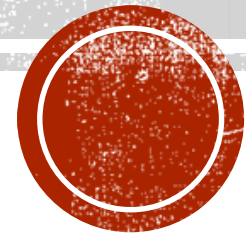


# PDS LAB: ASSIGNMENT 4

Dr. Pallab Dasgupta

Professor, CSE, IIT Kharagpur

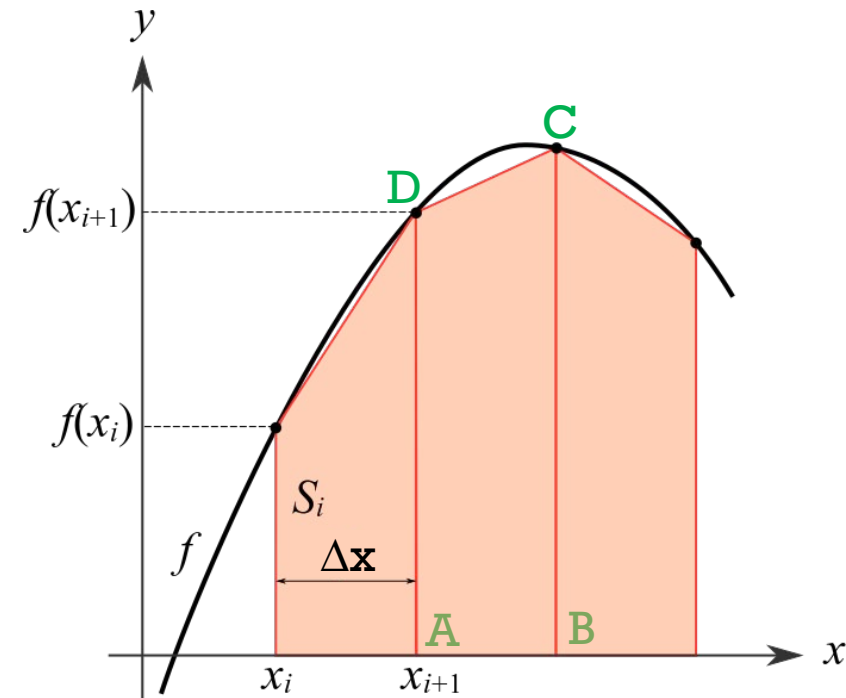


# TRAPEZOIDAL RULE

The trapezoidal rule is a technique for approximating the definite integral or to approximate the area under the curve.

## Steps:

- Divide  $x$  in multiple small sections; .
- Calculate the  $y$  value by calling the actual function.
- Now, consider each section as a trapezium; like **ABCD**.
- Calculate the area each trapezium by using the following formula:  
( $\Delta x * 0.5 * (AD + BC)$ )
- Then sum up the area of all the trapeziums to find the approximated value of integral.



$$\int_a^b f(x) dx \approx \sum_{k=1}^N \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k = \frac{\Delta x}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + \dots + 2f(x_{N-1}) + f(x_N))$$
$$= \Delta x \left( \sum_{k=1}^{N-1} f(x_k) + \frac{f(x_N) + f(x_0)}{2} \right)$$



# TAXI CAB NUMBER

In mathematics, the  $n$ th taxicab number, typically denoted  $Ta(n)$  or Taxicab( $n$ ), also called the  $n$ th Hardy–Ramanujan number, is defined as the smallest integer that can be expressed as a sum of two positive integer cubes in  $n$  distinct ways. The most famous taxicab number is:

$$1729 = Ta(2) = 1^3 + 12^3 = 9^3 + 10^3.$$

**The background story as told by G. H. Hardy:**

“I remember once going to see him [Ramanujan] when he was lying ill at Putney. I had ridden in taxi-cab No. 1729, and remarked that the number seemed to be rather a dull one, and that I hoped it was not an unfavourable omen. "No," he replied, "it is a very interesting number; it is the smallest number expressible as the sum of two [positive] cubes in two different ways.”




# WHAT IS DEBUGGING?

Debugging is the process of:

1. Finding and
2. Resolving

defects within a computer program.

```
function d($arg) {  
    var_dump(debug_...);  
    trigger_error(...);  
    trigger_error(...);  
    trigger_error(...);  
}  
  
a('alpha');
```



# WHAT IS A BUG?



# WHAT IS A BUG IN COMPUTER SCIENCE DOMAIN?



Ariane 5's first test flight (Ariane 5 Flight 501) on 4 June 1996 failed, with the rocket self-destructing 37 seconds after launch.

**Reason:** A data conversion from 64-bit floating point value to 16-bit signed integer value.

# BUG! HOW MUCH DOES IT COST?



Bad

Cost





# DEBUGGING STEPS

- Find and solve errors and warnings:

Line Number



Actual line of code



Character Number



Color Coding



Error



Warning

```
suntron@DESKTOP-36SJQJS: ~
suntron@DESKTOP-36SJQJS:~$ gcc a.c
a.c: In function 'main':
a.c:6:11: error: expected ')' before 'a'
scanf("%d")a);
          ^
a.c:6:9: warning: format '%d' expects a matching 'int *' argument [-Wformat=]
scanf("%d")a);
          ^
a.c:8:1: error: expected ';' before 'scanf'
scanf("%d",&c);
^~~~~~
a.c:9:3: warning: implicit declaration of function 'pow' [-Wimplicit-function-dec
e*pow((1+b/100),c)
    ^~~
a.c:9:3: warning: incompatible implicit declaration of built-in function 'pow'
a.c:9:3: note: include '<math.h>' or provide a declaration of 'pow'
a.c:10:1: error: expected ';' before 'printf'
printf("%d", &d);
^~~~~~
a.c:11:10: warning: format '%d' expects argument of type 'int', but argument 2 ha
printf("%d", &e);
    ~^  ~~
    %1s
suntron@DESKTOP-36SJQJS:~$
```

# DEBUGGING STEPS

```
Text Editor
Open
#include<stdio.h>
int main(){
int a,b,c,d,e;
d=a+a*b/100*c;
scanf("%d" a);
scanf("%d",&b);
scanf("%d",&c);
e=a*pow((1+b/100),c)
printf("%d", &d);
printf("%d", &e);
return 0;
}
```

```
suntron@DESKTOP-36SJQJS: ~
suntron@DESKTOP-36SJQJS:~$ gcc a.c
a.c: In function 'main':
a.c:6:11: error: expected ')' before 'a'
scanf("%d"a);
      ^
a.c:6:9: warning: format '%d' expects a matching 'int *' argument [-Wformat=]
scanf("%d"a);
      ~^
a.c:8:1: error: expected ';' before 'scanf'
scanf("%d",&c);
^~~~~
a.c:9:3: warning: implicit declaration of function 'pow' [-Wimplicit-function-dec
e*pow((1+b/100),c)
  ^~~
a.c:9:3: warning: incompatible implicit declaration of built-in function 'pow'
a.c:9:3: note: include '<math.h>' or provide a declaration of 'pow'
a.c:10:1: error: expected ';' before 'printf'
printf("%d", &d);
^~~~~
a.c:11:10: warning: format '%d' expects argument of type 'int', but argument 2 ha
printf("%d", &e);
      ~^  ~
          %}s
suntron@DESKTOP-36SJQJS:~$
```

# DEBUGGING STEPS

```
Text Editor
Open
#include<stdio.h>
int main(){
int a,b,c,d,e;
d=a+a*b/100*c;
scanf("%d" a);
scanf("%d",&b);
scanf("%d",&c);
e=a*pow((1+b/100),c)
printf("%d", &d);
printf("%d", &e);
return 0;
}
```

```
1/*****
2Created By      : Sumanta Dey
3Created Date   : 22-Jan-2020
4Modified On    : 27-Jan-2020
5Description    : Program to calculate simple and compound interest amount.
6*****/
7
8#include<stdio.h>
9#include<math.h>
10
11/*Function to calculate simple and compound interest amount.*/
12int main()
13{
14     float flt_principle_amount, flt_interest_rate, flt_tenure, flt_simple_interest_amount, flt_compound_interest_amount;
15
16     printf("Enter principle amount, interest rate(%), and tenure(year): \n");
17     scanf("%f", &flt_principle_amount);
18     scanf("%f",&flt_interest_rate);
19     scanf("%f",&flt_tenure);
20
21     /*Simple interest calculation*/
22     flt_simple_interest_amount = flt_principle_amount * (flt_interest_rate / 100) * flt_tenure;
23
24     /*Compound interest calculation*/
25     flt_compound_interest_amount = (flt_principle_amount * pow((1 +(flt_interest_rate/100)), flt_tenure)) - flt_principle_amount;
26
27     printf("Simple Interst amount is: %f\n", flt_simple_interest_amount);
28     printf("Compound Interst amount is: %f\n", flt_compound_interest_amount);
29
30     return 0;
31}
```

# HOW TO MAKE YOUR CODE UNDERSTANDABLE?

## 1. Proper comments

## 2. Proper naming of the variables

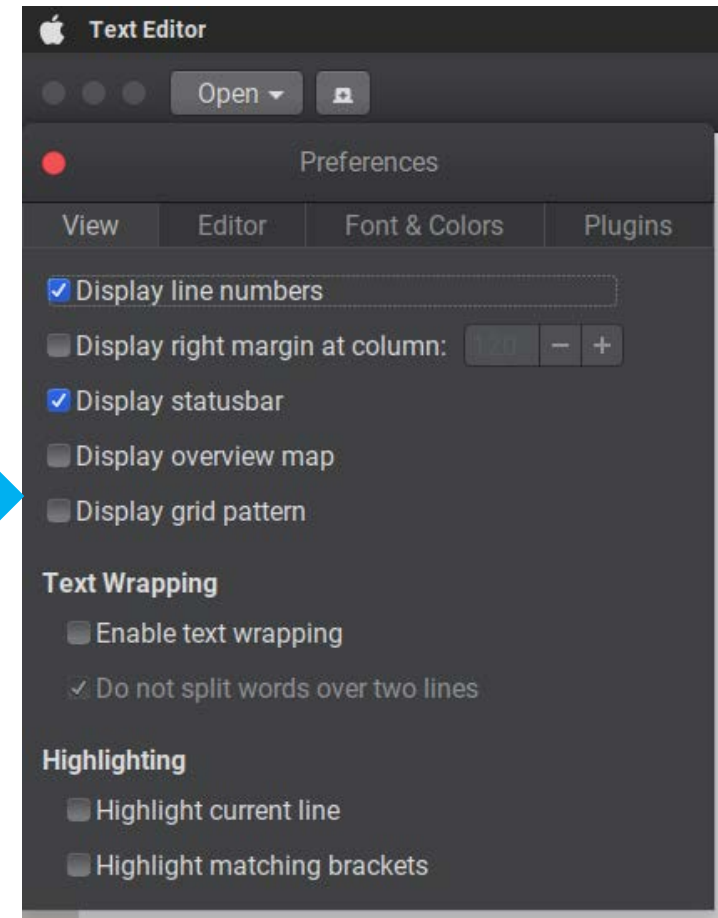
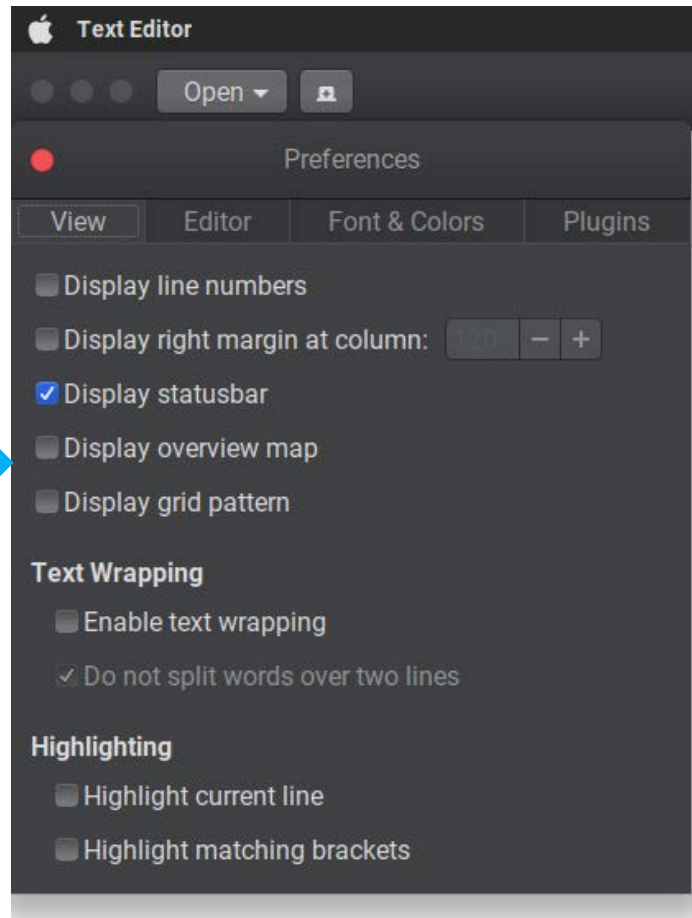
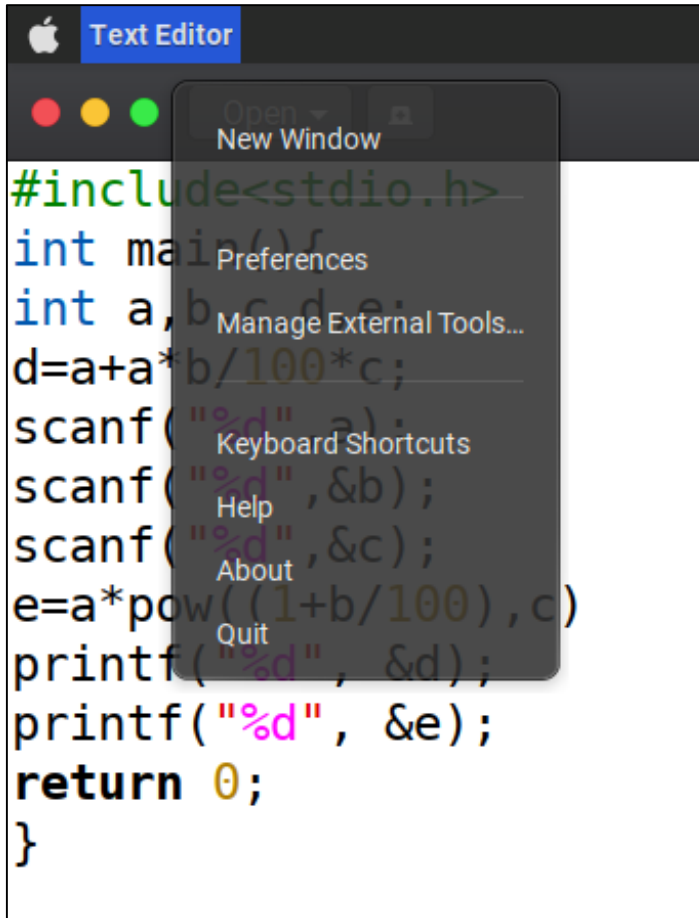
## 3. Proper Indentation

```
1/*****
2Created By      : Sumanta Dey
3Created Date   : 22-Jan-2020
4Modified On    : 27-Jan-2020
5Description    : Program to calculate simple and compound interest amount.
6*****/
7
8#include<stdio.h>
9#include<math.h>
10
11/*Function to calculate simple and compound interest amount.*/
12int main()
13{
14     float flt_principle_amount, flt_interest_rate, flt_tenure, flt_simple_interest_amount, flt_compound_interest_amount;
15
16     printf("Enter principle amount, interest rate(%), and tenure(year): \n");
17     scanf("%f", &flt_principle_amount);
18     scanf("%f",&flt_interest_rate);
19     scanf("%f",&flt_tenure);
20
21     /*Simple interest calculation*/
22     flt_simple_interest_amount = flt_principle_amount * (flt_interest_rate / 100) * flt_tenure;
23
24     /*Compound interest calculation*/
25     flt_compound_interest_amount = (flt_principle_amount * pow((1 +(flt_interest_rate/100)), flt_tenure)) - flt_principle_amount;
26
27     printf("Simple Interst amount is: %f\n", flt_simple_interest_amount);
28     printf("Compound Interst amount is: %f\n", flt_compound_interest_amount);
29
30     return 0;
31 }
```

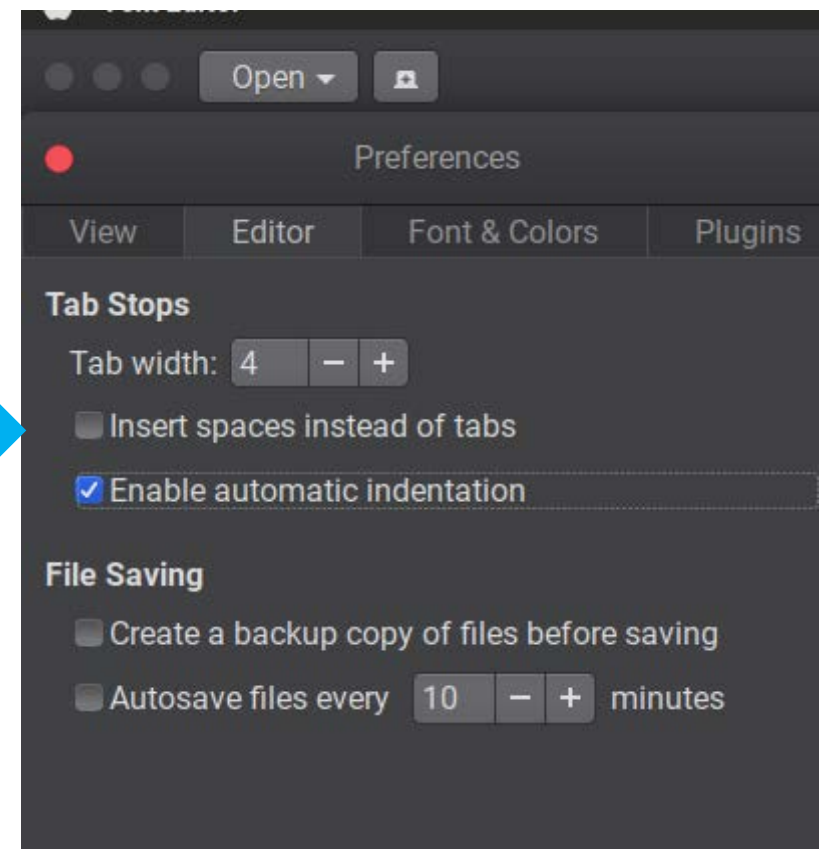
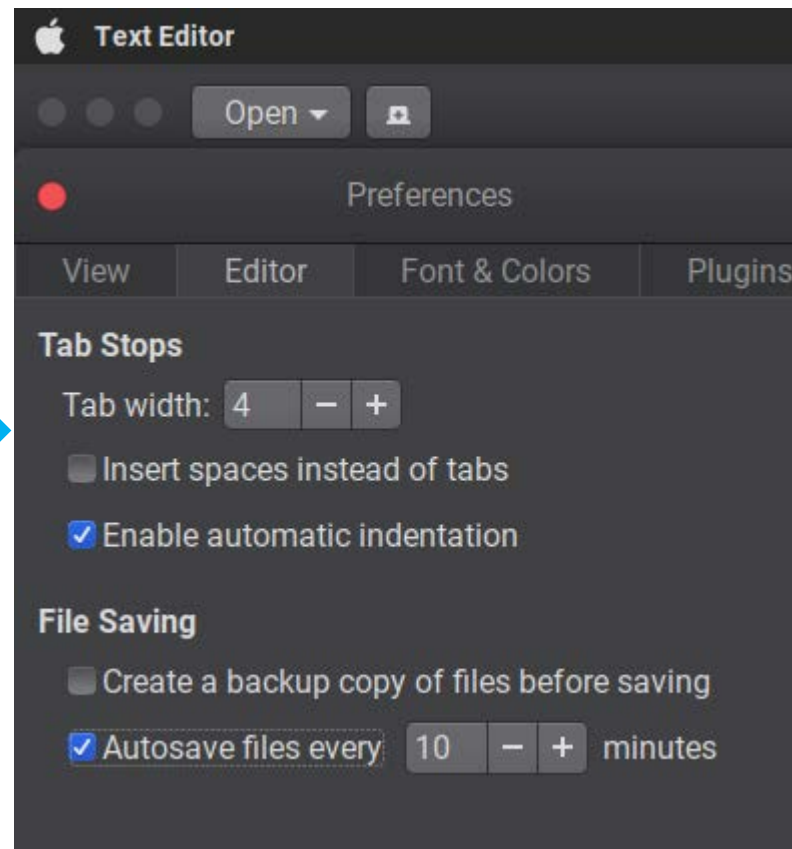
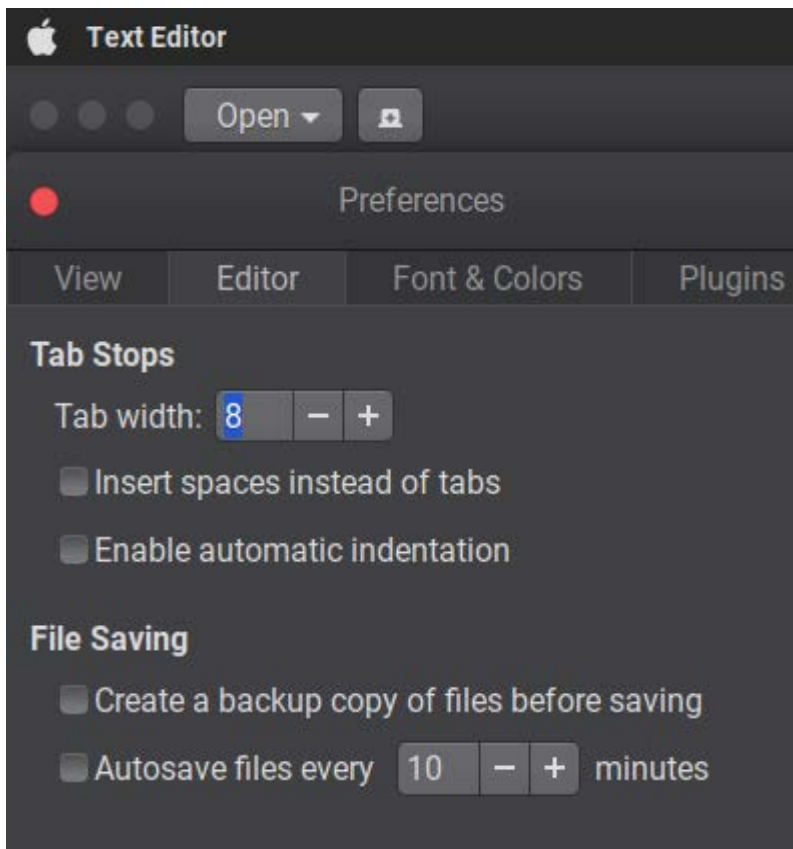


# **GEDIT SETUP**

# SETUP GEDIT - STEP 1



# SETUP GEDIT - STEP 2



# SETUP GEDIT - STEP 3

```
1 /*****
2 Created By : Sumanta Dey
3 Created On : 24-Jan-2020
4 Modified On : 24-Jan-2020
5 Description : Program to print cross pattern.
6 *****/
7 #include<stdio.h>
8
9 void main()
10 {
11     int row_index, col_index;
12
13     for(row_index=0; row_index<=10; row_index++)
14     {
15         for(col_index=0; col_index<=10; col_index++)
16         {
17             if(row_index == col_index)
18             {
19                 printf("* ");
20             }
21             else if(row_index+col_index == 10)
22             {
23                 printf("* ");
24             }
25             else
26             {
27                 printf(" ");
28             }
29         }
30         printf("\n");
31     }
32 }
```



# COMMON MISTAKES IN LOOPS

## Case 1: Giving comma instead of semicolon

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

**Result:**

**Syntax Error**

# COMMON MISTAKES IN LOOPS

**Case 2: Incrementing another variable j instead of i (the variable present in the condition)**

```
void main()
```

```
{
```

```
    int i, j;
```

```
    for(i=0, i<10, j++)
```

```
    {
```

```
        printf("%d, ",i);
```

```
    }
```

```
}
```

**Result:**

**0, 0, 0, 0, 0 ..... Infinite loop.**

# COMMON MISTAKES IN LOOPS

**Case 3: Loop followed by a ';' means it will loop that line only.**

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

**Result:**

**10**

# COMMON MISTAKES IN LOOPS

## Case 4: Using condition that may never satisfy

```
void main()
{
    int i;
    for(i=0; i==100; i++)
    {
        printf("%d", i);
    }
}
```

**Result:**

**Nothing**

# COMMON MISTAKES IN LOOPS

## Case 5: Using same variable in the nested loop

```
void main()
{
    int i;
    for(i=0; i<100; i++)
    {
        for(i=0; i<10; i++)
        {
            printf("%d,", i);
        }
    }
}
```

### Result:

**0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5..... infinite loop**

# COMMON MISTAKES IN LOOPS

## Case 6: Using same variable in the nested loop

```
void main()
{
    int i;
    for(i=0; i<10; i++)
    {
        for(i=0; i<10; i++)
        {
            printf("%d,", i);
        }
    }
}
```

**Result:**

**0,1,2,3,4,5,6,7,8,9**

# COMMON MISTAKES IN LOOP

1. `for(i=0,i<10,i++)` : Comma instead of semicolon
2. `for(i=0;i<10;j++)` : Incrementing another variable j instead of i (will leads to infinite loop).
3. `for(i=0;i<10;i++);` : Loop followed by a ';' means it will loop that line only.
4. `for(i=0;i==10;i++)` : Will not enter in the loop.
5. `for(i=0;i<10;i++){` : Using same variable name in two nested loop. Create confusions and errors.  
`for(i=0;i<10;i++){`  
`}`  
`}`
6. `while(i<10);` : Same as the for loop.
7. `while(i<10)` : Will leads to infinite loop.  
`{`  
`//No statement to increment or decrement the value of i.`  
`}`
8. For more do a [Google](#) search with “[Common Loop Mistakes in C](#)”.

# CODING : BEST PRACTICES

1. Assign proper names to the variables.
2. Use of proper indentation.
3. Always write the if-else and loops blocks inside {}.
4. Try to follow up all the warnings during compilation.
5. Giving proper comments.
6. Always remember the common mistakes, and try to avoid those. Like:
  - i. `scanf("%d", a);`
  - ii. `printf("%d", &a);`
  - iii. `if(a==5);`
  - iv. `if(a=5)`
  - v. "a" multiplied by "b" means "a\*b", not "ab" or "a.b"
  - vi. Using a variable before assigning its value.
  - vii. Using brackets for long mathematical expressions.
7. For more visit: <https://www.thecrazyprogrammer.com/2014/08/15-common-errors-in-c-and-cpp-programming.html> or just Google it.



**THANK YOU**

Happy Coding!!

