1. Answers:
    a. 3
    b. 3
    c. 1
    d. 4
    e. FALSE
    f. 3
    g. Answers:
        i. 12345
        ii. 10
        iii. 10
        iv. Value at address 10 or Garbage value or error
2. Answers:
    a. Output: DEF
    Explanation: The conditional operator takes an expression and executes the first statement if the expression evaluates to be true, and the second statement if the expression evaluates to be false. In this case, variable i's value is checked whether equal to 11 or not. Since it was initialized to be 10, so, the expression is evaluated to be FALSE. So DEF is printed.
    b. Output: Some garbage value 200
    Explanation: Integer variables a, b, c are declared and a is initialized to 300. In the if statement, a's value is checked, whether it is greater than or equal to 400 or not. Since it is not, the single statement in the if-block is not executed. (if there are no curly braces, only one statement is allowed in if-block). The next statement is executed and c is initialized to 200. Next, the printf() statement is executed. Since the auto variable b is not initialized with any particular value, it contains garbage value. So first garbage is printed, next the value of c, that is 200 is printed.
    c. Output: 1 1 1 1 1… infinite number of times
    Explanation: It is an infinite while loop, since the value of the counter variable I is never updated, so the loop condition is always satisfied, as 1 is always <=10. So 1 is printed an infinite number of times.
    d. Output: 2 3 4 5 6
    Explanation: Following for loop execution, in the first iteration, first the loop counter variable i is initialized to 1. Next it is checked if i<=5, which is satisfied. Next the single statement in the for loop is executed and i is incremented by 1, so i value becomes 2. Next the printf() is executed, so 2 is printed. In the second iteration, it is checked if i<=5, which is satisfied. Next the single statement in the for loop is executed and i is incremented by 1, so i value becomes 3. Next the printf() is executed, so 3 is printed and so on. In the last iteration, i value is incremented to 6 and i is printed. In the next check, i is not <=5, so the loop terminates.
    e. Output: 5 2
    Explanation: Pointers to variables i and j are passed from main() to func(). These addresses are stored in pointer variables i and j in func() – these variables are local to func(). The calculations done in func() use the values of variables i and j of main(), but store them in the local variables i and j of func() – note that it is not *i = *i**i; So, no changes to the values of variables i and j are made in main(). So, the original values if variables i and j are printed – 5 2.

3.
```c
#include <stdio.h>
//Swapping two variables without using third variable, using call by reference
void swap_r( float * i, float * j ){
    *i = *i+*j;
    *j = *i-*j;
    *i = *i-*j;
}
int main(){
float a, b;
printf("Please enter the two float variables to be swapped: ");
scanf("%f %f", &a, &b);
printf("\nThe float variables before swapping are a = %f and b = %f",a,b);
swap_r( &a, &b ) ; //calling function using reference
printf("\nThe float variables after swapping are a = %f and b = %f",a,b);
return 0;
}
```

4.
```c
#include <stdio.h>
int main(){
int a; //input integer
int op=0, ep=0, on=0, en=0, z=0;//counters
char option; //to store option whether user wants to continue or exit
do{//controlled inifinite loop based on user choice
    printf("Please enter any integer: ");
    scanf("%d",&a);
    if(a==0)
        z++;
    else if((a>0)&&(a%2==0))
        ep++;
    else if((a>0)&&(a%2!=0))
        op++;
    else if((a<0)&&(a%2==0))
        en++;
    else if((a<0)&&(a%2!=0))
        on++;
    printf("\nDo you want to continue (y/n): ")   ;
    scanf(" %c",&option);
}while(option=='y');//checking user option
printf("\nThe count of zeros = %d",z);
printf("\nThe count of odd positive = %d",op);
printf("\nThe count of even positive = %d",ep);
printf("\nThe count of odd negative = %d",on);
printf("\nThe count of even negative = %d",en);
return 0;}
```

5.

```c
#include <stdio.h>
int fibonacci(int n){  // fibonacci() funtion definition
    if (n == 1)     // first stopping condition check
        return 0;
    else if (n == 2)// second stopping condition check
        return 1;
    // else calling the fibonacci() function recursively till stopping conditions
    else
        return fibonacci(n-1) + fibonacci(n-2);
}

int main(){
    int num; // variable to store how many terms to be displayed in the series
    printf("Enter the number of terms to be printed in the series : ");
    scanf("%d", &num); // taking user input
    int i;
    for (i=1;i<=num;i++)
    {
        printf("%d  ", fibonacci(i)); // calling fibonacci() function for each iteration
            and printing the returned value
    }
    return 0;
}
```