimeHours = hoursWorked - regularHours;
    double overtimeWages = 2 * wagesPerHour;

    weeklyWages = (wagesPerHour * regularHours) + (overtimeWages * overtimeHours);
}

// Display the weekly wages
printf("Weekly wages: %.2lf\n", weeklyWages);

return 0;
}

```

question3.

sol:-

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

```
int main()
```

```
{
```

```
    int C=500,a=50,b=35,c=10,d=15;
```

```
    printf("the price of 2 kg apple is %d",a);
```

```
        printf("\nthe price of 1.5 kg mango is %d",b);
```

```

printf("\nthe price of 2.5 kg potato is %d",c);

printf("\nthe price of 1kg tomato is %d",d);

int TMR=C-(a+b+c+d);

printf("\ntotal money returned will be %d",TMR);

return 0;

}

```

question 4,

sol:-

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

```
int main()
```

```
{
```

```
printf("NAME:-HARSH CHAUHAN");
```

```
printf("\nDATE OF BIRTH:-28/10/2005");
```

```
printf("\nMOBILE NO.:-821XX350XX");
```

```
}
```

question 5.

sol:-#include <stdio.h>

```
int main() {
```

```
    int integer;
```

```
    char character;
```

```
float floating;
```

```
printf("Enter an integer: ");
```

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

```
printf("Enter a character: ");
```

```
scanf(" %c", &character);
```

```
printf("Enter a float value: ");
```

```
scanf("%f", &floating);
```

```
printf("You entered:\n");
```

```
printf("Integer: %d\n", integer);
```

```
printf("Character: %c\n", character);
```

```
printf("Float: %.2f\n", floating);
```

```
return 0;
```

```
}
```

question 6.

sol:-

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

```
int main() {
```

```
    double cost = 172.53;
```

```
printf("The sales total is : $ %.2f in c\n", cost);

return 0;
}
```

question no. 7

sol:-

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

```
int main() {
    int apples_per_person = 6;
    double half_apples = 0.5;

    double total_apples = (apples_per_person + half_apples) * 3;

    printf("Raju has a total of %.1f apples.\n", total_apples);

    return 0;
}
```

question no. 8

sol:-

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

```
int main() {  
float a=69.364582148184;  
printf("the value of float a upto 2 decimal places is %.2f",a);  
  
    return 0;  
}
```

question no.9

sol:-

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

```
int main() {  
double a;  
printf("enter your mobile no.");  
scanf("%lf",&a);  
  
printf("your mobile no. is %.0lf",a);  
  
    return 0;  
}
```

question 10.

sol:-#include <stdio.h>

```
int main() {  
int a=30000,b=a+(a*20/100),c=b+(b*30/100);  
printf("total population of city : %d",a);
```

```
printf("\nincrement in population of city in one year: %d",b);  
printf("\nincrement population of city in second year:%d ",c);  
printf(" \ntotal population of city after second year %d",c);
```

```
return 0;  
}
```

question 11.

sol:-

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

```
int main() {  
printf("enter a character: ");  
char a;  
scanf("%c",&a);  
printf("%d",a);
```

```
return 0;  
}
```

question no.12

sol:-

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

```
int main() {  
int a;
```

```

printf("Enter the basic pay of an employee: ");
scanf("%d",&a);
int HRA=a*0.15;
int TA=a*0.20;
int b=HRA+TA+a;
printf("\ntotal salary of the employee %d",b);

return 0;
}

```

question no. 13.

sol:-

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

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

```
int main() {
```

```
    double xp, yp, xq, yq, slope, angle;
```

```
    printf("Enter the coordinates of point P (xp yp): ");
```

```
    scanf("%lf %lf", &xp, &yp);
```

```
    printf("Enter the coordinates of point Q (xq yq): ");
```

```
    scanf("%lf %lf", &xq, &yq);
```

```
    slope = (yq - yp) / (xq - xp);
```



```

angle = atan(slope) * 180.0 / M_PI;

printf("Slope of the line: %.2lf\n", slope);
printf("Angle of inclination (degrees): %.2lf\n", angle);

return 0;
}

```

question no.15

sol:-

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

```

int main() {
    float frequency,wavelength,speed;
    printf("enter wavelength in Å: ");
    scanf("%f",&wavelength);
    printf("enter speed: ");
    scanf("%f",&speed);
    frequency=speed/wavelength;
    printf("The frequency of the wave is %.2lf Hz.\n", frequency);

    return 0;
}

```

question no.16

sol:-

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

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

```
int main() {
```

```
    double u = 30.0;
```

```
    double a = 5.0;
```

```
    double s = 70.0;
```

```
    double v;
```

```
    v = sqrt(u * u + 2 * a * s);
```

```
    printf("The final velocity of the car is %.2lf m/s.\n", v);
```

```
    return 0;
```

```
}
```

question no.17

sol:-

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

```
int main() {
```

```
    double u = 0.0;
```

```
    double a = 4.0;
```

```

double t = 3.0;

double v;

double s;

// Calculate the final velocity using  $v = u + at$ 
v = u + (a * t);

// Calculate the distance traveled using  $s = ut + 0.5 * a * t^2$ 
s = (u * t) + (0.5 * a * t * t);

printf("(a) The final velocity of the horse is %.2lf m/s.\n", v);
printf("(b) The distance traveled by the horse is %.2lf meters.\n", s);

return 0;
}

```

question no.18

sol:-

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

```

int main() {
    int n=20050089,count=0,r;
    for(int i=1;i<=4;i++)
    {
        r=n%10;
        count+=r;
        n=n/10;
    }
}

```

```
printf("sum of last 4 digits %d",count);  
    return 0;  
}
```

question no.19

sol:-

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

```
int main() {
```

```
    double heightCm = 175.0;
```

```
    double weightKg = 70.0;
```

```
    const double CM_TO_INCHES = 0.393701;
```

```
    const double KG_TO_POUNDS = 2.20462;
```

```
    double heightFeet = heightCm * CM_TO_INCHES / 12.0;
```

```
    double weightPounds = weightKg * KG_TO_POUNDS;
```

```
    printf("Height: %.2lf cm is %.2lf feet\n", heightCm, heightFeet);
```

```
    printf("Weight: %.2lf kg is %.2lf pounds\n", weightKg, weightPounds);
```

```
    return 0;
```

```
}
```

question no.20

sol:-

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

```
int main() {
```

```
    char option;
```

```
    int sum = 0;
```

```
    float product = 1.0;
```

```
        return 0;
```

```
}
```

question no.21

sol:-

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

```
int main() {
```

```
    int numbers[9];
```

```
    printf("Enter nine integers, one at a time:\n");
```

```
    for (int i = 0; i < 9; i++) {
```

```
        scanf("%d", &numbers[i]);
```

```
    }
```

```

printf("The numbers in groups of three separated by commas:\n");

for (int i = 0; i < 9; i++) {
    printf("%d", numbers[i]);

    if ((i + 1) % 3 != 0) {
        printf(", ");
    }

    if ((i + 1) % 3 == 0) {
        printf("\n");
    }
}

return 0;
}

```

question no.22

sol:-Header files in C programming are files that contain declarations and definitions of functions, variables, macros, and other constructs used in C programs. These files typically have a ".h" extension and are included in C source code files using the #include preprocessor directive.

The main uses of header files in C programming are as follows:

**Modularization:** Header files help in modularizing your code by separating declarations (interface) from definitions (implementation). This allows you to divide your code into manageable and reusable components.

**Function Prototypes:** Header files contain function prototypes (declarations), which provide information about the functions used in your program, including their names, return types, and parameter lists. This enables the compiler to check the correctness of function calls.

**Global Variables:** You can declare global variables in header files, allowing them to be shared across multiple source files. To prevent multiple definitions of these global variables, you typically use the `extern` keyword in the header file and provide the actual definition in a single source file.

**Constants and Macros:** Header files often contain constant definitions using `#define` and macro definitions. This allows you to define symbolic names for constants and macros, making your code more readable and maintainable.

**Library Interfaces:** Header files are commonly used to declare the interfaces of standard C libraries and user-defined libraries. Including the appropriate header file gives your code access to the functions and types defined in those libraries.

**Avoid Code Duplication:** Header files allow you to write reusable code that can be included in multiple source files. This helps avoid code duplication and promotes code reusability.

**Ease of Maintenance:** Separating declarations in header files from the actual code in source files makes it easier to maintain and update your code. Changes to function prototypes and variable declarations only need to be made in one place (the header file), which propagates to all source files that include it.

Commonly used standard C header files include `<stdio.h>` for input and output functions, `<stdlib.h>` for general utilities, `<math.h>` for mathematical functions, and many others.

question no.23

sol:-OUTPUT WILL BE:-

56    70    38

question no. 24

sol:-

OUTPUT WILL BE:-

GLA UNIVERSITY15

question no. 25

sol:-

Library functions in C are pre-defined functions that are part of the C Standard Library (also known as the Standard Library) and are accessible by including the appropriate header files. These functions perform common tasks and operations, making it easier for programmers to develop C programs without having to write code from scratch for every task. Here are four commonly used library functions in C: printf, scanf, strlen, rand.

question no.26

sol:-

OUTPUT WILL BE :-31 37 1f

question no.27

sol:-

The statement `printf("%d", scanf("%d%d", &a, &b));` is a bit tricky and can be confusing because it involves nested functions and the return values of `printf` and `scanf`.

question no.28

sol:- compile error is there because of void main if we replace void with int then output will be "C % FOR % PLACEMENT".



question no.29

sol:-

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

```
int main() {
```

```
    double distance;
```

```
    double time = 4.0;
```

```
    printf("Enter the distance between GLA University and Delhi (in kilometers): ");
```

```
    scanf("%lf", &distance);
```

```
    double speed = distance / time;
```

```
    printf("The speed of the bus is %.2lf kilometers per hour.\n", speed);
```

```
    return 0;
```

```
}
```

question no.30

sol:-

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

```
int main() {
```

```
    int satyamMarks = 50;
```

```
    int sumanMarks = 70;
```

```
int shyamMarks = 80;

double averageMarks = (satyamMarks + sumanMarks + shyamMarks) / 3.0;

printf("Average marks of Satyam, Suman, and Shyam: %.2lf\n", averageMarks);

return 0;
}
```

question no. 31

sol:-

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

```
int main() {
    int a,b, temp;

    printf("Enter the amount of money given to Saurav: ");
    scanf("%d", &a);

    printf("Enter the amount of money given to Sajal: ");
    scanf("%d", &b);

    // Swap the money between Saurav and Sajal using a temporary variable
    temp = a;
    a = b;
```

```
b = temp;

// Display the corrected amounts of money
printf("After rectifying the mistake:\n");
printf("Amount of money for Saurav: %d\n", a);
printf("Amount of money for Sajal: %d\n", b);

return 0;
}
```

question no.32

sol:-

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

```
int main() {
    int speed=4,time=3,distance;
    distance= speed*time;
    printf("distance travelled by me is %d metre ",distance);
    return 0;
}
```

question no.33

sol:-Yes, you can combine multiple escape sequences, such as `\n` (newline) and `\t` (tab), in a single line of program code in C or many other programming languages. Escape sequences are used to represent special characters and control characters in string literals.

question no.34

sol:-

Comments in a C program are used to provide explanations, documentation, or notes within the source code. Comments are ignored by the C compiler and have no effect on the program's execution. They are intended for human readers, including programmers, to better understand the code. Comments are a crucial part of writing clean, maintainable, and understandable code.

.  
.  
.

.we can use `'/'` to do single comment and for multiple line comment we use `'/*'` at both i.e starting and ending of line.

question no.35:-. What is wrong in this statement? `scanf("%d",number);`

sol:-we have to insert `'&'` before the `'number'` statment....after putting the `'&'`:- the statement look like :-  
`scanf("%d",&number);` .

question no.36

sol:-OUTPUT WILL BE :- YES

However, it's important to note that the behavior of this code can be influenced by the specific compiler and platform you are using. In practice, on most common platforms and compilers, `sizeof(int)` will be greater than -1, and "Yes" will be printed.

question no.37

sol:-gross-salary: This variable name contains a hyphen, which is not allowed. Therefore, it is invalid.

question no.38

sol:-#include <stdio.h>

```
int main() {  
    float a=175,b=25,c;  
    printf("total galloons ,that toms has to drain :%.0f",a);  
    printf("\namount of galloons that he able to drain in 1hr: %.0f",b);  
    c=a/b;  
    printf("\ntotal time to drain 175 galloons : %.0f hours",c);  
    return 0;  
}
```

question no.39

sol:-#include <stdio.h>

```
int main() {  
    // 75% means 0.75  
    float y=0.75,x;  
    // given eq=  $y=-0.2x+1$   
    // we have to find x.  
    x=(y-1)/-0.2;  
    printf("%.3f will be the hours at 75% battery",x);  
    return 0;  
}
```

question no.40

sol:- (A)compiler.

question 41.

sol:-(C)%o

question 42.

sol:-(B)%.2f

question no.43

sol:-(B)array

question no.44

sol:-(C)"hell"8

question no. 45

sol:-

question no.46

sol:-(B)basic\_pay

question no. 47

sol:-(A)c1.

question no.53

sol:-32770

question no. 54

sol:-temperature in fahrenheit is 41.00

