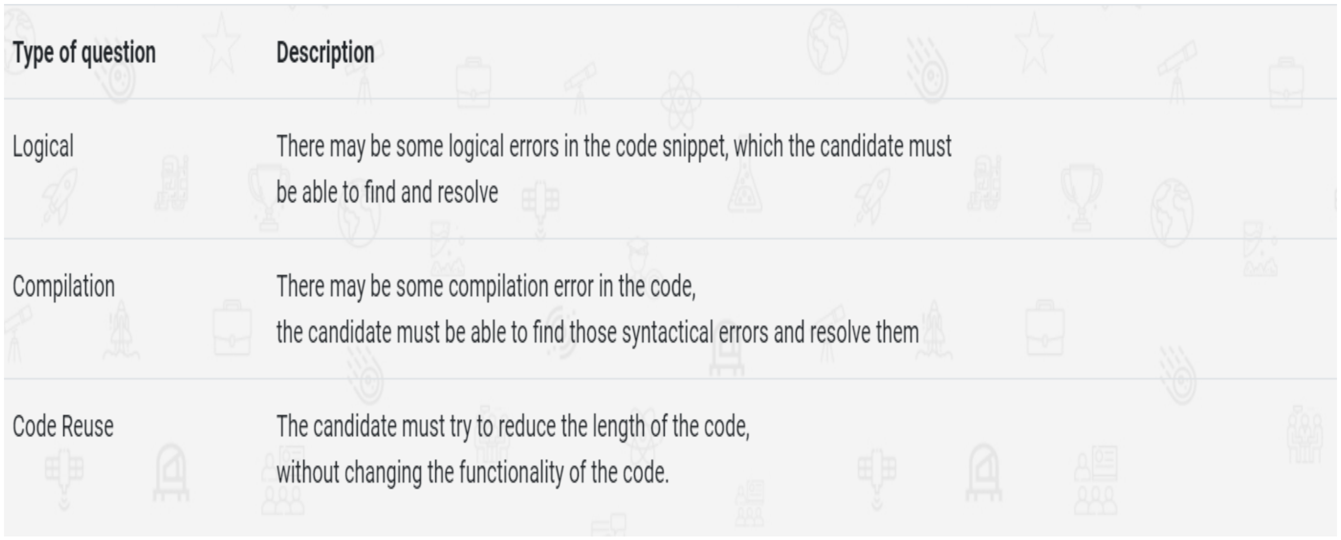
Automata Fix

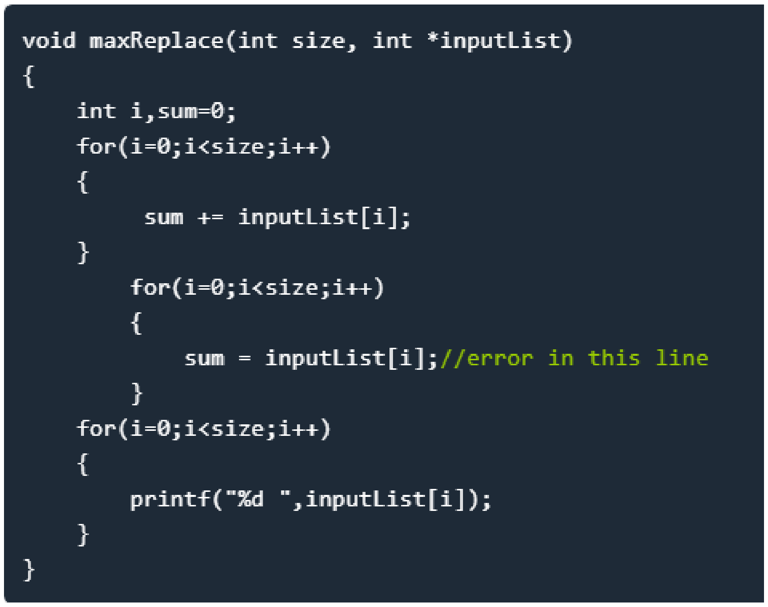
Types:



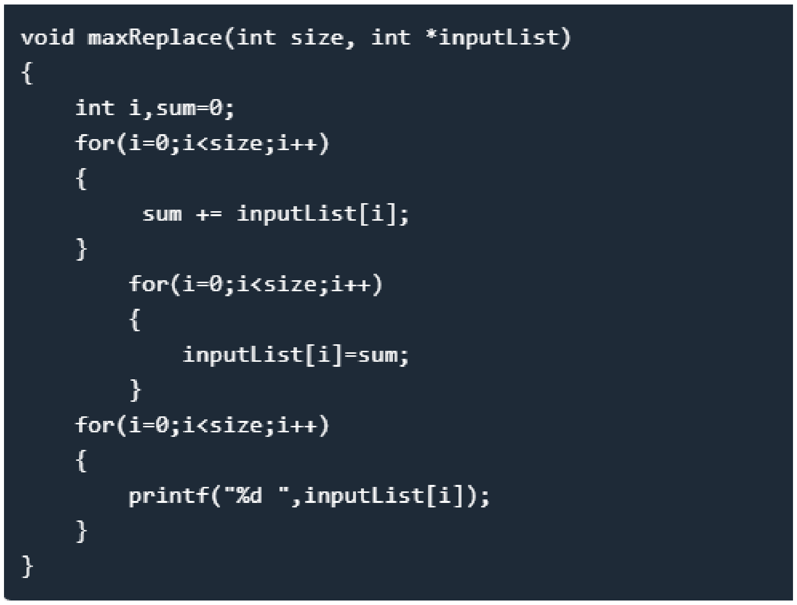
Type of question	Description
Logical	There may be some logical errors in the code snippet, which the candidate must be able to find and resolve
Compilation	There may be some compilation error in the code, the candidate must be able to find those syntactical errors and resolve them
Code Reuse	The candidate must try to reduce the length of the code, without changing the functionality of the code.

Question 1

1. The function **maxReplace** print space separated integers representing the input list after replacing all the elements of the input list with the sum of all the element of the input list.
2. The function **maxReplace** accepts two arguments – *size* an integer representing the size of the input list and *inputList*, a list of integers representing the input list respectively.
3. The function **maxReplace** compiles unsuccessfully due to compilation errors. Your task is to fix the code so that it passes all the test cases.



```c
void maxReplace(int size, int *inputList)
{
    int i,sum=0;
    for(i=0;i<size;i++)
    {
         sum += inputList[i];
    }
        for(i=0;i<size;i++)
        {
            sum = inputList[i];//error in this line
        }
    for(i=0;i<size;i++)
    {
        printf("%d ",inputList[i]);
    }
}
```



```c
void maxReplace(int size, int *inputList)
{
    int i,sum=0;
    for(i=0;i<size;i++)
    {
         sum += inputList[i];
    }
        for(i=0;i<size;i++)
        {
            inputList[i]=sum;
        }
    for(i=0;i<size;i++)
    {
        printf("%d ",inputList[i]);
    }
}
```

Question 2:

Check for syntax error/ logical error and correct the error to get the desired output.

Given n, print 0 to n by identifying whether the n is even or odd.

Test Case : 1

n : 10

Output

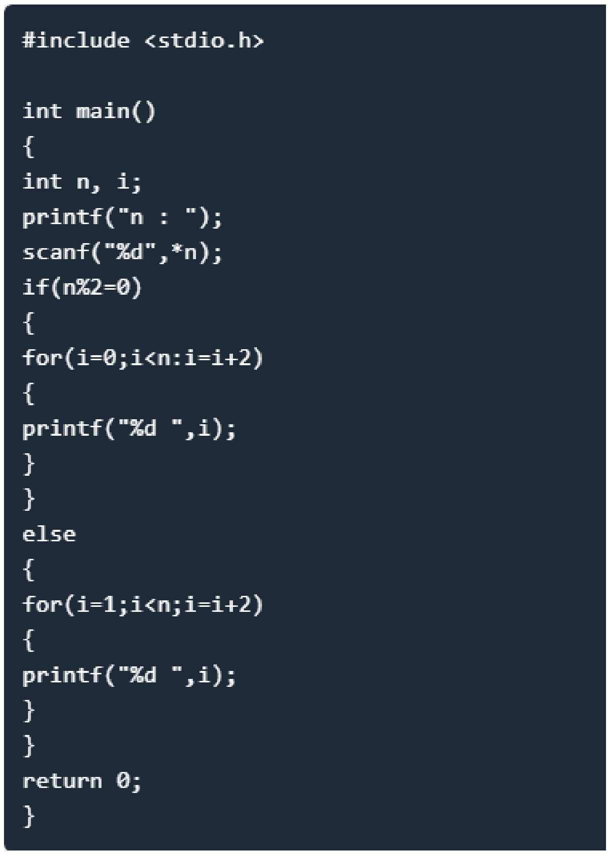
```
0  2  4  6  8  10
```

Test Case : 2

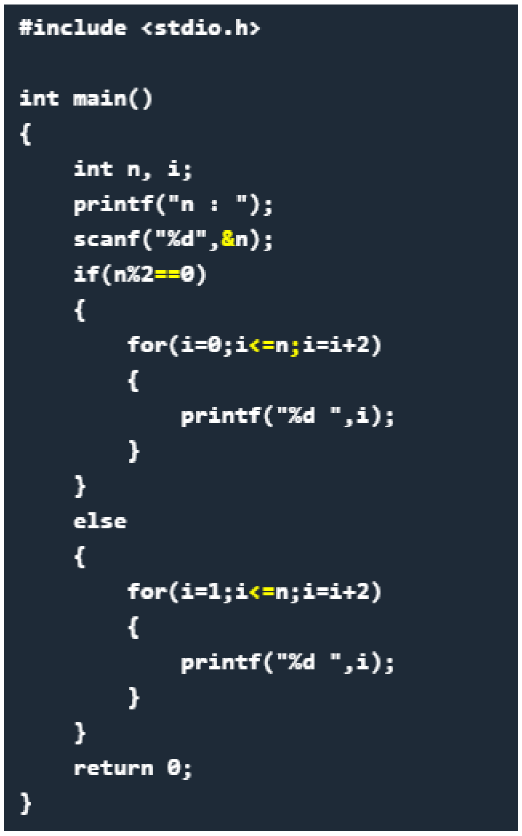
n : 9

Output

```
   1   3  5  7  9
```



```
#include <stdio.h>

int main()
{
int n, i;
printf("n : ");
scanf("%d",*n);
if(n%2=0)
{
for(i=0;i<n:i=i+2)
{
printf("%d ",i);
}
}
else
{
for(i=1;i<n;i=i+2)
{
printf("%d ",i);
}
}
return 0;
}
```



```
#include <stdio.h>

int main()
{
    int n, i;
    printf("n : ");
    scanf("%d",&n);
    if(n%2==0)
    {
        for(i=0;i<=n;i=i+2)
        {
            printf("%d ",i);
        }
    }
    else
    {
        for(i=1;i<=n;i=i+2)
        {
            printf("%d ",i);
        }
    }
    return 0;
}
```

Question 3:

Find the factorial of the given number.

```
#include<stdio.h>
int main()
{
 int i,fact=1,number;
 printf("Enter a number: ");
  scanf("%d",&number);
    for(i=1;i<=number;i++){
      fact=fact*i;
  }
  printf("Factorial of %d is: %d",number,fact);
return 0;
}
```

```
long factorial(int n)
{
  if (n == 0)
    return 1;
  else
    return(n * factorial(n-1));
}
```

Question 4:

Given n, print from n to 0

```
int main()
{
  int n;
  scanf("%d", &n);
  unsigned int i = n;
  while(i >= 0)
  {
       printf("%dn", i);
       i--;
  }
 return 0;
}
```

Input: 4

Output: Infinite loop

Answer: Error - Logical error

unsigned int i = n; unsigned integer ranges from 0 to 65535, which will be taken in the cyclic order.Soi-- will keep repeating in a cyclic way. The loop will never be terminated. So it should be written as **int i = n;**

Question 5:

2) Find the factorial of a given number.

```
int main()
{
long int fact = 1, n, i;
scanf("%d", &n);

for(i =1; i <= n; i++)
{
fact = fact * i;
}
printf("%d", fact);
return 0;
}
```

Input: 20

Output: -2102132736

Answer: Error Logical error

The fact and n are declare d as long int, so in scanf and printf%ld should be used in place of %d.

Question 6:

3) Check whether the below program print the below pattern

1111

222

33

```
void main()
{
 int i, j, n;
 scanf("%d", &n);
 for(i = 1; i<=n; i++)
 {
   for(j = 1; j<=n; j++)
   {
       printf("%d", i);
   }
   printf("\n");
 }
}
```

Input: 3

Output:

111

222

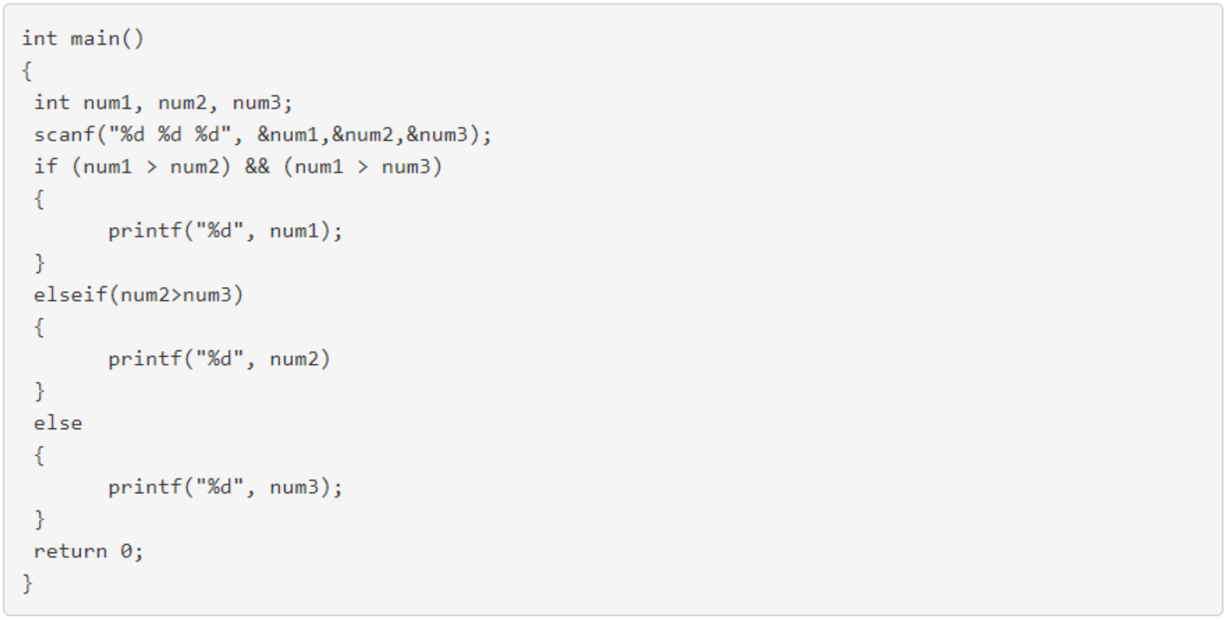
333

Answer: Error: Logical error

The inner for loop has to be written in this way: **for(j = i-1; j<n+1; j++)**

Question 7:

4) Find the greatest of three numbers.



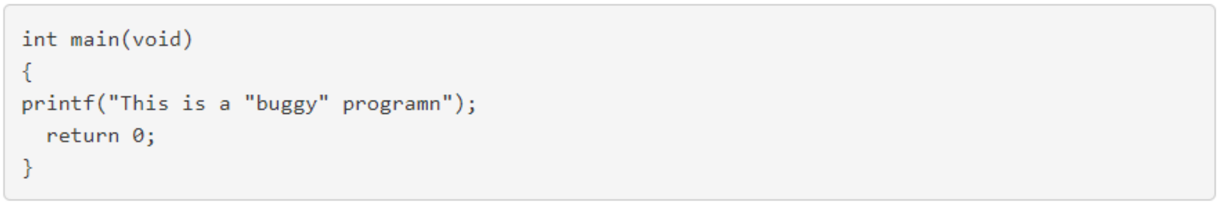
```
int main()
{
 int num1, num2, num3;
 scanf("%d %d %d", &num1,&num2,&num3);
 if (num1 > num2) && (num1 > num3)
 {
       printf("%d", num1);
 }
 elseif(num2>num3)
 {
       printf("%d", num2)
 }
 else
 {
       printf("%d", num3);
 }
 return 0;
}
```

Answer: Error: Syntax error

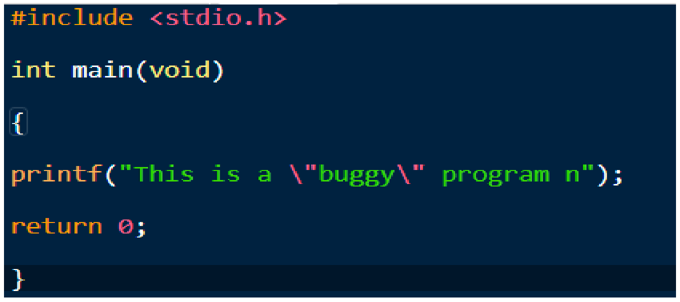
if (num1 > num2) && (num1 > num3) it has to be written as if ((num1 > num2) && (num1 > num3)) and this line **else if** (num2>num3) should be rewritten as **else if** (num2>num3)

Question 8:

5) Fix the error, recompile and match against the output provided.



```
int main(void)
{
printf("This is a "buggy" programn");
  return 0;
}
```



```
#include <stdio.h>
int main(void)
{
printf("This is a \"buggy\" program n");
return 0;
}
```

Question 9:

6) Code reuse: Convert Binary to Decimal by using the existing function.

```
voidbinarytodecimal(number)
{
       // Type your code here
}
void main()
{
  int num;
  scanf("%d", &num);
  printf(%d, binarytodecimal(num);
}
```

Answer:

```
voidbinarytodecimal(number)
{
 int dval=0, base=1, rem;
 while(number > 0)
 {
       rem = number % 10;
       dval = dval + rem * base;
         num = number / 10;
         base = base * 2;
 }
 return dval;
}
```

Question 10:

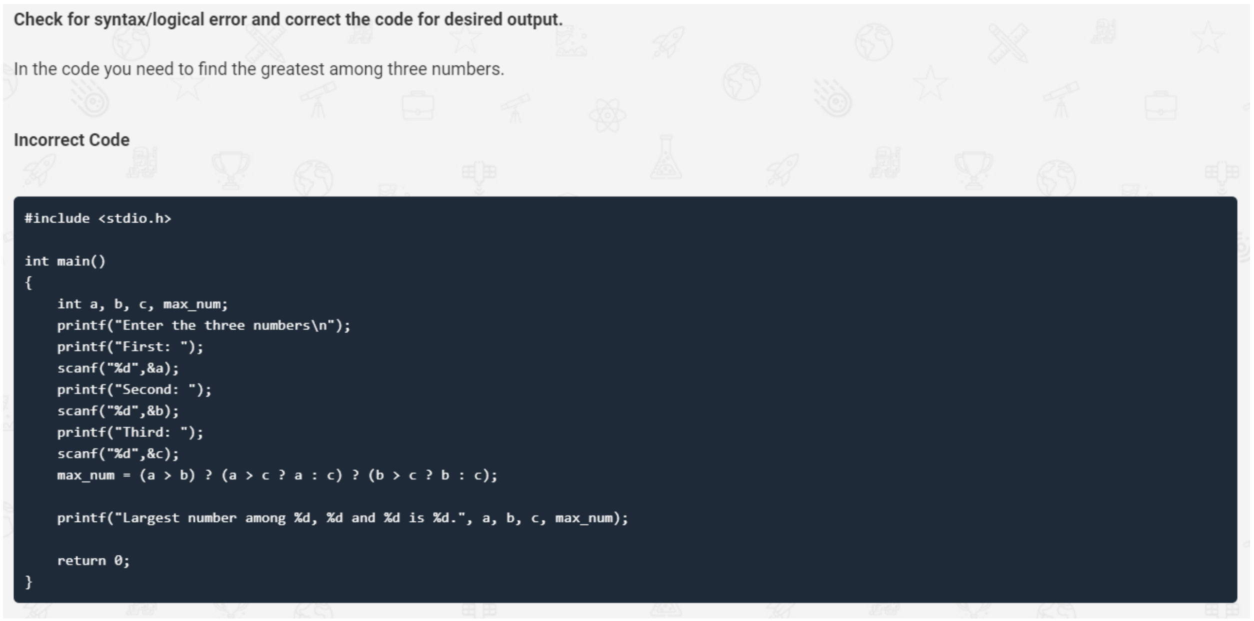
7) Print the prime numbers from an array up to given value n by using existing function.

```
int isprime(int num)
{
// type your code here
}
int main()
{
int n, m, arr[100], size=0, i;
scanf("%d", &n);
for(m = 2; m <= n; m++)
{
if(isprime(m))
{
  arr[size++]= m;
}
}
for(i = 0; i < size; i++)
{
printf("%dn", arr[i]);
}
return 0;
}
```

Answer:

```
int isprime(int num)
{
 int i;
 int isprime = 1;
 for(i = 2; i <= num / 2; i++)
 {
     if(num % i == 0)
     {
            isprime = 0;
               break;
     }
 }
 return isprime;
}
```

Question 11:

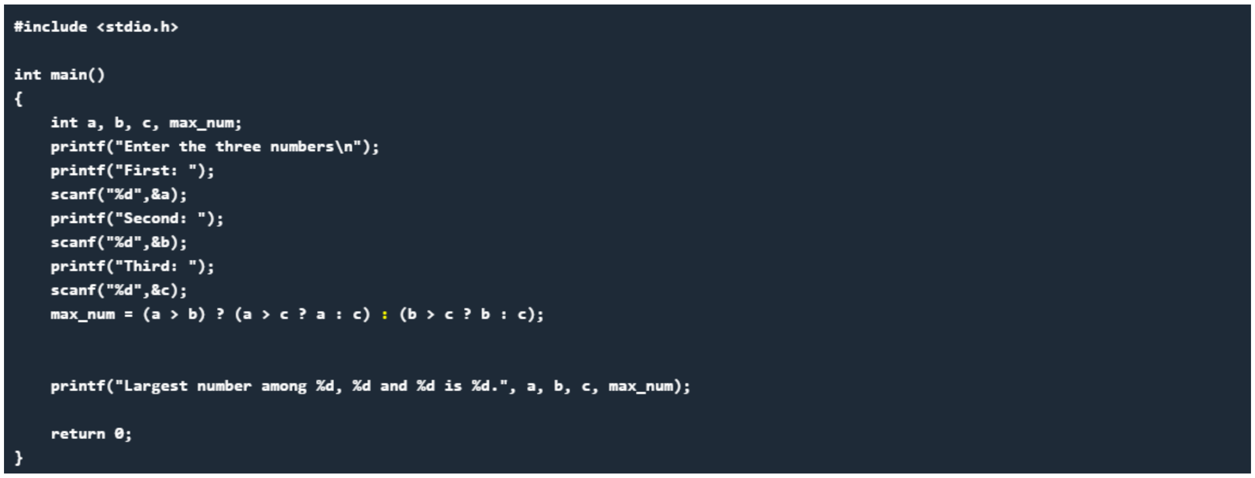


Check for syntax/logical error and correct the code for desired output.

In the code you need to find the greatest among three numbers.

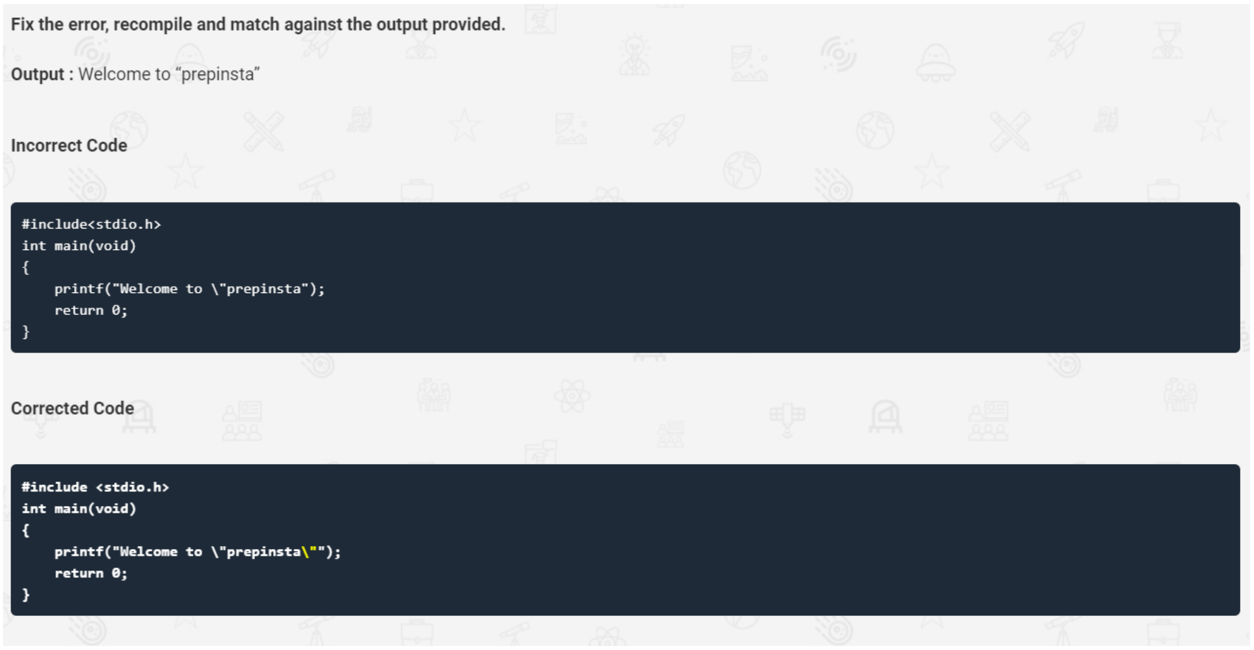
Incorrect Code

```
#include <stdio.h>

int main()
{
    int a, b, c, max_num;
    printf("Enter the three numbers\n");
    printf("First: ");
    scanf("%d",&a);
    printf("Second: ");
    scanf("%d",&b);
    printf("Third: ");
    scanf("%d",&c);
    max_num = (a > b) ? (a > c ? a : c) ? (b > c ? b : c);

    printf("Largest number among %d, %d and %d is %d.", a, b, c, max_num);

    return 0;
}
```



```
#include <stdio.h>

int main()
{
    int a, b, c, max_num;
    printf("Enter the three numbers\n");
    printf("First: ");
    scanf("%d",&a);
    printf("Second: ");
    scanf("%d",&b);
    printf("Third: ");
    scanf("%d",&c);
    max_num = (a > b) ? (a > c ? a : c) : (b > c ? b : c);


    printf("Largest number among %d, %d and %d is %d.", a, b, c, max_num);

    return 0;
}
```

Question 12:



Fix the error, recompile and match against the output provided.

Output : Welcome to “prepinsta”

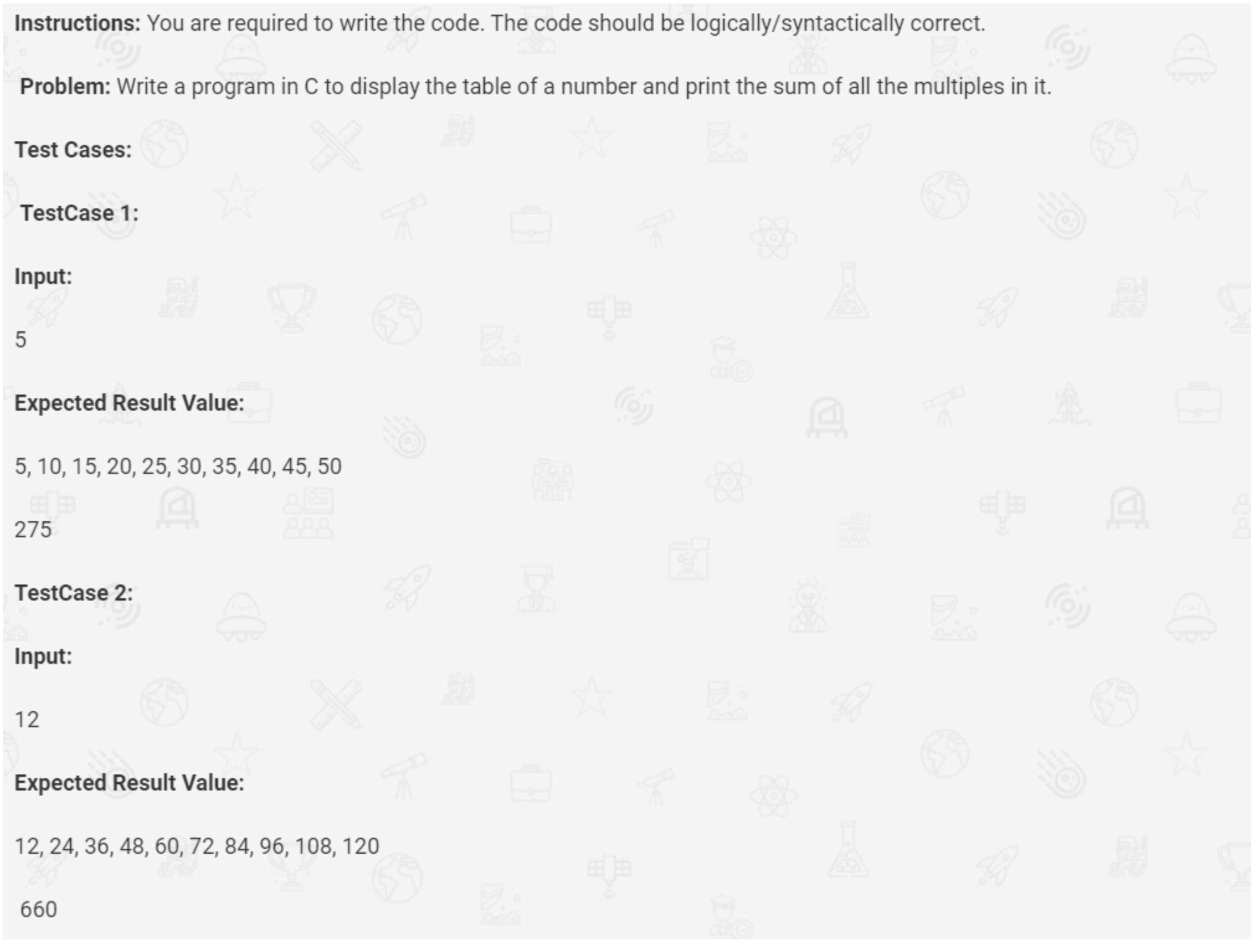
Incorrect Code

```
#include<stdio.h>
int main(void)
{
    printf("Welcome to \"prepinsta");
    return 0;
}
```

Corrected Code

```
#include <stdio.h>
int main(void)
{
    printf("Welcome to \"prepinsta\"");
    return 0;
}
```

Question 13:



Instructions: You are required to write the code. The code should be logically/syntactically correct.

Problem: Write a program in C to display the table of a number and print the sum of all the multiples in it.

Test Cases:

TestCase 1:

Input:

```
5
```

Expected Result Value:

```
5, 10, 15, 20, 25, 30, 35, 40, 45, 50

275
```

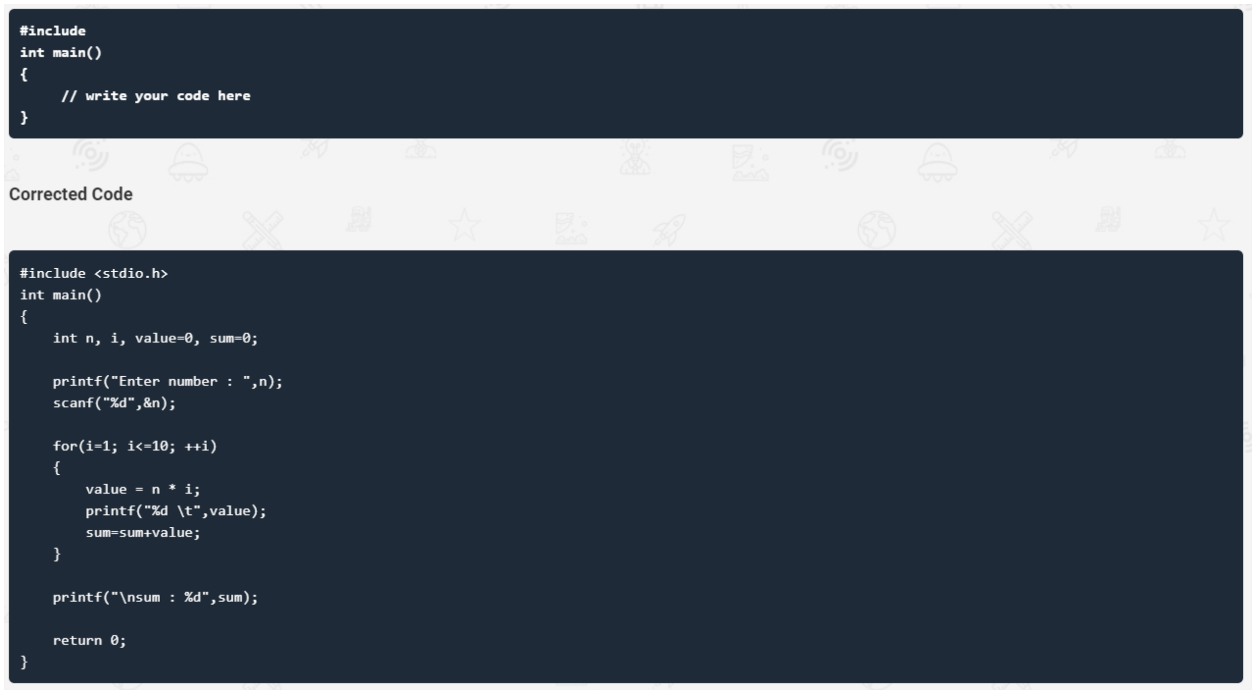
TestCase 2:

Input:

```
12
```

Expected Result Value:

```
12, 24, 36, 48, 60, 72, 84, 96, 108, 120

 660
```



```
#include
int main()
{
    // write your code here
}
```

Corrected Code

```
#include <stdio.h>
int main()
{
    int n, i, value=0, sum=0;

    printf("Enter number : ",n);
    scanf("%d",&n);

    for(i=1; i<=10; ++i)
    {
        value = n * i;
        printf("%d \t",value);
        sum=sum+value;
    }

    printf("\nsum : %d",sum);

    return 0;
}
```

Question 14:

Question

You have to find the security key from the data transmitted.

Input

The input consists of an integer data, representing the data to be transmitted.

Output

Print an integer representing the security key for the given data.

Example

Input

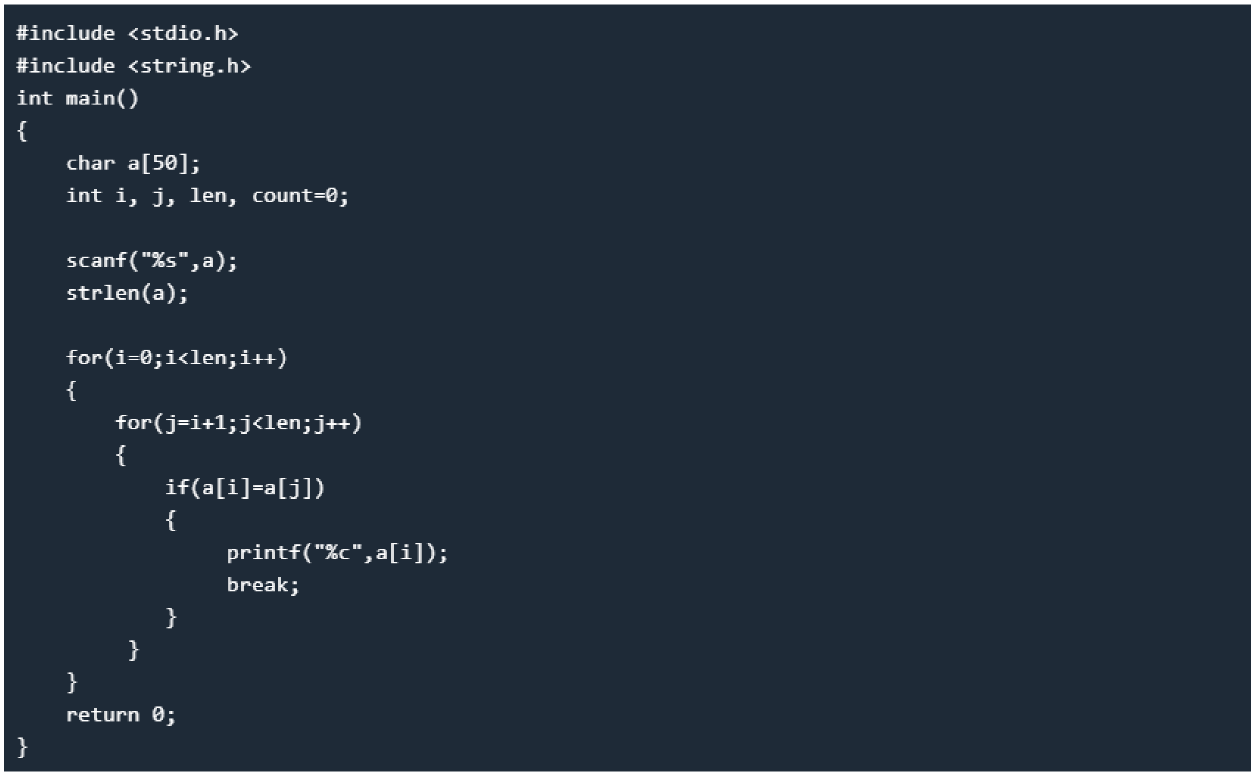
578378923

Output

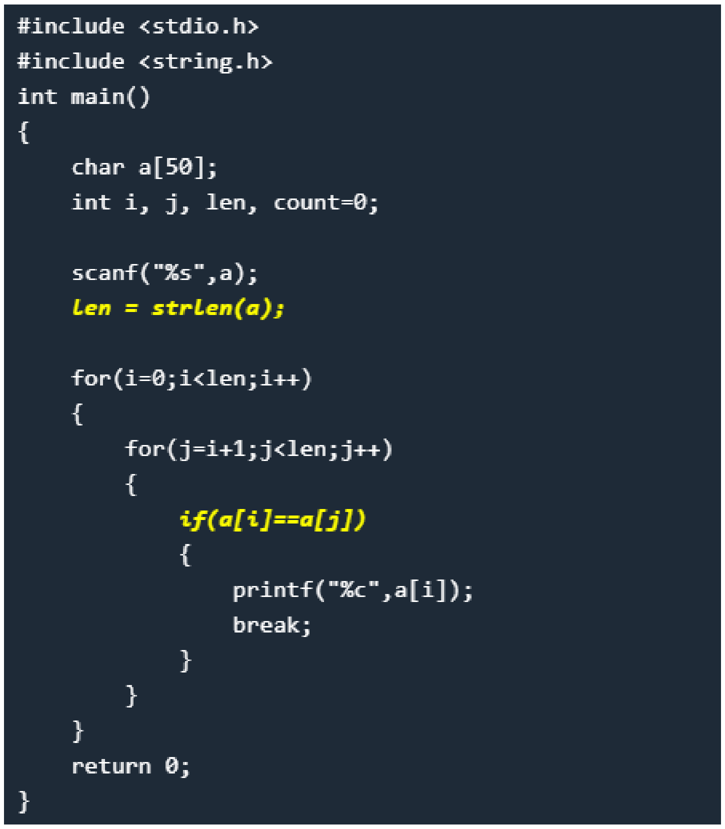
783

Explanation

The repeated digits in the data are 7, 8 and 3. So, the security key is 783



```c
#include <stdio.h>
#include <string.h>
int main()
{
    char a[50];
    int i, j, len, count=0;

    scanf("%s",a);
    strlen(a);

    for(i=0;i<len;i++)
    {
        for(j=i+1;j<len;j++)
        {
            if(a[i]=a[j])
            {
                 printf("%c",a[i]);
                 break;
            }
         }
    }
    return 0;
}
```



```c
#include <stdio.h>
#include <string.h>
int main()
{
    char a[50];
    int i, j, len, count=0;

    scanf("%s",a);
    len = strlen(a);

    for(i=0;i<len;i++)
    {
        for(j=i+1;j<len;j++)
        {
            if(a[i]==a[j])
            {
                printf("%c",a[i]);
                break;
            }
        }
    }
    return 0;
}
```

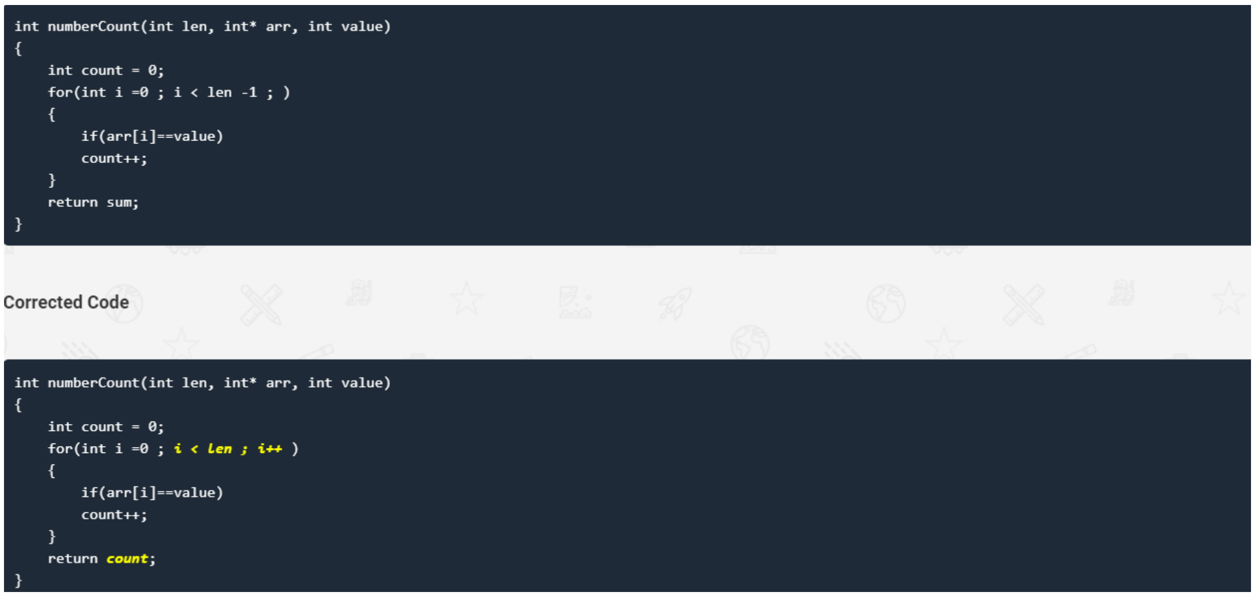
Question 15:

Function/method ***numberCount*** accepts three arguments-***len***, an integer representing the length of input list. ***arr,*** a list of integers. and ***value,*** an integer value. It returns an integer representing the sum of all the elements of ***arr*** that are equal to the given integer value for example

len = 9, arr = [1,2,4,3,2,5,4,1,2], value = 2

function /method will return 6

function/method compiles successfully but fails to return the desired result for some/all cases due to incorrect implementation. Your task is to debug the code so that it passes all test cases.



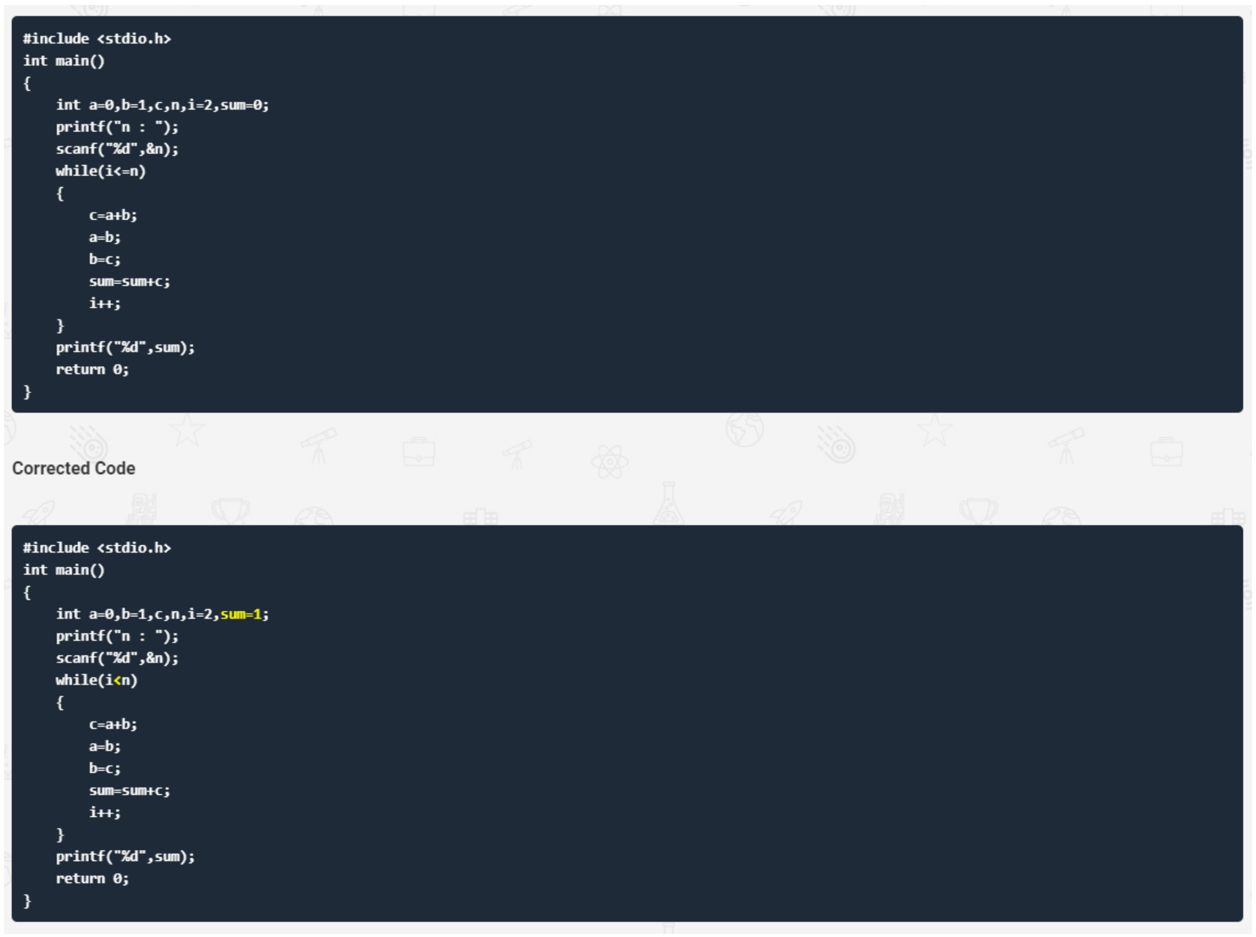
```
int numberCount(int len, int* arr, int value)
{
    int count = 0;
    for(int i =0 ; i < len -1 ; )
    {
        if(arr[i]==value)
        count++;
    }
    return sum;
}
```

Corrected Code

```
int numberCount(int len, int* arr, int value)
{
    int count = 0;
    for(int i =0 ; i < len ; i++ )
    {
        if(arr[i]==value)
        count++;
    }
    return count;
}
```

Question 16:

Print the sum of fibonacci series n th term.



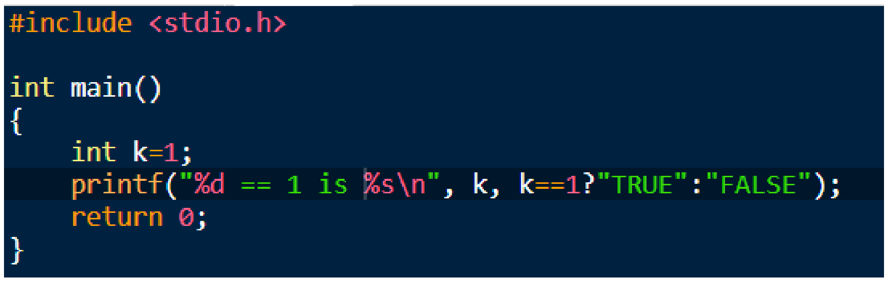
```
#include <stdio.h>
int main()
{
    int a=0,b=1,c,n,i=2,sum=0;
    printf("n : ");
    scanf("%d",&n);
    while(i<=n)
    {
        c=a+b;
        a=b;
        b=c;
        sum=sum+c;
        i++;
    }
    printf("%d",sum);
    return 0;
}
```

Corrected Code

```
#include <stdio.h>
int main()
{
    int a=0,b=1,c,n,i=2,sum=1;
    printf("n : ");
    scanf("%d",&n);
    while(i<n)
    {
        c=a+b;
        a=b;
        b=c;
        sum=sum+c;
        i++;
    }
    printf("%d",sum);
    return 0;
}
```

Question 17:

What will be the output of the program ?



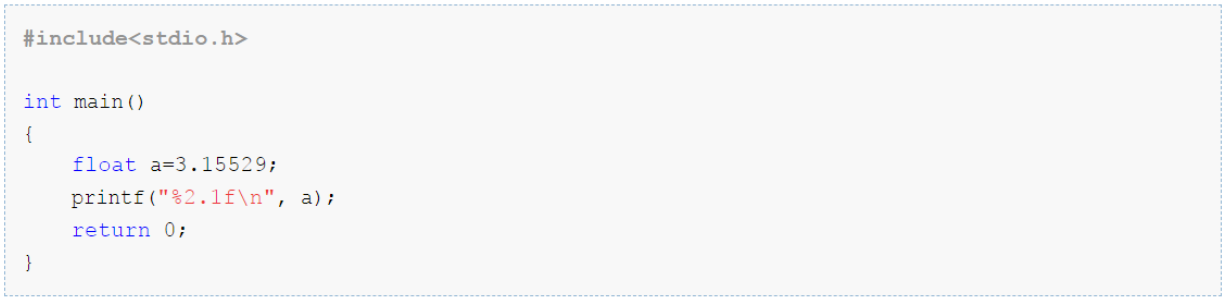
```
#include <stdio.h>

int main()
{
    int k=1;
    printf("%d == 1 is %s\n", k, k==1?"TRUE":"FALSE");
    return 0;
}
```

=> `printf("%d == 1 is" "%s\n", k, k==1?"TRUE":"FALSE");` becomes
=> `k==1?"TRUE":"FALSE"`
=> `1==1?"TRUE":"FALSE"`
=> `"TRUE"`

Therefore the output of the program is **1 == 1 is TRUE**

Question 18:

What will be the output of the program ?



```
#include<stdio.h>

int main()
{
    float a=3.15529;
    printf("%2.1f\n", a);
    return 0;
}
```

A. 3.00

B. 3.15

C. 3.2

D. 3

Answer: Option **C**

Explanation:

`float a=3.15529;` The variable `a` is declared as an `float` data type and initialized to value 3.15529;

`printf("%2.1f\n", a);` The precision specifier tells `.1f` tells the printf function to place only one number after the `.`(dot).

Hence the output is 3.2

Question 19:

What will be the output of the program?

```
#include<stdio.h>
int main()
{
    int i=-3, j=2, k=0, m;
    m = ++i && ++j && ++k;
    printf("%d, %d, %d, %d\n", i, j, k, m);
    return 0;
}
```

A. -2, 3, 1, 1

B. 2, 3, 1, 2

C. 1, 2, 3, 1

D. 3, 3, 1, 2

Step 1: `int i=-3, j=2, k=0, m;` here variable `i, j, k, m` are declared as an integer type and variable `i, j, k` are initialized to -3, 2, 0 respectively.

Step 2: `m = ++i && ++j && ++k;`

becomes `m = -2 && 3 && 1;`

becomes `m = TRUE && TRUE;` Hence this statement becomes TRUE. So it returns '1'(one). Hence m=1.

Step 3: `printf("%d, %d, %d, %d\n", i, j, k, m);` In the previous step the value of i,j,k are increemented by '1'(one).

Hence the output is "-2, 3, 1, 1".

Question 20:

```
#include<stdio.h>
int main()
{
    int x=12, y=7, z;
    z = x!=4 || y == 2;
    printf("z=%d\n", z);
    return 0;
}
```

A. z=0

B. z=1

C. z=4

D. z=2

Answer: Option **B**

Explanation:

Step 1: `int x=12, y=7, z;` here variable `x, y` and `z` are declared as an integer and variable `x` and `y` are initialized to 12, 7 respectively.

Step 2: `z = x!=4 || y == 2;`
becomes `z = 12!=4 || 7 == 2;`
then `z = (condition true) || (condition false);` Hence it returns 1. So the value of `z=1`.

Step 3: `printf("z=%d\n", z);` Hence the output of the program is "z=1".

Question 21:

What will be the output of the program?

```
#include<stdio.h>
int main()
{
    static int a[20];
    int i = 0;
    a[i] = i  ;
    printf("%d, %d, %d\n", a[0], a[1], i);
    return 0;
}
```

A. 1, 0, 1

B. 1, 1, 1

C. 0, 0, 0

D. 0, 1, 0

Answer: Option **C**

Explanation:

Step 1: `static int a[20];` here variable `a` is declared as an integer type and `static`. If a variable is declared as `static` and it will be automatically initialized to value '0'(zero).

Step 2: `int i = 0;` here vaiable `i` is declared as an integer type and initialized to '0'(zero).
Step 3: `a[i] = i ;` becomes `a[0] = 0;`
Step 4: `printf("%d, %d, %d\n", a[0], a[1], i);`
Here a[0] = 0, a[1] = 0(because all staic variables are initialized to '0') and i = 0.
Step 4: Hence the output is "0, 0, 0".

Question 22:

What will be the output of the program?

```
#include<stdio.h>
int main()
{
    int i=4, j=-1, k=0, w, x, y, z;
    w = i || j || k;
    x = i && j && k;
    y = i || j &&k;
    z = i && j || k;
    printf("%d, %d, %d, %d\n", w, x, y, z);
    return 0;
}
```

A. 1, 1, 1, 1

B. 1, 1, 0, 1

C. 1, 0, 0, 1

D. 1, 0, 1, 1

Explanation:

Step 1: `int i=4, j=-1, k=0, w, x, y, z;` here variable `i, j, k, w, x, y, z` are declared as an integer type and the variable `i, j, k` are initialized to 4, -1, 0 respectively.

Step 2: `w = i || j || k;` becomes `w = 4 || -1 || 0;`. Hence it returns TRUE. So, w=1

Step 3: `x = i && j && k;` becomes `x = 4 && -1 && 0;` Hence it returns FALSE. So, x=0

Step 4: `y = i || j &&k;` becomes `y = 4 || -1 && 0;` Hence it returns TRUE. So, y=1

Step 5: `z = i && j || k;` becomes `z = 4 && -1 || 0;` Hence it returns TRUE. So, z=1.

Step 6: `printf("%d, %d, %d, %d\n", w, x, y, z);` Hence the output is "1, 0, 1, 1".

Question 23:

```c
#include<stdio.h>
int main()
{
    int x=4, y, z;
    y = --x;
    z = x--;
    printf("%d, %d, %d\n", x, y, z);
    return 0;
}
```

A. 4, 3, 3

B. 4, 3, 2

C. 3, 3, 2

D. 2, 3, 3

Answer: Option **D**

Explanation:

Step 1: `int x=4, y, z;` here variable `x`, `y`, `z` are declared as an integer type and variable x is initialized to 4.

Step 2: `y = --x;` becomes `y = 3;` because (--x) is pre-decrement operator.

Step 3: `z = x--;` becomes `z = 3;`. In the next step variable `x` becomes 2, because (x--) is post-decrement operator.

Step 4: `printf("%d, %d, %d\n", x, y, z);` Hence it prints "2, 3, 3".

Question 24:

```c
#include<stdio.h>
int main()
{
    int i=3;
    i = i++;
    printf("%d\n", i);
    return 0;
}
```

A. 3

B. 4

C. 5

D. 6

Question 25:

```c
#include<stdio.h>
int main()
{
    int a=100, b=200, c;
    c = (a == 100 || b > 200);
    printf("c=%d\n", c);
    return 0;
}
```

A. c=100

B. c=200

C. c=1

D. c=300

Answer: Option **C**

Explanation:

Step 1: `int a=100, b=200, c;`
Step 2: `c = (a == 100 || b > 200);`
becomes `c = (100 == 100 || 200 > 200);`
becomes `c = (TRUE || FALSE);`
becomes `c = (TRUE);`(ie. c = 1)
Step 3: `printf("c=%d\n", c);` It prints the value of variable `i=1`
Hence the output of the program is '1'(one).

Question 26:

```
#include<stdio.h>
int main()
{
    int x=55;
    printf("%d, %d, %d\n", x<=55, x=40, x>=10);
    return 0;
}
```

A. 1, 40, 1

B. 1, 55, 1

C. 1, 55, 0

D. 1, 1, 1

Answer: Option **A**

Explanation:

Step 1: `int x=55;` here variable `x` is declared as an integer type and initialized to '55'.

Step 2: `printf("%d, %d, %d\n", x<=55, x=40, x>=10);`

In printf the execution of expressions is from Right to Left.

here `x>=10` returns TRUE hence it prints '1'.

`x=40` here `x` is assigned to 40 Hence it prints '40'.

`x<=55` returns TRUE. hence it prints '1'.

Step 3: Hence the output is "1, 40, 1".

Question 27:

```
#include<stdio.h>
int main()
{
    int k, num=30;
    k = (num>5 ? (num <=10 ? 100 : 200): 500);
    printf("%d\n", num);
    return 0;
}
```

A. 200

B. 30

C. 100

D. 500

Answer: Option **B**

Explanation:

Step 1: `int k, num=30;` here variable `k` and `num` are declared as an integer type and variable `num` is initialized to '30'.

Step 2: `k = (num>5 ? (num <=10 ? 100 : 200): 500);` This statement does not affect the output of the program. Because we are going to print the variable `num` in the next statement. So, we skip this statement.

Step 3: `printf("%d\n", num);` It prints the value of variable `num` '30'

Step 3: Hence the output of the program is '30'

Question 28:

```
#include<stdio.h>
int main()
{
    int i=2;
    int j = i + (1, 2, 3, 4, 5);
    printf("%d\n", j);
    return 0;
}
```

A. 4

B. 7

C. 6

D. 5

Answer: Option **B**

Explanation:

Because, comma operator used in the expression `i (1, 2, 3, 4, 5)`. The comma operator has left-right associativity. The left operand is always evaluated first, and the result of evaluation is discarded before the right operand is evaluated. In this expression 5 is the right most operand, hence after evaluating expression (1, 2, 3, 4, 5) the result is 5, which on adding to i results into 7.

Question 29:

Expected output:
choice is 2

```
#include
void main()
{
   float x = 2.2;
   switch (x)
   {
       case 2: printf("Choice is 2");
                 break;
       default: printf("Invalid choice");
                break;
   }
}
```

Answer:

```
#include
void main()
{
   int x = 2;  //rectifying syntax error
   switch (x)
   {
       case 2: printf("Choice is 2");
                 break;
       default: printf("Invalid choice");
                break;
   }
}
```

Question 30:

Programming Pattern

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

```
#include <stdio.h>
int main() {
   int i, space, rows, k = 0;
   printf("Enter the number of rows: ");
   scanf("%d", &rows);
   for (i = 1; i <= rows; ++i, k = 0) {
      for (space = 1; space <= rows - i; ++space) {
         printf("  ");
      }
      while (k != 2 * i - 1) {
         printf("* ");
         ++k;
      }
      printf("\n");
   }
   return 0;
}
```