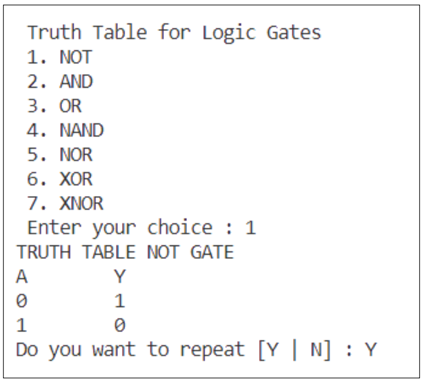
# Practical No. 06

Program:

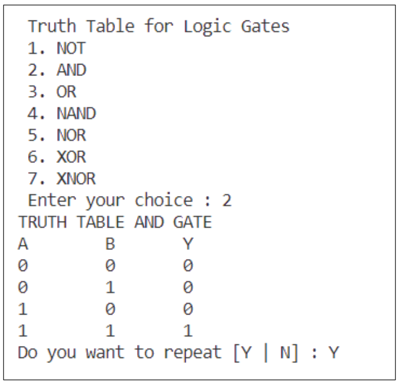
```
// Write C program to implement Truth table for logic gates.

#include<stdio.h>

void not()
{
  int a[2] = {0, 1};
  int i, j;
  printf("TRUTH TABLE NOT GATE\n");
  printf("A\t Y\n");
  for(i = 0; i < 2; i++)
  {
    printf("%d\t %d\n", a[i], !a[i]);
  }
}

void and()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE AND GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], a[i]&&b[i]);
  }
}

void or()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE OR GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], a[i]||b[i]);
  }
}

void nand()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE NAND GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], !(a[i]&&b[i]));
  }
}

void nor()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE NOR GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], !(a[i]||b[i]));
  }
}

void xor()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE XOR GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], a[i] ^ b[i]);
  }
}

void xnor()
{
  int a[4] = {0,0,1,1};
  int b[4] = {0,1,0,1};
  int i, j;
  printf("TRUTH TABLE XNOR GATE\n");
  printf("A\t B\t Y\n");
  for(i = 0, j = 0; i < 4, j < 4; i++, j++)
  {
    printf("%d\t %d\t %d\n", a[i], b[i], !(a[i] ^ b[i]));
  }
}
```

```
int main()
{
  int choice;
  char ch;

  do{
    printf("\n Truth Table for Logic Gates\n");
    printf(" 1. NOT\n 2. AND\n 3. OR\n 4. NAND\n 5. NOR\n 6. XOR\n 7. XNOR\n
");
    printf("Enter your choice : ");
    scanf("%d", &choice);

    switch(choice)
    {
      case 1: not();
      break;
      case 2: and();
      break;
      case 3: or();
      break;
      case 4: nand();
      break;
      case 5: nor();
      break;
      case 6: xor();
      break;
      case 7: xnor();
      break;
      default:
      printf("\n Enter Valid Input...");
    }

      printf("Do you want to repeat [Y | N] : ");
      scanf(" %c" ,&ch);

    }while(ch=='Y' || ch=='y');


  return 0;
}
```

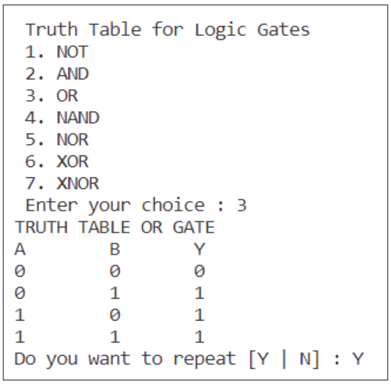
Output :



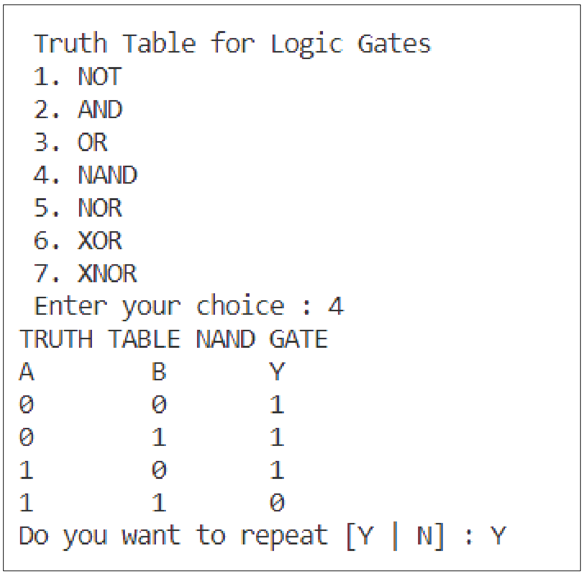
```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 1
TRUTH TABLE NOT GATE
A        Y
0        1
1        0
Do you want to repeat [Y | N] : Y
```



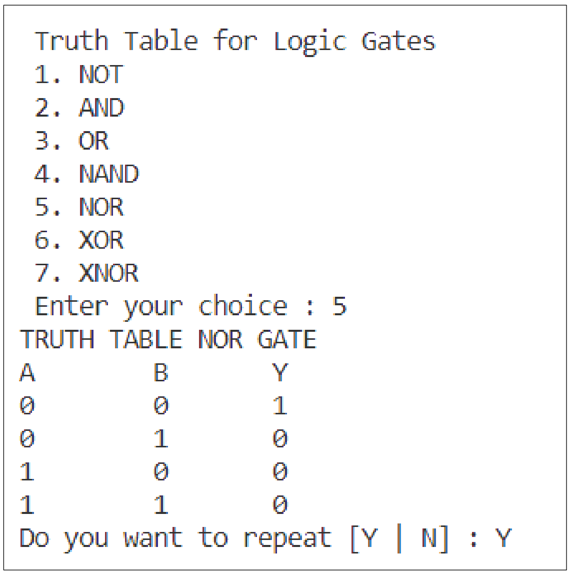
```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 2
TRUTH TABLE AND GATE
A          B         Y
0          0         0
0          1         0
1          0         0
1          1         1
Do you want to repeat [Y | N] : Y
```



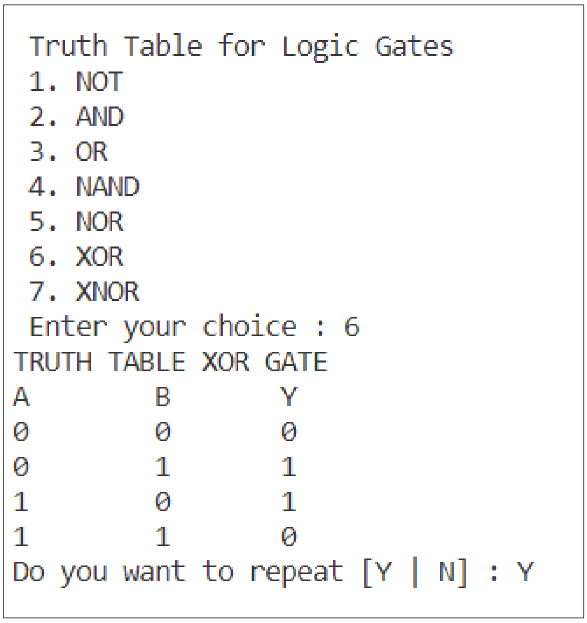
```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 3
TRUTH TABLE OR GATE
A          B         Y
0          0         0
0          1         1
1          0         1
1          1         1
Do you want to repeat [Y | N] : Y
```



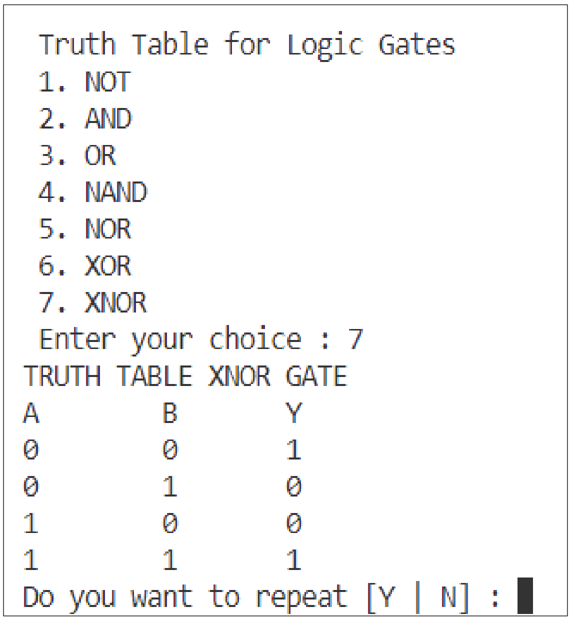
```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 4
TRUTH TABLE NAND GATE
A         B         Y
0         0         1
0         1         1
1         0         1
1         1         0
Do you want to repeat [Y | N] : Y
```



```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 5
TRUTH TABLE NOR GATE
A         B         Y
0         0         1
0         1         0
1         0         0
1         1         0
Do you want to repeat [Y | N] : Y
```



```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 6
TRUTH TABLE XOR GATE
A         B         Y
0         0         0
0         1         1
1         0         1
1         1         0
Do you want to repeat [Y | N] : Y
```



```
 Truth Table for Logic Gates
 1. NOT
 2. AND
 3. OR
 4. NAND
 5. NOR
 6. XOR
 7. XNOR
 Enter your choice : 7
TRUTH TABLE XNOR GATE
A         B         Y
0         0         1
0         1         0
1         0         0
1         1         1
Do you want to repeat [Y | N] :
```