C_35 ⇒ Properties of For Loop

Syntax :-

```
for(expression 1; expression 2; expression 3)
{
  // Body of loop
}
```

Properties of expression 1 :-

optional.

```
for((i=1), i<=5; i++)
{
}
```

(Initialization skipped in for loop since i value already initialized.)

```
#include <stdio.h>
#include <conio.h>
Void main()
{
 int i=1;
 for( ;i<=5;i++)
 {
  printf("%d\n",i);
 }
 getch();
}
```

⇒

O/P

```
1
2
3
4
5
```

---

② void main() (i value not initialized by default i value initialized to zero)

```
{
 int i; → Not Initialized.
 for( ;i<=5;i++)
 {
  printf("%d\n",i);
 }
 getch();
}
```

O/P

```
0
1
2
3
4
5
```

---

③ More than one variable can be initialized between only commas.

```
void main()
{
 int i,j;
 // int i=1,j=0;
 // for( ;i<=5;i++)
 for(i=1,j=0;i<=5;i++)
 {
  printf("%d %d\n",i,j);
 }
}
```

O/P

```
10
20
30
40
50
```

Continued in another notebook - 2

Properties of Expression 2 :-

(4)
```
Void main()
{
  int i,j;
  for(i=1,j=0;_;i++)
  {
    printf("%d %d\n",i,j);
  }
}
```

No condition so loop runs infinite no. of times

* So to terminate the loop we have to put Condition.

(5)
```
Void main()
{
  int i,j;
  for(i=1,j=0;i<=5,j<3;i++)
  {
    printf("%d %d\n",i,j);
  }
}
```

* We can put more than one Conditions but last Condition is treated as termination Condition

o/p

```
1 0
2 0
:
:
:
infinite loop.
```

→ * Here no Conditions for termination for i value; so we have infinite loop.

Properties of expression 3:

⑥ We can put more than one update or modify expression.

O/P

10
21
32

```
Void main()
{
  int i,j;
  for(i=1, j=0; i<=5, j<3; i++, j++)
  {
    printf("%d %d\n", i, j);
  }
}
```

* It also checks only last condition; but in expression 3, all the number of updates will be executed.

⑦ Void main() ⇒ Loop terminates without printing nothing because at start itself condition fails.

O/P

No output

```
{
  int i,j;
  for(i=1, j=0; i==10; i++, j++)
  {
    printf("%d %d", i, j);
  }
```

Property of Expression 2:

* We can also write any logical condition also ...

O/P

10
22
33
44
55

⑧ any one false → loop terminates.

```
for(i=1, j=0; i<=5 && j<=5; i++, j++)
  printf("%d %d\n", i, j);
```

(9) `for(i=1,j=0; i<=5 || j<=5; i++, j++)`

```
{
  printf("%d %d \n", i, j);
}
```

O/P

```
1 1
2 2
3 3
4 4
5 5
```

(10) `for(i=1,j=1; i<=10 || j<=5; i++,j++)`

```
{
  printf("%d %d \n", i, j);
}
```

O/P

```
1 1
2 2
3 3
:
10 10
```

(11) `for(i=1,j=1; i<=10 && j<=5; i++,j++)`

```
{
  printf("%d %d \n", i, j);
}
```

O/P

```
1 1
2 2
3 3
4 4
5 5
```

Properties of Expression 3

* Modify or update expression can also be put within the body of the loop.

(12)

```
for(i=1,j=0; i<=10, j<3;)
{
  printf("%d %d \n", i, j);
  i++;
  j++;
}
```

O/P:

```
1 0
2 1
3 2
```

* When there is no update or modify expression;

O/P:

01
01
01
.
.
.
infinite times

(13)

```
for(i=0,j=1; i<=10,j<3; )
{
  printf("%d %d\n",i,j);
}
```

* Not necessary to always modify/update to go inside loop.

* Without modification, the loop also gets executed until it finds the condition suitable to terminate.

* Hence, here there is no modification so loop will be executed infinite times with initial values of i and j

(14)

```
void main()
{
 for(i=0,j=1; i<=10, j==0; )
 {
  printf("%d %d\n",i,j);
 }
}
```

O/P

No output

* Here at the start itself condition fails $j=1$ ($j==0$)
* So it won't enter into loop so we wont get output.

(15) for (i=1, j=0; i<=10, j<3;)

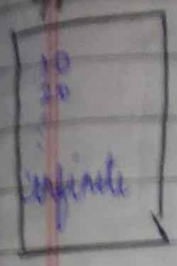
{

printf("%d %d\n", i, j);

i++;

}

or



* Here also we get infinite loop because there is no condition suitable to terminate j value.

* But i value incremented with same value of j=0 and goes infinite times.

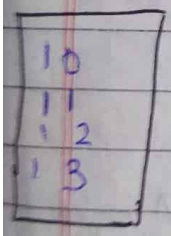
(16) for (i=0, j=0; i<=10, j<3;)

{

printf("%d %d\n", i, j);

j++;

}



* Here j values finds suitable terminate condition with modification in j. but i values no update

* Loop gets terminated with modifications to j value but loop always gets executed independent to modification of j value.

Properties of expression 1:-

```
int i, a=10, b=10;
(17) for(i=(a+b)/10; i<=5; i++)
     {
       printf("%d\n", i);
     }
```

O/P

```
2
3
4
5
```

* We can also use expressions during initialization.

Properties of expression 3:-

```
int i, a=10, b=10;
(18) for(i=(a+b)/10; i<=5; i=i+3)
                           i=i+a
     {
       printf("%d\n", i);
     }
```

O/P

```
2
3.
```

* In modify/update expression; we can write any updation rather than i++ or i--

* Any updation or expression like i=i+a ; i=i+3 (or) i+=3.

(19) Property [All expressions are optional]

```
for(;;)
{
  printf("Jenny\n");
}
```

O/P

```
Jenny
:
Infinite
```

(20) property (with only condition):

```
int i=1;
for( ; i<5; )
{
  printf("Jenny\n");
  i++;
}
```

o/p:
Jenny
Jenny
Jenny
Jenny

(20) // remove i++ ⇒ o/p: Infinity

* 4 times Jenny printed.

(21) Property for decreasing values.

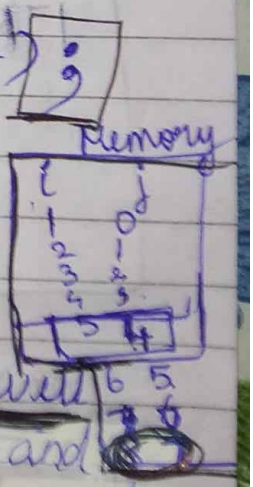
```
for(i=10; i>=0; i--)
{
  printf("%d\n", i);
}
```

o/p: 10 9 8 ... 1

(22) Property [(;) after for loop statement]

```
for(i=1, j=0; i<5, j<=6; i++, j++);
{
  printf("%d %d\n", i, j);
}
```

o/p: 5 6

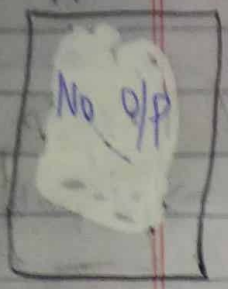


* Since it is semicolon, it will treated as program statement and updation is done in memory and once the loop terminates, it will only print the updated values of i and j.

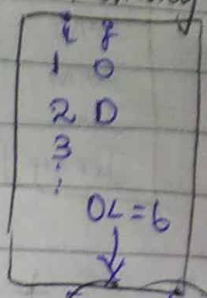
(23) `for(i=1,j=0; i<5, j<=6; i++);`

```
{
    printf("%d %d\n", i, j);
    j++;
}
```

O/P



Memory



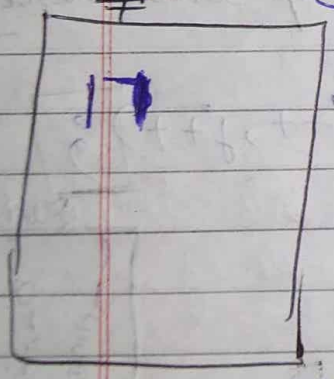
infinite times

* Here j value always; $0 <= 6$ and there is no termination

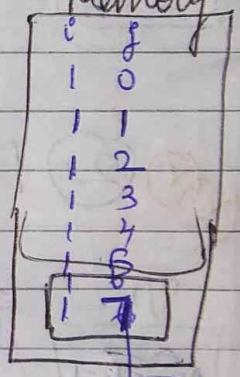
(24) `for(i=1,j=0; i<5, j<=6; j++);`

```
{
    printf("%d %d\n", i, j);
}
```

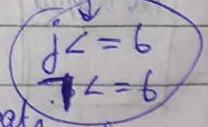
O/P:



Memory



loop terminate

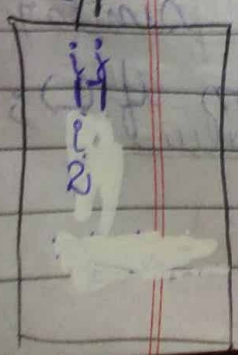


main()

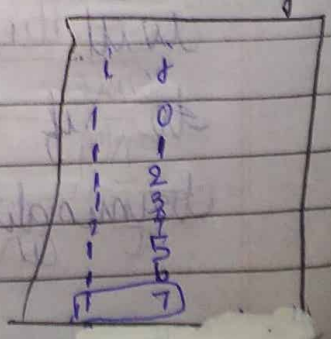
(25) `for(i=1,j=0; i<5, j<=6; j++);`

```
{
    printf("%d %d\n", i, j);
    i++;
}
printf("%d", i);
```

O/P



Memory



(26) for(i=1, j=0, k=3; i<=5, j<=6, k>1; i++, j++, k--);

{

printf("%d %d %d", i, j, k);

}

O/P

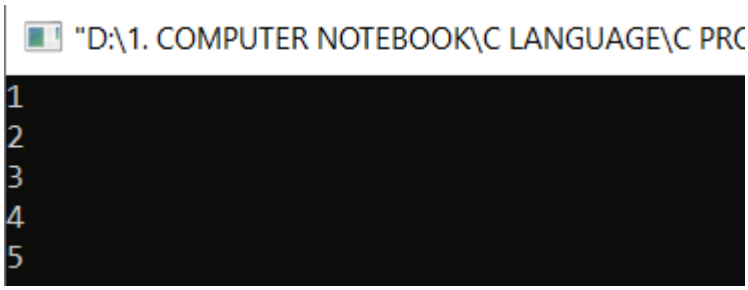
i	j	k
3	2	1

Memory

i	j	k
1	0	3
2	1	2
3	2	1

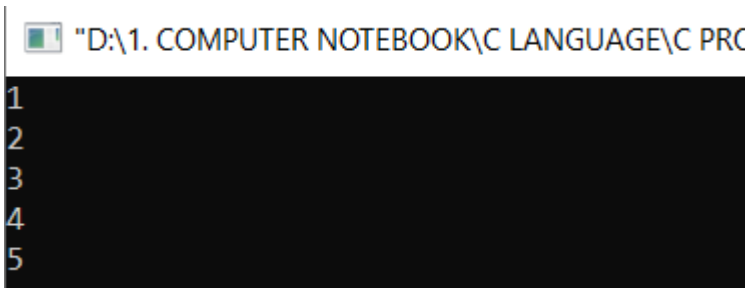
CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
/* 1 - for */
int main()
{
    //int i;
    for(int i=1;i<=5;i++) //Declaration & Initialization i=1
    {
     printf("%d\n",i);
    }
  getch();
}
```



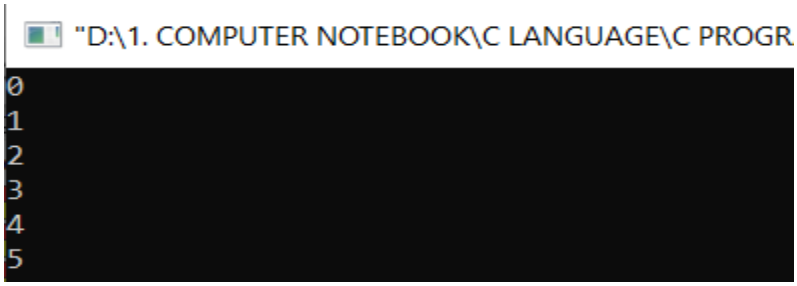
CODE 2:

```c
#include <stdio.h>
#include <stdlib.h>
/* 2 - for */
int main()
{
 int i=1; //Declaration & Initialization
 for(;i<=5;i++) //Initialization part skipped since already i value initialized
 {
  printf("%d\n",i);
  }
  getch();
}
```



CODE 3:

```c
#include <stdio.h>
#include <stdlib.h>
/* 3 - for */
int main()
{
 int i; // i is only Declared
 for(;i<=5;i++) // i value is not initialized
 {
  printf("%d\n",i);
  }
  getch();
}
/* In this case, by default i value is initialized to 0
   Value of i starts from 0 and ends till 5 */
```



CODE 4:

```c
#include <stdio.h>
#include <stdlib.h>
/* 4 for */
int main()
{
    int i,j;              // int i=1,j=0;
    for(i=1,j=0;i<=5;i++) //for(;i<=5;i++)
   // {
     printf("%d%d\n",i,j);
   // } for single statement no need for curly braces

    getch();
}
/* We can declare and use more than one variables inside for loop
   But variables should be separated by commas */
```

```c
#include <stdio.h>
#include <stdlib.h>
/* 4 for */
int main()
{
    /* int i,j; */                  int i=1,j=0;
    /* for(i=1,j=0;i<=5;i++) */     for(;i<=5;i++)
    // {
      printf("%d%d\n",i,j);
    // } for single statement no need for curly braces

     getch();
}
/* We can declare and use more than one variables inside for loop
   But variables should be separated by commas */
```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys
10
20
30
40
50
```

CODE 5:

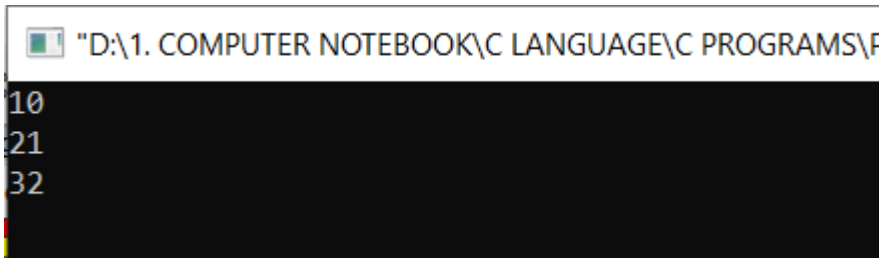
```c
#include <stdio.h>
#include <stdlib.h>
/* 5 - for */
int main()
{
int i,j;
   for(i=1,j=0; ;i++)
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
/* Here no condition in for loop, only i value is incremented, so i value
is incremented till infinite number of times and the loop is executed
infinite number of times */
```

CODE 6:

```c
#include <stdio.h>
#include <stdlib.h>
/* 6 - for */
int main()
{
int i,j;
   for(i=1,j=0;i<=5,j<3;i++)
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
/* We can give more than one condition but only last condition is taken
other are considered as statement...
  In this case, value of i is incremented and there is no condition to
  terminate i value so loop is executed infinite number of times */
```

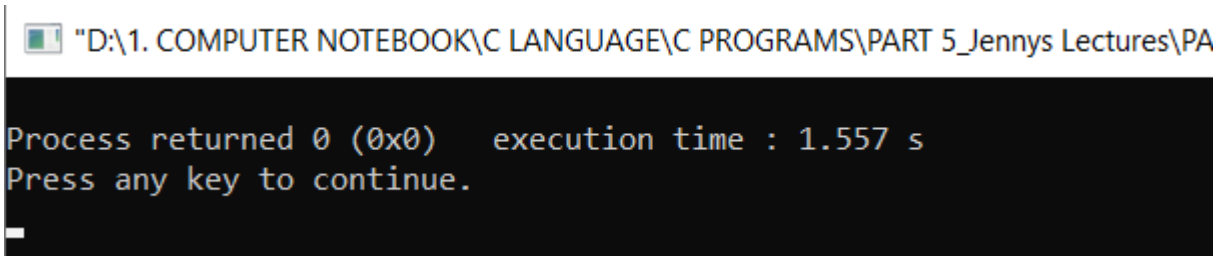
CODE 7:

```c
#include <stdio.h>
#include <stdlib.h>
/* 7 - for */
int main()
{
int i,j;
   for(i=1,j=0;i<=1,j<3;i++,j++) //i<=5 any condition for i, but it wont check
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
/* Here we can also have more than one update/modify expression
   and all those updates will be executed.... but only last condition
   is executed.... */
```



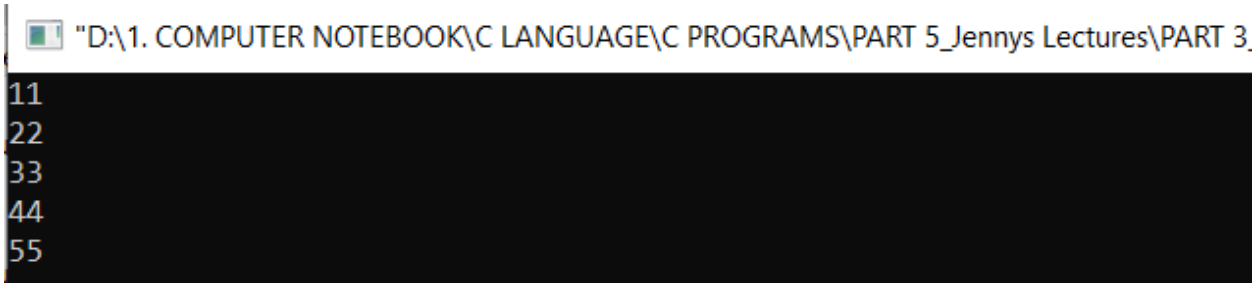
CODE 8:

```c
#include <stdio.h>
#include <stdlib.h>
/* 8 - for */
int main()
{
int i,j;
   for(i=1,j=0;i==10;i++,j++) //i==10 condition fails, so loop terminates
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
// No output
```



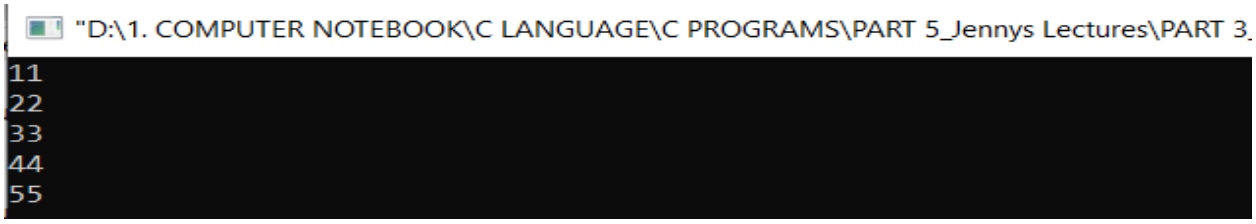
CODE 9:

```c
#include <stdio.h>
#include <stdlib.h>
/* 9 - for */
int main()
{
int i,j;
   for(i=1,j=1;i<=5,j<=5;i++,j++) //one condition with logical AND
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
```



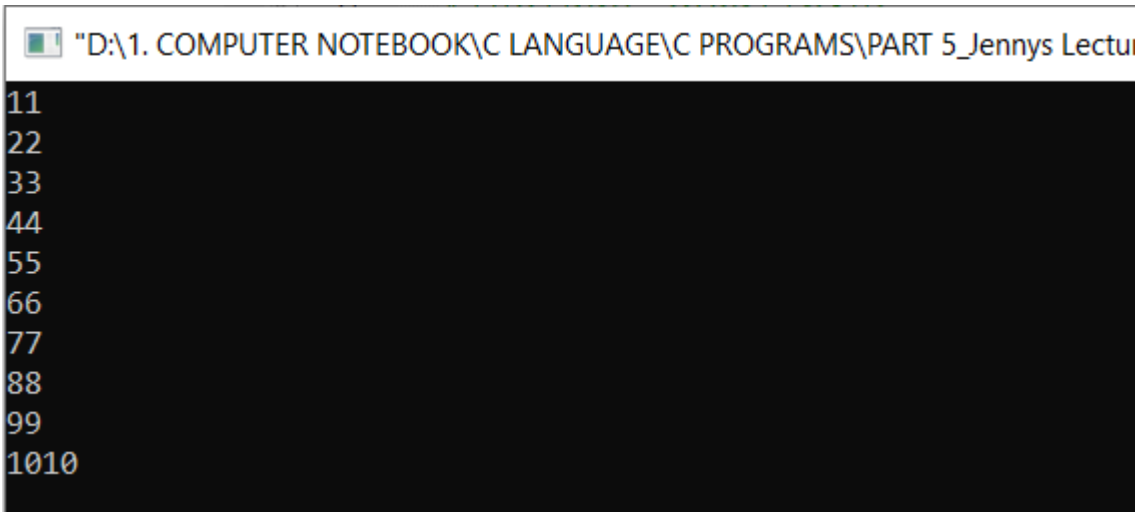
CODE 10:

```c
#include <stdio.h>
#include <stdlib.h>
/* 10 - for */
int main()
{
int i,j;
   for(i=1,j=1;i<=5 || j<=5;i++,j++) //one condition with logical OR
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
```



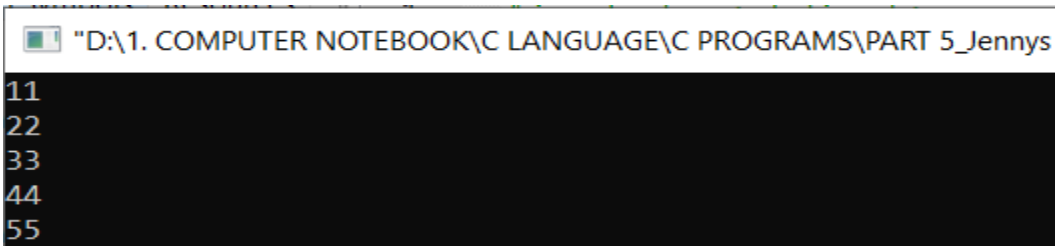
CODE 11:

```c
#include <stdio.h>
#include <stdlib.h>
/* 11 - for */
int main()
{
int i,j;
   for(i=1,j=1;i<=10 || j<=5;i++,j++) //one condition with logical OR i<=10
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
```



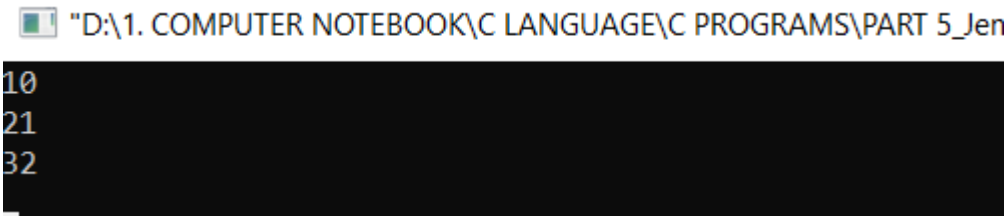
CODE 12:

```c
#include <stdio.h>
#include <stdlib.h>
/* 12 - for */
int main()
{
int i,j;
   for(i=1,j=1;i<=10 && j<=5;i++,j++) //one condition with logical AND i<=10
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
```



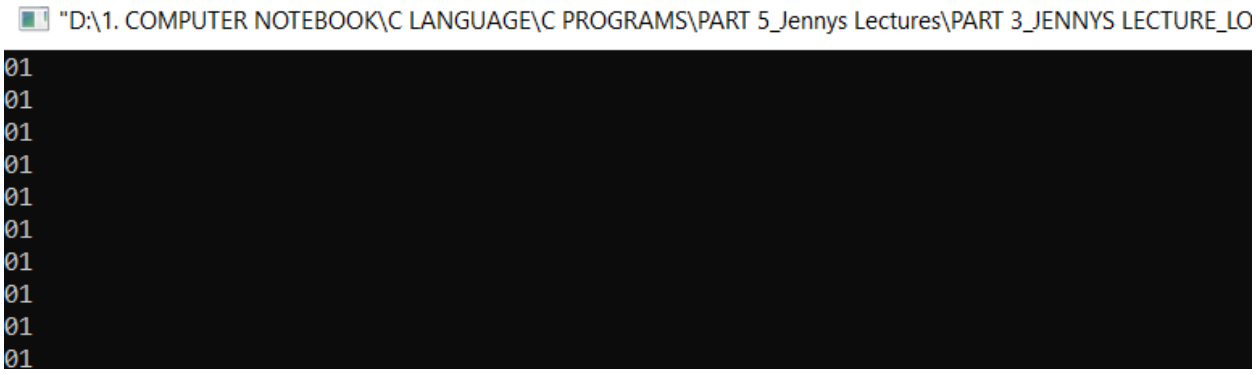
CODE 13:

```c
#include <stdio.h>
#include <stdlib.h>
/* 13 - for */
int main()
{
int i,j;
 for(i=1,j=0;i<=10,j<3; )//2 conditions but modify expression placed inside loop
   {
     printf("%d%d\n",i,j);
     i++; //modify expression inside loop
     j++; //modify expression inside loop
   }
    getch();
}
```



CODE 14:

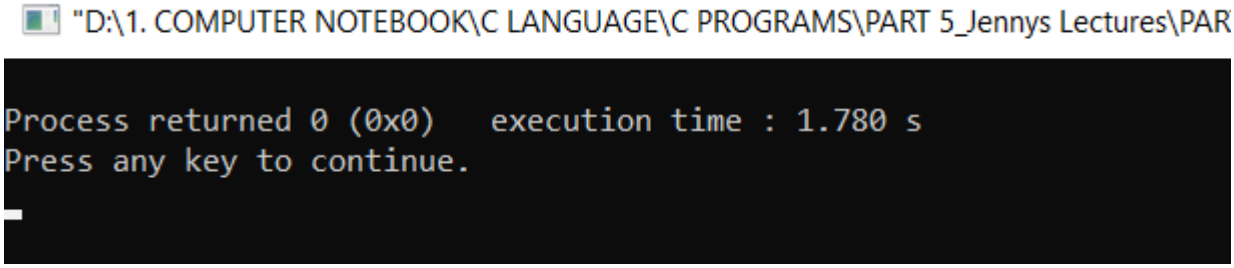
```c
#include <stdio.h>
#include <stdlib.h>
/* 14 - for */
int main()
{
int i,j;
 for(i=0,j=1;i<=10,j<3; )// without modify expression
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
/* But every time the value of j<3 that is 0<3 is executed..
   Loop will get executed independent of modify expression.....
   Loop will be dependent only on condition...
   Loop will be terminated once it finds terminate condition, till then executed
   Hence here, loop get executed infinite number of times.... */
```



CODE 15:

```c
#include <stdio.h>
#include <stdlib.h>
/* 15 - for */
int main()
{
int i,j;
 for(i=0,j=1;i<=10,j==0; )// without modify expression
   {
     printf("%d%d\n",i,j);
   }
    getch();
}
/* j==0 condition fails, since j value is 1
   Loop will get executed independent of modify expression.....
   Loop will be dependent only on condition...
   Loop will be terminated once it finds terminate condition, till then executed
   Hence here, at the start itself condition fails....
   So, It wont enter into the loop and no value is printed, NO OUTPUT*/

```



CODE 16:

```c
#include <stdio.h>
#include <stdlib.h>
/* 16 - for */
int main()
{
int i,j;
 for(i=1,j=0;i<=10,j<3; )// no modify for j
   {
     printf("%d%d\n",i,j);
     i++; // i value modified
   }
    getch();
}
// loop gets executed infinite times with increasing values of i & with j=0

```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 3_
52080
52090
52100
52110
```

CODE 17:

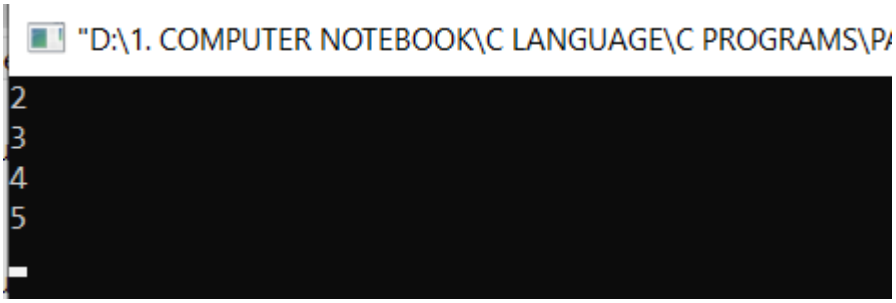
```c
#include <stdio.h>
#include <stdlib.h>
/* 17 - for */
int main()
{
int i,j;
 for(i=1,j=0;i<=10,j<=3; )
   {
     printf("%d%d\n",i,j);
     j++; // j value modified & hence loop terminated
   }
    getch();
}
/* Here loop gets terminated with modification to j value but loop always
gets executed independent to the modification to j value */

```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5
10
11
12
13
```

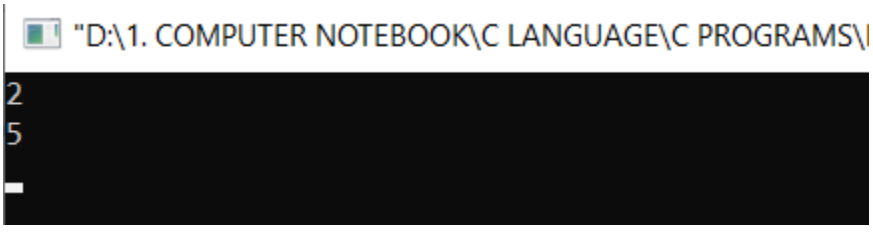
CODE 18:

```c
#include <stdio.h>
#include <stdlib.h>
/* 18 - for */
int main()
{
int i,a=10,b=10;
 for(i=(a+b)/10;i<=5;i++) // Initialization also have expressions
    {
      printf("%d\n",i);
    }
     getch();
}
```



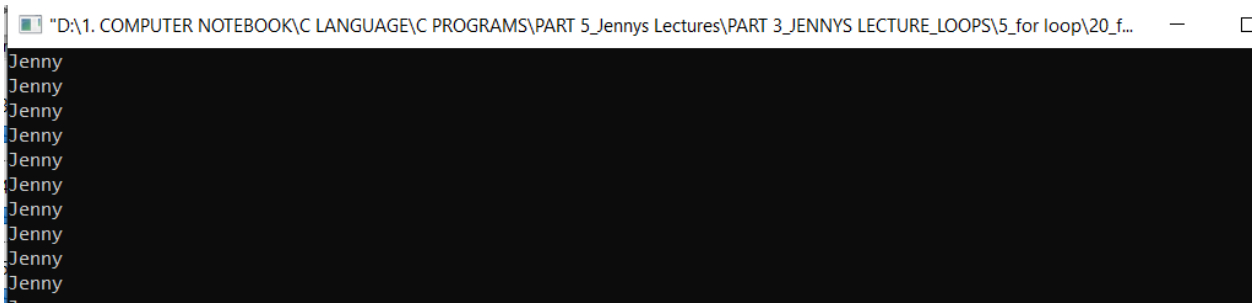
CODE 19:

```c
#include <stdio.h>
#include <stdlib.h>
/* 19 - for */
int main()
{
int i,a=10,b=10;
 for(i=(a+b)/10;i<=5;i+=3) // Modify/update can also have expressions i=i+3
    {
      printf("%d\n",i);
    }
     getch();
}
```



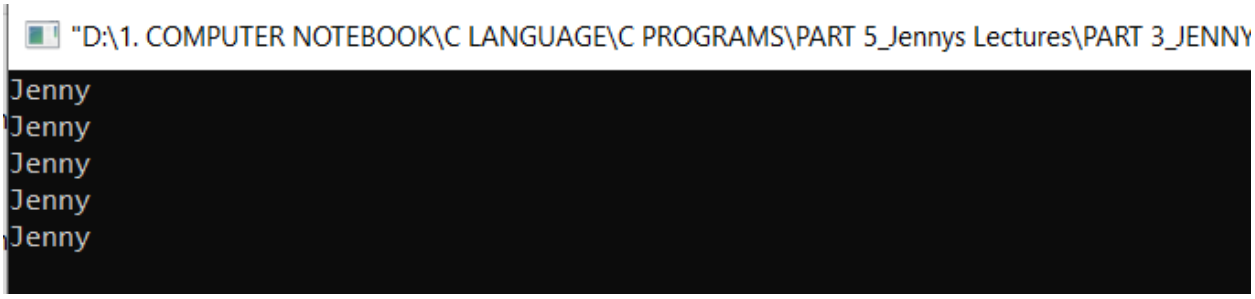
CODE 20:

```c
#include <stdio.h>
#include <stdlib.h>
/* 20 - for */
int main()
{
    for(;;)//all expressions are optional
    {
     printf("Jenny\n");
    }
    getch();
}
/* Here no initialization condition or modification but only body of loop..
   so the loop will be executed infinite number of times...
   It will terminated only if it finds the suitable condition of termination.. */
```



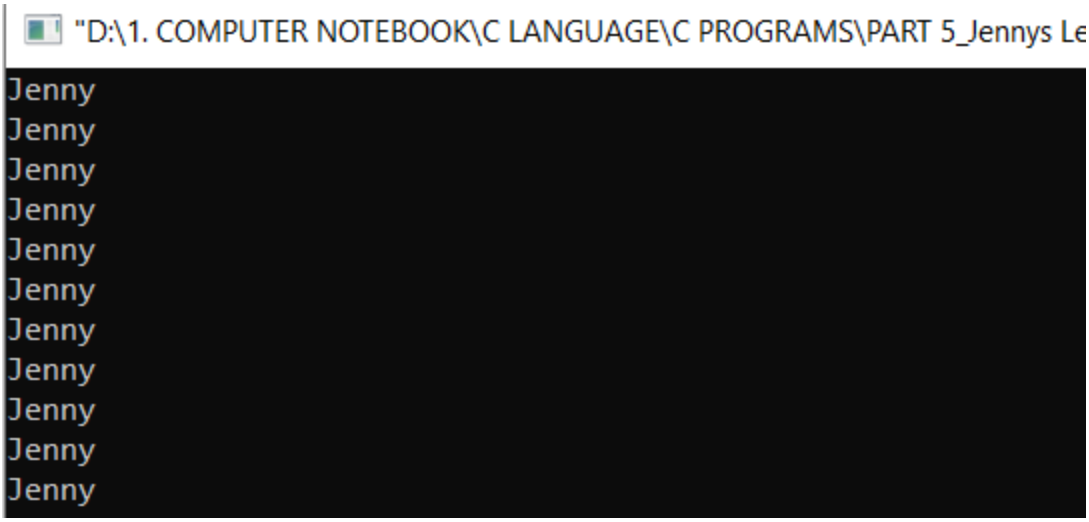
CODE 21:

```c
#include <stdio.h>
#include <stdlib.h>
/* 21 - for */
int main()
{
    int i=1; // Here declaration of i must be necessary
    for(;i<=5;)//initialization and modiy are only optional
    {
     printf("Jenny\n");
     i++; //When i modified loop termites
    }
    getch();
}
/* with condition the loop gets executed 5 times and after 5th time it gets
   terminated */
```



CODE 22:

```c
#include <stdio.h>
#include <stdlib.h>
/* 22 - for */
int main()
{
    int i=1; // Here declaration of i must be necessary
    for(;i<=5;)//initialization and modiy are only optional
    {
     printf("Jenny\n");
     //i++; //When i modified loop termites
    }
    getch();
}
/* No modification to i, so every time i=1 and i<=5 executed infinite number
  of times and body of loop be executed infinite times.... */
```



CODE 23:

```c
#include <stdio.h>
#include <stdlib.h>
/* 23 - for */
int main()
{
    int i;
    for(i=10;i>=1;i--) //decreasing values of i
    {
     printf("%d\n",i);
    }
    getch();
}
```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 3_JEN
10
9
8
7
6
5
4
3
2
1
```

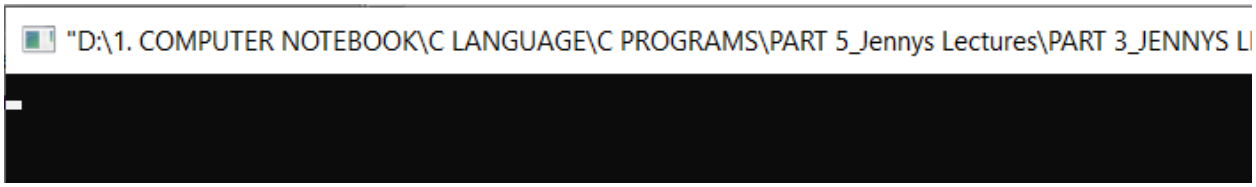
CODE 24:

```c
#include <stdio.h>
#include <stdlib.h>
/* 24 - for */
//for loop statement with semicolon
int main()
{
    int i,j;
    for(i=1,j=0;i<5,j<=6;i++,j++);
    // semicolon, it will be considered as statement
    {
     printf("%d%d\n",i,j);
    }
    getch();
}
/* semicolon, it will be considered as statement and it will executed till
 the condition fails and after that only it goes inside the body of loop and
 prints the values for executed condition values......*/
```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C I
87
```

CODE 25:

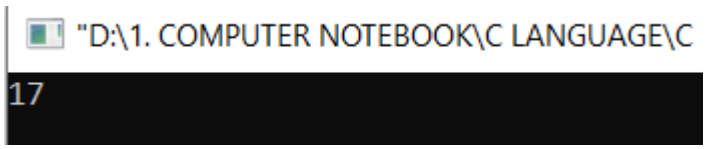
```c
#include <stdio.h>
#include <stdlib.h>
/* 25 - for */
//for loop statement with semicolon
int main()
{
    int i,j;
    for(i=1,j=0;i<5,j<=6;i++);
    // semicolon, it will be considered as statement
    {
     printf("%d%d\n",i,j);
     j++; //j modified inside loop
    }
    getch();
}
/* Condition for j do not terminate but only i value incremented and
   we wont get enter into loop so we wont get output */
```



CODE 26:

```c
#include <stdio.h>
#include <stdlib.h>
/* 26 - for */
//for loop statement with semicolon
int main()
{
    int i,j;
    for(i=1,j=0;i<5,j<=6;j++);
    // semicolon, it will be considered as statement
    {
     printf("%d%d\n",i,j);

    }
    getch();
}
/* Here we have termination for j value since we have modification in j value
   hence loop gets executed after the for statement with semicolon.... */
```



CODE 27:

```c
#include <stdio.h>
#include <stdlib.h>
/* 27 - for */
//for loop statement with semicolon
int main()
{
    int i,j;
    for(i=1,j=0;i<5,j<=6;j++);
    // semicolon, it will be considered as statement
    {
     printf("i=%d j=%d\n",i,j);
      i++;
/* Even if i value is modified but no condition for i, So it wont enter into
the body of loop... But i value will be changed in memory.... */
    }
    printf("i=%d\n",i);
    getch();
}
```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS
i=1 j=7
i=2
```

CODE 28:

```c
#include <stdio.h>
#include <stdlib.h>
/* 28 - for */
//for loop statement with semicolon
int main()
{
    int i,j,k;
    for(i=1,j=0,k=3;i<=5,j<=6,k>1;i++,j++,k--);
    // semicolon, it will be considered as statement
    {
     printf("i=%d j=%d k=%d\n",i,j,k);
    }
  getch();
}
```

```
"D:\1. COMPUTER NOTEBOOK\C LANGUAGE\C PROGRAMS\PART
i=3 j=2 k=1
```