

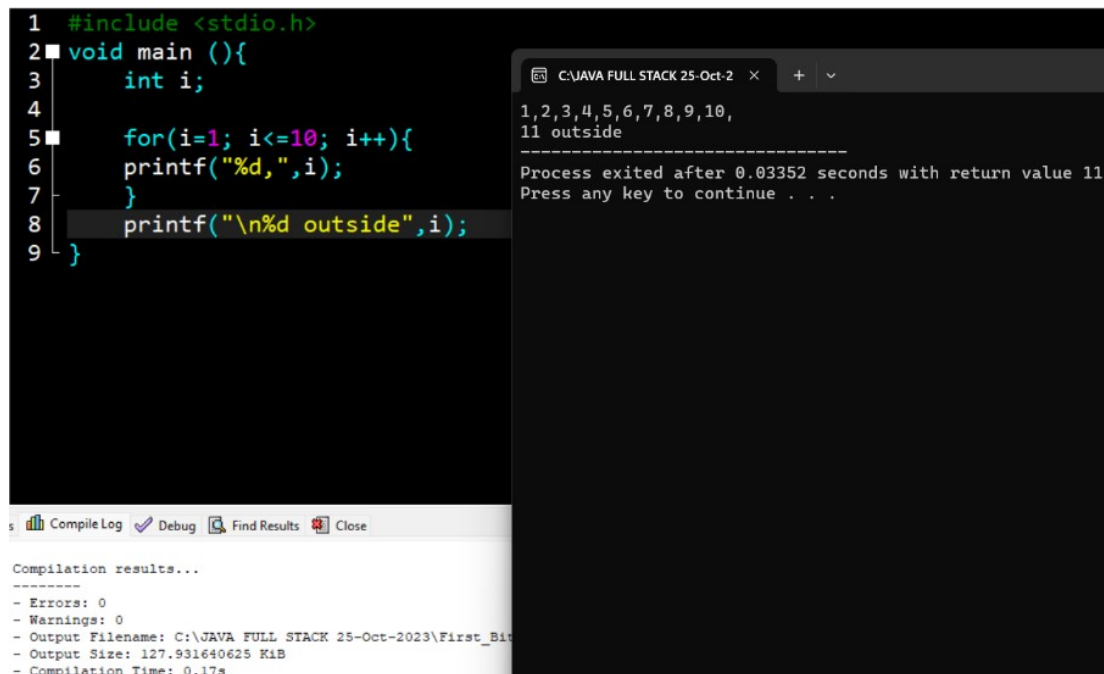
## Assignment 3

### 1. Print numbers from 1 to 10.

#### For Loop Code :

```
#include <stdio.h>
void main (){
    int i;

    for(i=1; i<=7; i++){
        printf("%d,",i);
    }
    printf("\n%d outside",i);
}
```



The screenshot shows a C program being compiled and executed. The code is as follows:

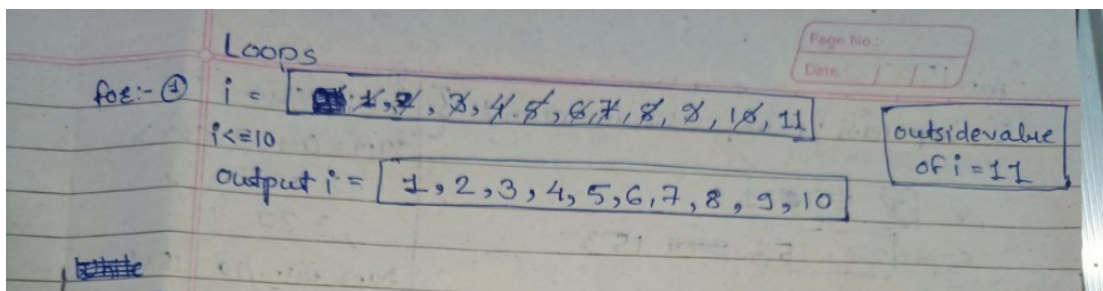
```
1 #include <stdio.h>
2 void main (){
3     int i;
4
5     for(i=1; i<=10; i++){
6         printf("%d,",i);
7     }
8     printf("\n%d outside",i);
9 }
```

The output of the program is:

```
1,2,3,4,5,6,7,8,9,10,
11 outside
-----
Process exited after 0.03352 seconds with return value 11
Press any key to continue . . .
```

Below the code editor, the compilation results are shown:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit
- Output Size: 127.931640625 KiB
- Compilation Time: 0.17s
```



#### While Loop code :

```
#include <stdio.h>
void main (){
    int i=1;
    while (i<=10) {
        printf("\n%d",i);
    }
```

```

        i++;
    }
}

```

The screenshot shows a code editor with a C program that uses a while loop to print numbers 1 through 10. The terminal output shows the command to compile and run the program, followed by the execution of the program which prints the numbers 1 to 10 on separate lines.

```

Assignments > 03_Assignment_Loop > While_Loop > 01_1TO10.c
1  #include <stdio.h>
2  void main (){
3      int i=1;
4      while (i<=10) {
5          printf("\n%d",i);
6          i++;
7      }
8  }
9
> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 01_1TO10.c -o 01_1TO10 } ; if ($?) { .\01_1TO10 }
1
2
3
4
5
6
7
8
9
10
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

## 2. Print table for the given number.

### For Loop Code :

```

#include <stdio.h>
void main (){
    int i,n,a;

    printf("Enter any number = ");
    scanf("%d",&n); //2

    for(i=1; i<=10; i++){
        //      printf("\n%d inside",i);
        a=i*n;

        printf("\n%d * %d = %d",i,n,a);
    }
    printf("\n%d outside",i);
}

```

```

1 #include <stdio.h>
2 void main (){
3     int i,n,a;
4
5     printf("Enter any number = ");
6     scanf("%d",&n); //2
7
8     for(i=1; i<=10; i++){
9         // printf("\n%d inside",i);
10        a=i*n;
11
12        printf("\n%d * %d = %d",i,n,a);
13    }
14    printf("\n%d outside",i);
15 }

```

Enter any number = 2

```

1 * 2 = 2
2 * 2 = 4
3 * 2 = 6
4 * 2 = 8
5 * 2 = 10
6 * 2 = 12
7 * 2 = 14
8 * 2 = 16
9 * 2 = 18
10 * 2 = 20
11 outside
-----
Process exited after 1.626 seconds with return value 11
Press any key to continue . . .

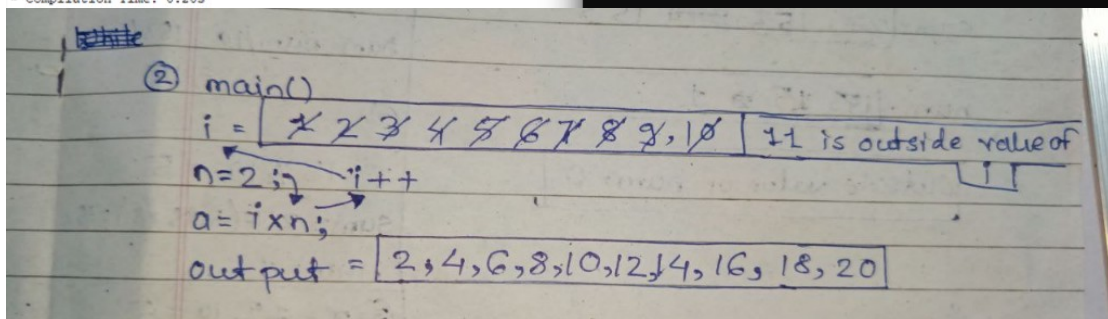
```

Compilation results...

```

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming
- Output Size: 128.1015625 KiB
- Compilation Time: 0.20s

```



## While Loop code :

```

#include <stdio.h>
void main (){
    int i=1,n,a;
    printf("Enter any number = ");
    scanf("%d",&n);
    while(i<=10){
        a=i*n;
        printf("\n%d * %d = %d",i,n,a);
        i++;
    }
}

```

```

Assignments > 03_Assignment_Loop > While_Loop > C 02_TableOf.c
1 #include <stdio.h>
2 void main (){
3     int i=1,n,a;
4     printf("Enter any number = ");
5     scanf("%d",&n);
6     while(i<=10){
7         a=i*n;
8         printf("\n%d * %d = %d",i,n,a);
9         i++;
10    }
11 }
12

```

```

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 02_TableOf.c -o 02_TableOf } ; if ($?) { .\02_TableOf }
Enter any number = 2

1 * 2 = 2
2 * 2 = 4
3 * 2 = 6
4 * 2 = 8
5 * 2 = 10
6 * 2 = 12
7 * 2 = 14
8 * 2 = 16
9 * 2 = 18
10 * 2 = 20
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

### 3. Calculate sum of numbers in the given range.

#### For Loop Code :

// Calculate sum of numbers in the given range.

#include <stdio.h>

void main (){

int i,sum=0;

for(i=1; i<=10; i++){

// printf("\n\n1)%d i value inside",i);

sum=sum+i;

// printf("\n\n2)%d sum value inside ",sum);

// printf("\n\n3)%d i value after sum inside",i);

}

printf("\n\n%d outside loop",sum);

printf("\n%d i value outside",i);

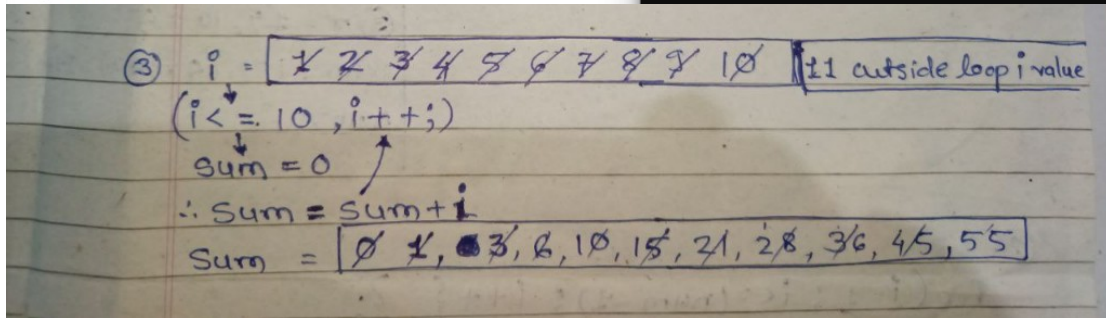
}

The screenshot shows a C program in a code editor and its execution output in a terminal window. The code is a for loop that calculates the sum of numbers from 1 to 10. The output shows the sum of 55 and the value of i as 11 after the loop.

```
1 // Calculate sum of numbers in the given range.
2 #include <stdio.h>
3 void main (){
4     int i,sum=0;
5     for(i=1; i<=10; i++){
6         // printf("\n\n1)%d i value inside",i);
7         sum=sum+i;
8         // printf("\n\n2)%d sum value inside ",sum);
9         // printf("\n\n3)%d i value after sum inside",i);
10    }
11    printf("\n\n%d outside loop",sum);
12    printf("\n%d i value outside",i);
13 }
14 }
```

Output:

```
55 outside loop
11 i value outside
Process exited after 0.02855 seconds with return value 19
Press any key to continue . . .
```



#### While Loop code :

// Calculate sum of numbers in the given range.

#include <stdio.h>

void main (){

int i=1,sum=0;

while(i<=10){

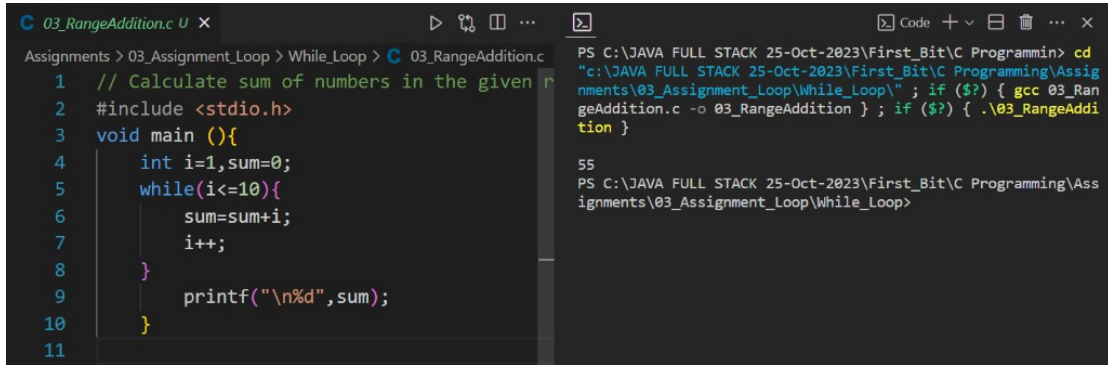
sum=sum+i;

i++;

}

```
printf("\n%d",sum);
```

```
}
```



```
03_RangeAddition.c U x
Assignments > 03_Assignment_Loop > While_Loop > 03_RangeAddition.c
1 // Calculate sum of numbers in the given range
2 #include <stdio.h>
3 void main () {
4     int i=1, sum=0;
5     while(i<=10){
6         sum=sum+i;
7         i++;
8     }
9     printf("\n%d", sum);
10 }
11

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programmin> cd
"c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 03_RangeAddition.c -o 03_RangeAddition } ; if ($?) { .\03_RangeAddition }
55
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>
```

#### 4. Check number is prime or not.

##### For Loop Code :

```
#include <stdio.h>
int main() {
    int n, i, count = 0;
    printf("Enter number to check PRIME or COMPOSITE = ");
    scanf("%d",&n); //6

    for(i=2; i<=n/2; i++)
    {
        // check for non prime number
        if(n%i==0)
        {
            count=1;
            break;
        }
    }
    if(n==1)printf("\n1 is nither PRIME nor COMPOSITE");
    else if (count==0) printf("\n%d is a PRIME number",n);
    else printf("\n%d is COMPOSITE number",n);
}
```

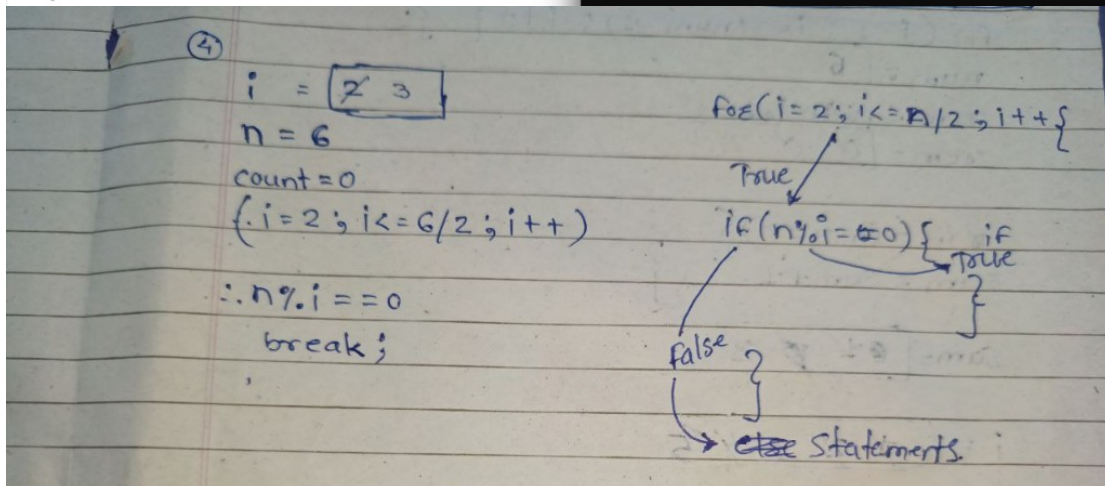
```

1 #include <stdio.h>
2 int main() {
3     int n, i, count = 0;
4     printf("Enter number to check PRIME or COMPOSITE = ");
5     scanf("%d",&n); //6
6
7     for(i=2; i<=n/2; i++)
8     {
9         // check for non prime number
10        if(n%i==0)
11        {
12            count=1;
13            break;
14        }
15    }
16    if(n==1)printf("\n1 is nither PRIME
17    else if (count==0) printf("\n%d is a

```

C:\JAVA FULL STACK 25-Oct-2 x + v  
 Enter number to check PRIME or COMPOSITE = 6  
 6 is COMPOSITE number  
 -----  
 Process exited after 3.96 seconds with return value 22  
 Press any key to continue . . .

Compile Log Debug Find Results Close  
 Compilation results...  
 - Errors: 0  
 - Warnings: 0  
 - Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Programmin  
 - Output Size: 128.6015625 KiB  
 - Compilation Time: 0.19s



## While Loop code :

//Check number is prime or not

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

```
void main (){
```

```
    int i=2, n, a=0;
```

```
    printf("Enter any number = ");
```

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

```
    while(i<=n/2)
```

```
{
```

```
    if (n%i==0){
```

```
        a=1;
```

```
        break;
```

```
}
```

```
    i++;
```

```
}
```

```
if(n==1) printf("1 is neither prime nor composite");
```

```
else if(a==0) printf("the given number is prime");
```

```
else printf("the given number is composite");
```



```
}

Assignments > 03_Assignment_Loop > While_Loop > C 04_PrimeNumber.c
1 //Check number is prime or not
2 #include <stdio.h>
3 void main (){
4     int i=2, n, a=0;
5     printf("Enter any number = ");
6     scanf("%d",&n);
7
8     while(i<=n/2)
9     {
10         if (n%i==0){
11             a=1;
12             break;
13         }
14         i++;
15     }
16     if(n==1) printf("1 is neither prime nor composite");
17     else if(a==0) printf("the given number is prime");
18     else printf("the given number is composite");
19 }
20
```

```
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 04_PrimeNumber.c -o 04_PrimeNumber } ; if ($?) { .\04_PrimeNumber }
Enter any number = 6
the given number is composite
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>
```

## 5. Check number is armstrong or not?

### For Loop Code :

```
#include <stdio.h>
void main(){
    int num,r,sum=0,temp;
    printf("Input a number = ");
    scanf("%d",&num);
//    num = 153;
    for(temp=num; num!=0; num=num/10){
        r=num % 10;
//        printf("\n1)%d r value",r);
        sum=sum+(r*r*r);
//        printf("\n2)%d sum value",sum);
//        printf("\n3)%d num value",num);
    }
//    printf("\n\n4)%d num outside value",num);

    if(sum==temp)
        printf("\n\n%d is an Armstrong number",temp);
    else
        printf("\n\n%d is not an Armstrong number",temp);
}
```

```

1  #include <stdio.h>
2  void main(){
3      int num,r,sum=0,temp;
4      printf("Input a number = ");
5      scanf("%d",&num);
6      // num = 153;
7      for(temp=num; num!=0; num=num/10){
8          r=num % 10;
9          printf("\n1)%d r value",r);
10         // sum=sum+(r*r*r);
11         printf("\n2)%d sum value",sum);
12         // printf("\n3)%d num value",num);
13     }
14     // printf("\n\n4)%d num outside value",num);
15
16     if(sum==temp)
17         printf("\n\n%d is an Armstrong number",sum);

```

Compilation results...

```

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignment1
- Output Size: 128.6015625 KiB
- Compilation Time: 0.20s

```

C:\JAVA FULL STACK 25-Oct-2 x + v

Input a number = 153

153 is an Armstrong number

-----

Process exited after 8.47 seconds with return value 28

Press any key to continue . . .

Handwritten notes illustrating the Armstrong number check for 153:

num = 153

⑤ temp = num; num != 0; num = num / 10 {

r = 3, 5, 1

sum = 27, 153, 153

num = 153, 15, 1

Outside value of num = 0

① r = 153 % 10 = 3

sum = 0 + (3 \* 3 \* 3)

= 0 + 27

= 27

num = num / 10 = 153 / 10

= 15

② r = 15 % 10 = 5

sum = 27 + (5 \* 5 \* 5) = 152

num = 1

③ r = 1 % 10 = 1

sum = 152 + (1 \* 1 \* 1) = 153

num = 0

## While Loop code :

```

#include <stdio.h>
int main() {
    int num, r, sum = 0, temp;
    printf("Input a number: ");
    scanf("%d", &num);

    temp = num;

    while (num != 0) {
        r = num % 10;
        sum = sum + (r * r * r);
    }

```



```

    num = num / 10;
}
if (sum == temp) printf("%d is an Armstrong number.\n", temp);
else printf("%d is not an Armstrong number.\n", temp);
return 0;
}

```

The screenshot shows a code editor with a file named `05_armstrong.c`. The code is as follows:

```

1  #include <stdio.h>
2  int main() {
3      int num, r, sum = 0, temp;
4      printf("Input a number: ");
5      scanf("%d", &num);
6
7      temp = num;
8
9      while (num != 0) {
10         r = num % 10;
11         sum = sum + (r * r * r);
12         num = num / 10;
13     }
14     if (sum == temp) printf("%d is an Armstrong number.\n", temp);
15     else printf("%d is not an Armstrong number.\n", temp);
16     return 0;
17 }

```

The terminal output on the right shows the execution of the program:

```

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop" ; if ($?) { gcc 05_armstrong.c -o 05_armstrong } ; if ($?) { .\05_armstrong }
Input a number: 153
153 is an Armstrong number.
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

## 6. Check number is perfect or not.

### For Loop Code :

```

#include <stdio.h>
int main()
{
    int num, rem, sum = 0, i;

    // printf("Enter a Number: ");
    // scanf("%d", &num);
    num=6;
    for (i = 1; i <= (num - 1); i++)
    {
        rem = num % i;
        // printf("\n\n1)%d rem value ",rem);
        if (rem == 0){
            sum = sum + i;
        }
        // printf("\n\n2)%d sum value ",sum);
        // printf("\n\n3)%d i value ",i);
    }
    // printf("\n\n4)%d i outside value ",i);
    if (sum == num)
        printf("\n%d is perfect number", num);
    else
        printf("\n%d is NOT a perfect number", num);
    return 0;
}

```

```

1 #include <stdio.h>
2 int main()
3 {
4     int num, rem, sum = 0, i;
5
6     // printf("Enter a Number: ");
7     // scanf("%d", &num);
8     num=6;
9     for (i = 1; i <= (num - 1); i++)
10 {
11     rem = num % i;
12     // printf("\n\n1)%d rem value ",rem);
13     if (rem == 0){
14         sum = sum + i;
15         // printf("\n2)%d sum value ",sum);
16     }
17     // printf("\n3)%d i value ",i);

```

Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Programming\Assignments\03
- Output Size: 127,931,640,625 KiB
- Compilation Time: 0.20s

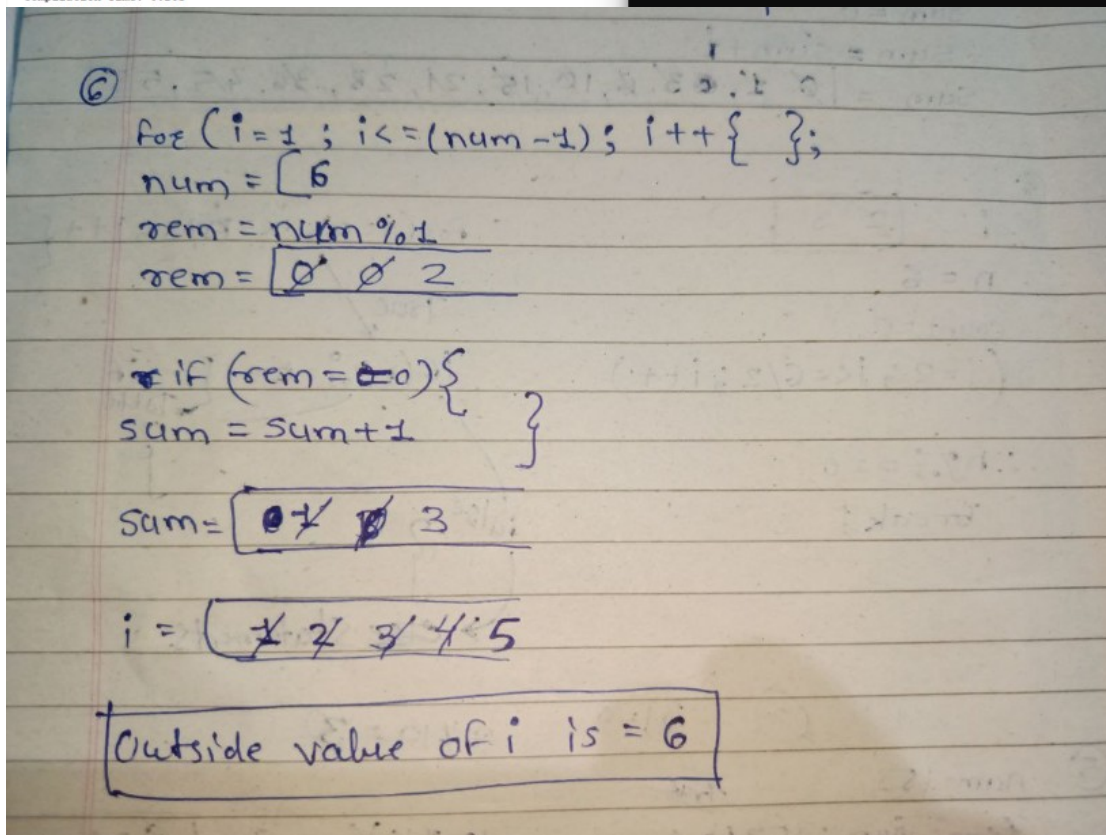
C:\JAVA FULL STACK 25-Oct-2 x + v

6 is perfect number

-----

Process exited after 0.03907 seconds with return value 0

Press any key to continue . . . |



## While Loop code :

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

```
int main(void)
```

```
{
```

```
    int number, i, sum = 0;
```

```
    printf("Enter the number: ");
```

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

```
    i = 1;
```

```
    while (i < number)
```

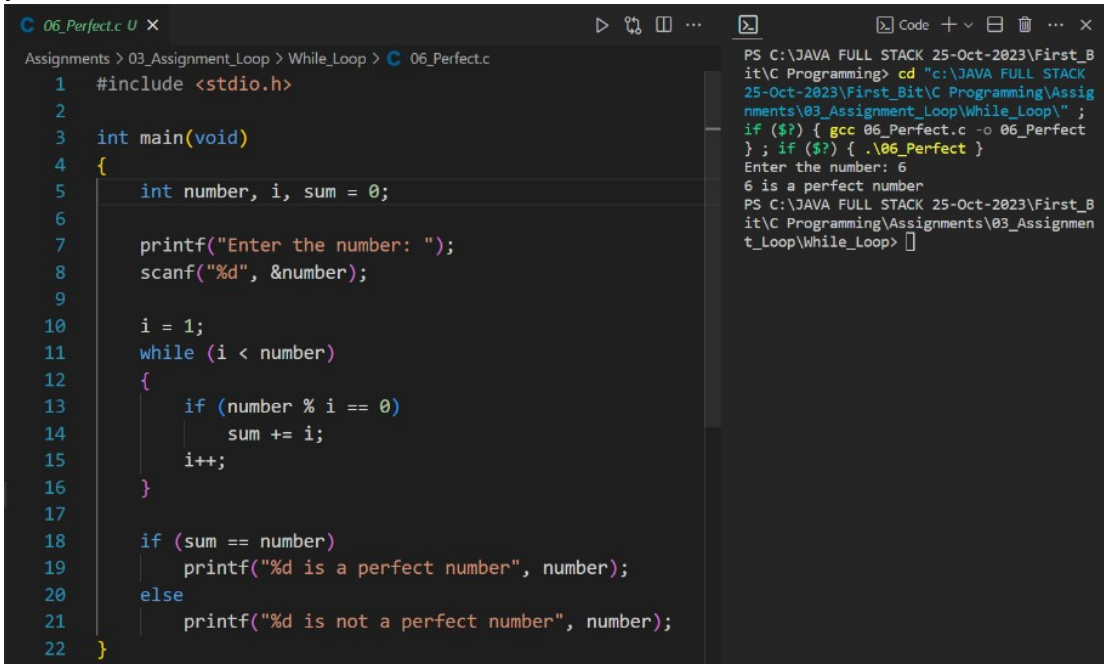
```
{
```

```

        if (number % i == 0)
            sum += i;
        i++;
    }

    if (sum == number)
        printf("%d is a perfect number", number);
    else
        printf("%d is not a perfect number", number);
}

```



```

06_Perfect.c U X
Assignments > 03_Assignment_Loop > While_Loop > 06_Perfect.c
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int number, i, sum = 0;
6
7      printf("Enter the number: ");
8      scanf("%d", &number);
9
10     i = 1;
11     while (i < number)
12     {
13         if (number % i == 0)
14             sum += i;
15         i++;
16     }
17
18     if (sum == number)
19         printf("%d is a perfect number", number);
20     else
21         printf("%d is not a perfect number", number);
22 }

```

```

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop" ; if ($?) { gcc 06_Perfect.c -o 06_Perfect } ; if ($?) { .\06_Perfect }
Enter the number: 6
6 is a perfect number
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

## 7. Find factorial of number.

### For Loop Code :

```

#include<stdio.h>
void main()
{
    int num,factorial, i;
    factorial =1;
    printf("Enter the number: ");
    scanf("%d",&num);

    for( i=1; i<=num; i++)
    {
        printf("\n\n0)value factorialof  %d",factorial);

        factorial=factorial*i;

        printf("\n1)value of i %d",i);
        printf("\n2)value of factorial after operation %d",factorial);
    }
    printf("\nout side value of i %d",i);
}

```

```
printf("\nThe number is factorial number %d",factorial);
```

```

1 #include<stdio.h>
2 void main()
3 {
4     int num,factorial, i;
5     factorial =1;
6     printf("Enter the number: ");
7     scanf("%d",&num);
8
9     for( i=1; i<=num; i++)
10    {
11        printf("\n\n0)value factorialof  %d",fa
12
13        factorial=factorial*i;
14
15        printf("\n1)value of i %d",i);
16        printf("\n2)value of factorial after op
17    }

```

Compilation results...

```

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\As
- Output Size: 128.6015625 KiB
- Compilation Time: 0.17s

```

0)value factorialof 1  
1)value of i 1  
2)value of factorial after operation 1

0)value factorialof 1  
1)value of i 2  
2)value of factorial after operation 2

0)value factorialof 2  
1)value of i 3  
2)value of factorial after operation 6

0)value factorialof 6  
1)value of i 4  
2)value of factorial after operation 24

0)value factorialof 24  
1)value of i 5  
2)value of factorial after operation 120

0)value factorialof 120  
1)value of i 6  
2)value of factorial after operation 720  
out side value of i 7  
The number is factorial number 720

Process exited after 1.524 seconds with return value 35  
Press any key to continue . . .

## While Loop code :

```

#include<stdio.h>
void main()
{
    int num,factorial, i=1;
    factorial =1;
    printf("Enter the number: ");
    scanf("%d",&num);

    while( i<=num)
    {
        factorial=factorial*i;
        i++;
    }
    printf("The number is factorial number %d",factorial);
}

```

```

C 07_Factorial.c U X
Assignments > 03_Assignment_Loop > While_Loop > C 07_Factorial.c
1 #include<stdio.h>
2 void main()
3 {
4     int num,factorial, i=1;
5     factorial =1;
6     printf("Enter the number: ");
7     scanf("%d",&num);
8
9     while( i<=num)
10    {
11        factorial=factorial*i;
12        i++;
13    }
14    printf("The number is factorial number %d",factorial);
15 }
16

```

PS C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Programming\Assignments\03\_Assignment\_Loop\While\_Loop\" ; if (\$?) { gcc 07\_Factorial.c -o 07\_Factorial } ; if (\$?) { .\07\_Factorial }  
Enter the number: 6  
The number is factorial number 720  
PS C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Programming\Assignments\03\_Assignment\_Loop\While\_Loop>

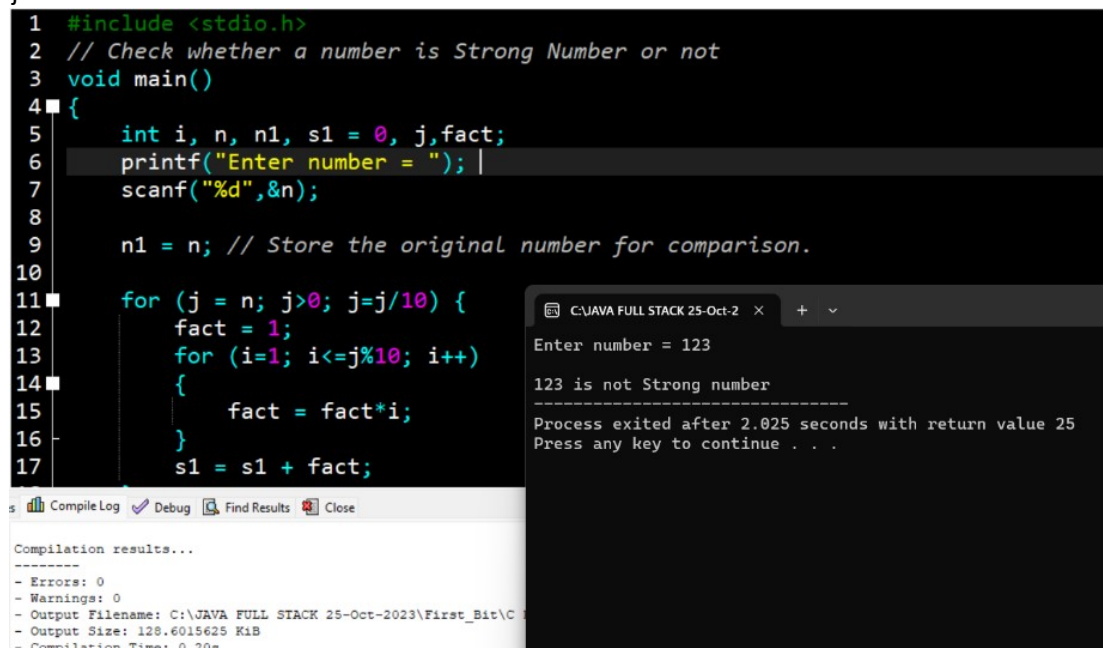
## 8. Check number is strong or not.

### For Loop Code :

```
#include <stdio.h>
// Check whether a number is Strong Number or not
void main()
{
    int i, n, n1, s1 = 0, j, fact;
    printf("Enter number = ");
    scanf("%d", &n);

    n1 = n; // Store the original number for comparison.

    for (j = n; j > 0; j = j/10) {
        fact = 1;
        for (i = 1; i <= j%10; i++)
        {
            fact = fact*i;
        }
        s1 = s1 + fact;
    }
    if (s1 == n1) {
        printf("\n%d is Strong number", n1);
    }
    else {
        printf("\n%d is not Strong number", n1);
    }
}
```



```
1 #include <stdio.h>
2 // Check whether a number is Strong Number or not
3 void main()
4 {
5     int i, n, n1, s1 = 0, j, fact;
6     printf("Enter number = ");
7     scanf("%d", &n);
8
9     n1 = n; // Store the original number for comparison.
10
11     for (j = n; j > 0; j = j/10) {
12         fact = 1;
13         for (i = 1; i <= j%10; i++)
14         {
15             fact = fact*i;
16         }
17         s1 = s1 + fact;
18     }
19     if (s1 == n1) {
20         printf("\n%d is Strong number", n1);
21     }
22     else {
23         printf("\n%d is not Strong number", n1);
24     }
25 }
```

Enter number = 123

123 is not Strong number

-----

Process exited after 2.025 seconds with return value 25

Press any key to continue . . .

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C
- Output Size: 128.6015625 KiB
- Compilation Time: 0.20s

### While Loop code :

```
#include <stdio.h>
// Check whether a number is Strong Number or not
void main()
```



```

{
    int i, n, n1, s1 = 0, j, fact;
    printf("Enter number = ");
    scanf("%d", &n);

    n1 = n; // Store the original number for comparison.
    j = n;
    while (j>0) {
        fact = 1;
        for (i=1; i<=j%10; i++)
        {
            fact = fact*i;
        }
        s1 = s1 + fact;
        j=j/10;
    }

    if (s1 == n1) {
        printf("\n%d is Strong number", n1);
    }
    else{
        printf("\n%d is not Strong number", n1);
    }
}

```

```

Assignments > 03_Assignment_Loop > While_Loop > C 08_Strong.c
7     scanf("%d", &n);
8
9     n1 = n; // Store the original number for comparison.
10    j = n;
11    while (j>0) {
12        fact = 1;
13        for (i=1; i<=j%10; i++)
14        {
15            fact = fact*i;
16        }
17        s1 = s1 + fact;
18        j=j/10;
19    }
20
21    if (s1 == n1) {
22        printf("\n%d is Strong number", n1);
23    }
24    else{
25        printf("\n%d is not Strong number", n1);
26    }
27 }

```

```

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 08_Strong.c -o 08_Strong } ; if ($?) { .\08_Strong }
Enter number = 145

145 is Strong number
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

## 9. Check the given number is palindrome or not?

### For Loop Code :

```

#include <stdio.h>

int main(){
    int num, r, sum = 0, t;

    printf("Input a number: ");

```

```

scanf("%d", &num);

// reversed integer is stored in reversed variable
for(t = num; num != 0; num = num / 10){
    r = num % 10;
    sum = sum * 10 + r;
}

if(t == sum)
    printf("%d is a palindrome number.\n", t);
else
    printf("%d is not a palindrome number.\n", t);
    return 0;
}

```

The screenshot shows a C++ IDE with the following code:

```

3  int num, r, sum = 0, t;
4
5  printf("Input a number: ");
6  scanf("%d", &num);
7
8  // reversed integer is stored in reversed variable
9  for(t = num; num != 0; num = num / 10){
10     r = num % 10;
11     sum = sum * 10 + r;
12 }
13
14 if(t == sum)
15     printf("%d is a palindrome number.\n", t);
16 else
17     printf("%d is not a palindrome number.\n", t);
18
19 }

```

The output window shows the following text:

```

C:\JAVA FULL STACK 25-Oct-2  x  +  v
Input a number: 1991
1991 is a palindrome number.

-----
Process exited after 4.071 seconds with return value 0
Press any key to continue . . .

```

The compilation results window shows the following text:

```

Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment
- Output Size: 128.6015625 KiB
- Compilation Time: 0.20s

```

## While Loop code :

```

#include <stdio.h>

int main() {
    int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;

    // reversed integer is stored in reversed variable
    while (n != 0) {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
        n /= 10;
    }

    // palindrome if original and reversed are equal
    if (original == reversed)
        printf("%d is a palindrome.", original);
    else
        printf("%d is not a palindrome.", original);
}

```

```

return 0;
}

```

The screenshot shows a C program in a text editor and its execution in a PowerShell terminal. The program, named 09\_Palindrome.c, uses a while loop to reverse a number and then compares it to the original to determine if it is a palindrome. The terminal output shows the user entering 1456, which is correctly identified as not a palindrome.

```

Assignments > 03_Assignment_Loop > While_Loop > 09_Palindrome.c
1  #include <stdio.h>
2  int main() {
3      int n, reversed = 0, remainder, original;
4      printf("Enter an integer: ");
5      scanf("%d", &n);
6      original = n;
7
8      // reversed integer is stored in reversed variable
9      while (n != 0) {
10         remainder = n % 10;
11         reversed = reversed * 10 + remainder;
12         n /= 10;
13     }
14
15     // palindrome if original and reversed are equal
16     if (original == reversed)
17         printf("%d is a palindrome.", original);
18     else
19         printf("%d is not a palindrome.", original);
20
21     return 0;
22 }

```

```

PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming> cd "c:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop\" ; if ($?) { gcc 09_Palindrome.c -o 09_Palindrome } ; if ($?) { .\09_Palindrome }
Enter an integer: 1456
1456 is not a palindrome.
PS C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming\Assignments\03_Assignment_Loop\While_Loop>

```

## 10.Add the (first and last) digit of a given number?

### For Loop Code :

```

#include <stdio.h>
int main()
{
    int n, first_digit,last_digit;
    printf("Enter the Number = ");
    scanf("%d", &n);
    last_digit = n % 10; //last digit of a number

    for(n; n > 0; n = n / 10)
    {
        first_digit = n % 10; //first digit of a number
    }
    printf("The first digit of the number is : %d",first_digit);
    printf("\nThe last digit of the number id : %d",last_digit);
    return 0;
}

```

The screenshot shows a C program in Visual Studio Code. The code is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int n, first_digit, last_digit;
5     printf("Enter the Number = ");
6     scanf("%d", &n);
7     last_digit = n % 10; //last digit of a number
8
9     for(n; n > 0; n = n / 10)
10    {
11        first_digit = n % 10; //first digit of a number
12    }
13
14    printf("The first digit of the number is : %d", first_digit);
15    printf("\nThe last digit of the number id : %d", last_digit);
16    return 0;
17 }
```

The output window shows the following text:

```
Enter the Number = 145
The first digit of the number is : 1
The last digit of the number id : 5
-----
Process exited after 1.583 seconds with return value 0
Press any key to continue . . .
```

The compilation results window shows the following text:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First_Bit\C Programming
- Output Size: 128.6015625 K1B
- Compilation Time: 0.20s
```

## While Loop code :

```
#include <stdio.h>
int main()
{
    int n, first_digit, last_digit;
    printf("Enter the Number = ");
    scanf("%d", &n);
    last_digit = n % 10; //last digit of a number

    while (n > 0)
    {
        first_digit = n % 10; //first digit of a number
        n = n / 10;
    }
    printf("The first digit of the number is : %d", first_digit);
    printf("\nThe last digit of the number id : %d", last_digit);
    return 0;
}
```

```
1 #include <stdio.h>
2 int main()
3 {
4     int n, first_digit, last_digit;
5     printf("Enter the Number = ");
6     scanf("%d", &n);
7     last_digit = n % 10; //last digit of a number
8
9     while (n > 0)
10    {
11        first_digit = n % 10; //first digit of a number
12        n = n / 10;
13    }
14    printf("The first digit of the number is : %d", first_digit);
15    printf("\nThe last digit of the number is : %d", last_digit);
16    return 0;
17 }
```

Compile Log | Debug | Find Results | Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Program\first\_and\_last\_digit.c
- Output Size: 128.6015625 KiB
- Compilation Time: 0.20s

C:\JAVA FULL STACK 25-Oct-2023\First\_Bit\C Program\first\_and\_last\_digit.c

Enter the Number = 156  
The first digit of the number is : 1  
The last digit of the number is : 6  
-----  
Process exited after 2.068 seconds with return value 0  
Press any key to continue . . .

=====

**END**