# Questions On Operators With Solutions

**Q1. Write a C program to find maximum between two numbers using conditional operator.**

```
// for greatest two numbers using conditional.
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d%d”,&a,&b);
a>b?printf(“a is greatest”):printf(“b is greatest”);
}
```

```
//using third variable.
void main()
{
int a,b,c;
scanf(“%d%d”,&a,&b);
c=a>b?a:b;
printf(“This is the greatest among two :%d”,c);
}
```

**Q2. Write a C program to find maximum between three numbers using conditional operator.**

```
//greatest among three using another variable.
void main()
{
int a,b,c,d;
scanf(“%d%d%d”,&a,&b,&c);
d=a>b?a:b;
d=c>d?c:d;
printf(“This is the greatest among two :%d”,d);
}
```

```
// greatest among three without another variable.
void main()
{
int a,b,c;
scanf(“%d%d%d”,&a,&b,&c);
a>b?((a>c)?printf(” a is gretest”):printf(“C is greatest”)):((b>c)?printf(“b is
greatest”):printf(“c is greatest”));
}
```

**Q3. Write a C program to find maximum between four numbers using conditional operator.**

```
// greatest among four
#include<stdio.h>
void main()
{
int a,b,c,d;
scanf(“%d%d%d%d”,&a,&b,&c,&d);
(a>b?(a>c?(a>d?printf(” a is gretest”):printf(“d is greatest”)):c>d?printf(“c is
greatest”):printf(“d is greatest”)):b>c?(b>d?printf(“b is greatest”):printf(“d is
greatest”)):c>d?printf(“c is greatest”):printf(“d is greatest”));
}
```

**Q4. Write a C program to check whether a number is even or odd using conditional operator.**

```
// even odd using bitwise by conditional
#include<stdio.h>
void main()
{
int a;
scanf(“%d”,&a);
a&1?printf(“inputed number is odd”):printf(“the inputed number is even”);
}
```

**Q5. Write a C program to check whether year is leap year or not using conditional operator.**

```
#include <stdio.h>
int main()
{
int year;
/* Input year from user */
printf("Enter any year: ");
scanf("%d", &year);

/* If year%4==0 and year%100==0 then print leap year else if year%400==0 then
print leap year else print common year */

(year%4==0 && year%100!=0) ? printf("LEAP YEAR") :
(year%400 ==0 ) ? printf("LEAP YEAR") : printf("COMMON YEAR");
return 0;
}
```

**Q6. Write a C program to check whether character is an alphabet or not using conditional operator.**

```
  #include <stdio.h>
  int main()
  {
  char ch;
  /* Input character from user    */
  printf("Enter any character: ");
  scanf("%c", &ch);
  /*
 * If (ch>'a' and ch<'z') or (ch>'A' and ch<'Z') then
 *     print alphabet
 * else
 *     print not alphabet
 */
  (ch>='a' && ch<='z') || (ch>='A' && ch<='Z')
   ? printf("It is ALPHABET")
   : printf("It is NOT ALPHABET");
return 0;
  }
```

**Q7. Write a C program to check Least Significant Bit (LSB) of a number is a set or not.**

```
#include <stdio.h>
int main()
{
int num;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* If (num & 1) evaluates to 1  as per bitwise & rule working for particular bit level
which i discussed in class*/
if(num & 1)
printf(“LSB of %d is set (1).”, num);
else
printf(“LSB of %d is unset (0).”, num);
return 0;
}
```

**Q8. Write a C program to check Most Significant Bit (MSB) of a number is a set or not.**

```
#include <stdio.h>
#define BITS sizeof(int) * 8
/* Total bits required to represent integer in memory so instead of long int i used
bits size to make my program more easy so this is also a directive defined by user so
we used #define  if it was part of compiler so we used  #include*/
int main()
{
int num, msb;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Move first bit of 1 to highest order */
msb = 1 << (BITS – 1);
/* Perform bitwise AND with msb and num */
if(num & msb)
printf(“MSB of %d is set (1).”, num);
else
printf(“MSB of %d is unset (0).”, num);
return 0;
}
```

**Q9. Write a C program to get nth bit of a number.**

```
#include <stdio.h>
int main()
{
int num, n, bitStatus;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Input bit position you want to check */
printf(“Enter nth bit to check (0-31): “);
scanf(“%d”, &n);
/* Right shift num, n times and perform bitwise AND with 1 */
bitStatus = (num >> n) & 1;
printf(“The %d bit is set to %d”, n, bitStatus);
return 0;
}
```

**Q10. Write a C program to set nth bit of a number.**

```
#include <stdio.h>
int main()
{
int num, n, newNum;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Input bit position you want to set */
printf(“Enter nth bit to set (0-31): “);
scanf(“%d”, &n);
/* Left shift 1, n times and perform bitwise OR with num */
newNum = (1 << n) | num;
printf(“Bit set successfully.\n\n”);
printf(“Number before setting %d bit: %d (in decimal)\n”, n, num);
printf(“Number after setting %d bit: %d (in decimal)\n”, n, newNum);
return 0;
}
```

**Q11. Write a C program to clear nth bit of a number.**

```
#include <stdio.h>
int main()
{
int num, n, newNum;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Input bit number you want to clear */
printf(“Enter nth bit to clear (0-31): “);
scanf(“%d”, &n);
/*
* Left shifts 1 to n times
* Perform complement of above
* finally perform bitwise AND with num and result of above
*/
newNum = num & (~(1 << n));
printf(“Bit cleared successfully.\n\n”);
printf(“Number before clearing %d bit: %d (in decimal)\n”, n, num);
printf(“Number after clearing %d bit: %d (in decimal)\n”, n, newNum);
return 0;
}
```

**Q12. Write a C program to toggle nth bit of a number.**

```
#include <stdio.h>
int main()
{
int num, n, newNum;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Input bit position you want to toggle */
printf(“Enter nth bit to toggle (0-31): “);
scanf(“%d”, &n);
/*
* Left shifts 1, n times
* then perform bitwise XOR with num
*/
newNum = num ^ (1 << n);
printf(“Bit toggled successfully.\n\n”);
printf(“Number before toggling %d bit: %d (in decimal)\n”, n, num);
printf(“Number after toggling %d bit: %d (in decimal)\n”, n, newNum);
return 0;
}
```

**Q13. Write a C program to get lowest set bit of a number.**

```
#include <stdio.h>
#define INT_SIZE sizeof(int) * 8 /* Integer size in bits */
int main()
{
int num, order, i;
printf(“Enter any number: “);
scanf(“%d”, &num);
/* Initially set the order to max size of integer */
order = INT_SIZE – 1;
/* Iterate through each bit of integer */
for(i=0; i<INT_SIZE; i++)
{
/* If current bit is set */
if((num>>i) & 1)
{
order = i;
/* Terminate the loop */
break;
}
}
printf("Lowest order set bit in %d is %d", num, order);
return 0;
}
```

**Q14. Write a C program to count trailing zeros in a binary number.**

```
#include <stdio.h>
#define INT_SIZE sizeof(int) * 8 /* Bits required to represent an integer */
int main()
{
int num, count, i;
printf("Enter any number: ");
scanf("%d", &num);
count = 0;
/* Iterate over each bit of the number */
for(i=0; i<INT_SIZE; i++)
{
/* If set bit is found the terminate from loop*/
if((num >> i ) & 1)
{
/* Terminate from loop */
break;
}

/* Increment trailing zeros count */
count++;
}
printf("Total number of trailing zeros in %d is %d.", num, count);
return 0;
}
```

**Q15. Write a C program to flip bits of a binary number using bitwise operator.**

```
#include <stdio.h>
int main()
{
int num, flippedNumber;
printf("Enter any number: ");
scanf("%d", &num);
flippedNumber = ~num;
printf("Original number = %d (in decimal)\n", num);
printf("Number after bits are flipped = %d (in decimal)", flippedNumber);
return 0;
}
```

**Q16. Write a C program to swap two numbers using bitwise operator.**

```
#include <stdio.h>
int main()
{
int num1, num2;
printf(“Enter any two numbers: “);
scanf(“%d%d”, &num1, &num2);
printf(“Original value of num1 = %d\n”, num1);
printf(“Original value of num2 = %d\n”, num2);

/* Swap two numbers */
num1 ^= num2;
num2 ^= num1;
num1 ^= num2;
printf(“Num1 after swapping = %d\n”, num1);
printf(“Num2 after swapping = %d\n”, num2);
return 0;
}
```

**Q17. Write a C program to check whether a number is even or odd using bitwise operator.**

```
#include <stdio.h>
int main()
{
int num;
printf(“Enter any number: “);
scanf(“%d”, &num);
if(num & 1)
{
printf(“%d is odd.”, num);
}
else
{
printf(“%d is even.”, num);
}
return 0;
}
```

## BITWISE Operator With Conditional

**//multiply any number by 64 with bitwise**

```
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
b=a<<6;
printf(“%d”,b);
}
```

```
// multiply any number by 63 using bitwise
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
b=a<<6;
b=b-a;
printf(“%d”,b);
}
```

```
//multiply any number by 65 using bitwise
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
b=a<<6;
b=b+a;
printf(“%d”,b);
}
```

```
// divide any number by 32 using bitwise
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
b=a>>5;
printf(“%d”,b);
}
```

```
//to check lsb using bitwise
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
a&1?printf(“lsb is 1”):printf(“lsb is 0”);
}
```

```
//to check msb using bitwise using conditional operator
#include<stdio.h>
void main()
{
int a,b;
scanf(“%d”,&a);
b=sizeof(int)*8;
b=1<<(b-1);
a&b?printf(“msb is 1”):printf(“msb is 0”);
}
```

**// to check any bit status including lsb or msb using conditional operator**

```
#include<stdio.h>
void main()
{
int a,b,pos;
printf(“enter any number\n”);
scanf(“%d”,&a);
b=(sizeof(int)*8)-1;
printf(“enter the bit position which u want to check from 0 to %d”,b);
scanf(“%d”,&pos);
((a>>pos)&1)?printf(“the status of bit at position %d is 1”,pos):printf(“the status of bit at position %d is 0”,pos);
}
```

**//to change the status of any bit using conditional operator**

```
#include<stdio.h>
void main()
{
int a,b,pos;
printf(“enter any number\n”);
scanf(“%d”,&a);
b=(sizeof(int)*8)-1;
printf(“enter the bit position which u want to check from 0 to %d”,b);
scanf(“%d”,&pos);
int newnum=(1<<pos)|a;
printf(“bit set successfully\n”);
printf(“number before setting bit position 0 is %d\n”,a);
printf(“number after setting that bit is %d\n”,newnum);
}
```