

## Solving Gates by **for Loop**

### 1) Simple/Basic Example of For-Loop

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

    /* The loop goes while i < 10, and i increases by one every loop*/

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

### 2) Using for Loop in 1-D Array(s) in **AND** Gate

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int A[4] = { 0, 0, 1, 1 };
    int B[4] = { 0, 1, 0, 1 };
    int i, output;

    for (i = 0; i < 5; i++) {
        // using '&&' Operator
```

```

output = A[i] && A[i];

printf("\n %d AND %d = %d",
      A[i], B[i], output);

}

}

```

## **2<sup>nd</sup> Method: By For-Loop and If-Else:**

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int A[4] = { 0, 0, 1, 1 };
    int B[4] = { 0, 1, 0, 1 };
    int i, output;

    for (i = 0; i < 5; i++) {

        if (A[i] == 0 && B[i] == 0)
            output = 0;

        else if (A[i] == 0 && B[i] == 1)
            output = 0;

        else if (A[i] == 1 && B[i] == 0)
            output = 0;

        else
            output = 1;

        printf("\n %d AND %d = %d", A[i], B[i], output );
    }
}

```

```
}
```

### 3) Using For-Loop in 2-D Array(s) in **OR** Gates

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

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int A[2][4] = {{ 0, 0, 1, 1 },  
                   { 0, 0, 1, 1 }};
```

```
    int B[2][4] = {{ 0, 1, 0, 1 },  
                   { 0, 1, 0, 1 }};
```

```
    int i, j, X;
```

```
    for (i = 0; i < 2; i++) {    //Loop for Row
```

```
        for (j=0; j<4; j++) {    //Loop for Column
```

```
            X = A[i][j] || B[i][j];
```

```
            printf("\n %d OR %d = %d" , A[i][j], B[i][j], X);
```

```
        }
```

```
    }
```

```
}
```