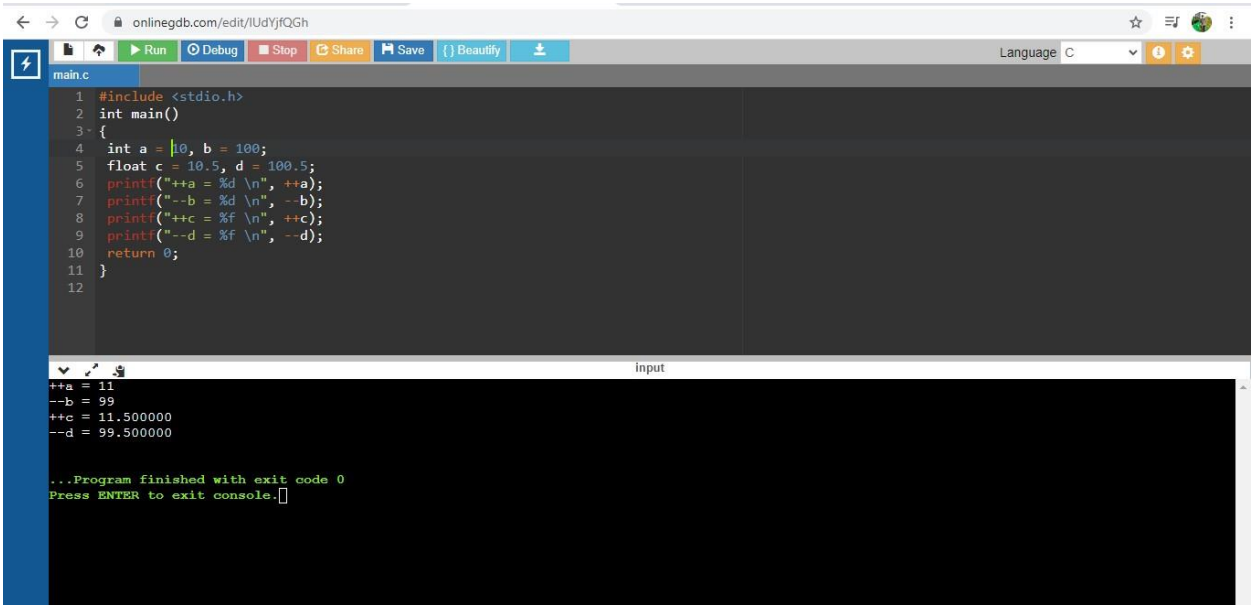
1)Type the following code and observe the output to know the working of increment and decrement operators

```
#include <stdio.h>

int main()

{

 int a = 10, b = 100;

 float c = 10.5, d = 100.5;

 printf("++a = %d \n", ++a);

 printf("--b = %d \n", --b);

 printf("++c = %f \n", ++c);

 printf("--d = %f \n", --d);

 return 0;

}
```

Output:

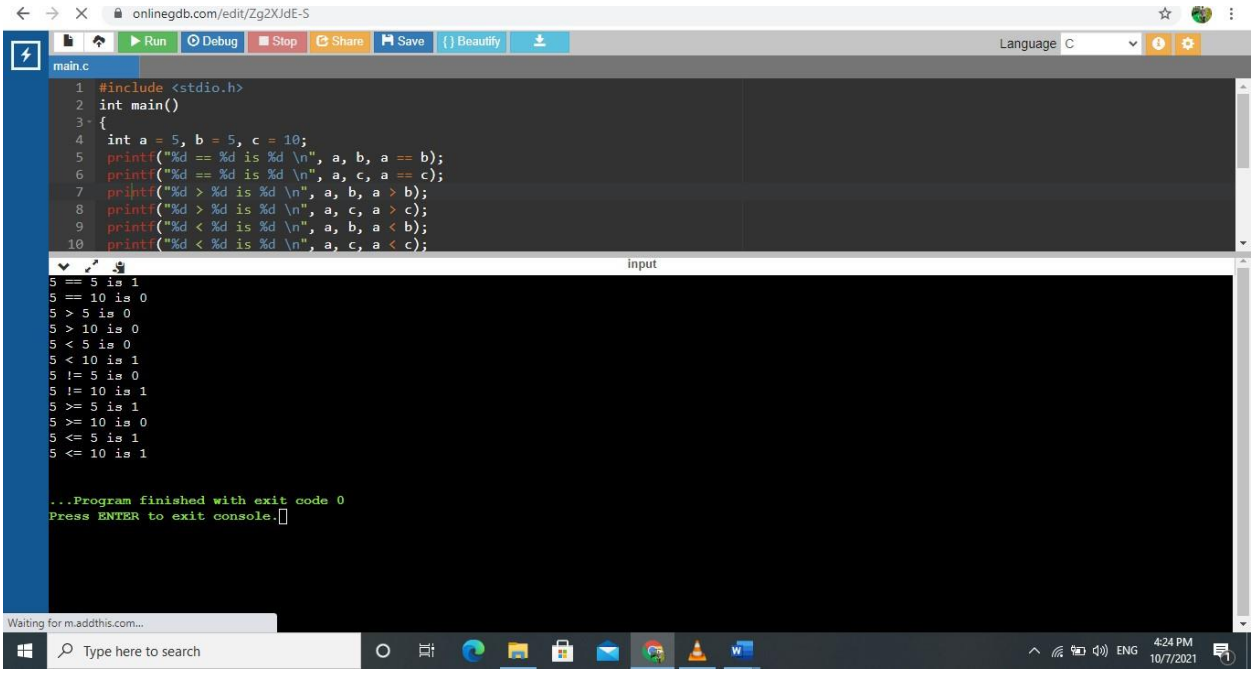


2) Type the following code and observe the output to know the working of relational operators

```
#include <stdio.h>

int main()

{

 int a = 5, b = 5, c = 10;
```

```
 printf("%d == %d is %d \n", a, b, a == b);
 printf("%d == %d is %d \n", a, c, a == c);
 printf("%d > %d is %d \n", a, b, a > b);
 printf("%d > %d is %d \n", a, c, a > c);
 printf("%d < %d is %d \n", a, b, a < b);
 printf("%d < %d is %d \n", a, c, a < c);
 printf("%d != %d is %d \n", a, b, a != b);
 printf("%d != %d is %d \n", a, c, a != c);
 printf("%d >= %d is %d \n", a, b, a >= b);
 printf("%d >= %d is %d \n", a, c, a >= c);
 printf("%d <= %d is %d \n", a, b, a <= b);
 printf("%d <= %d is %d \n", a, c, a <= c);
 return 0;
}
```

Output:

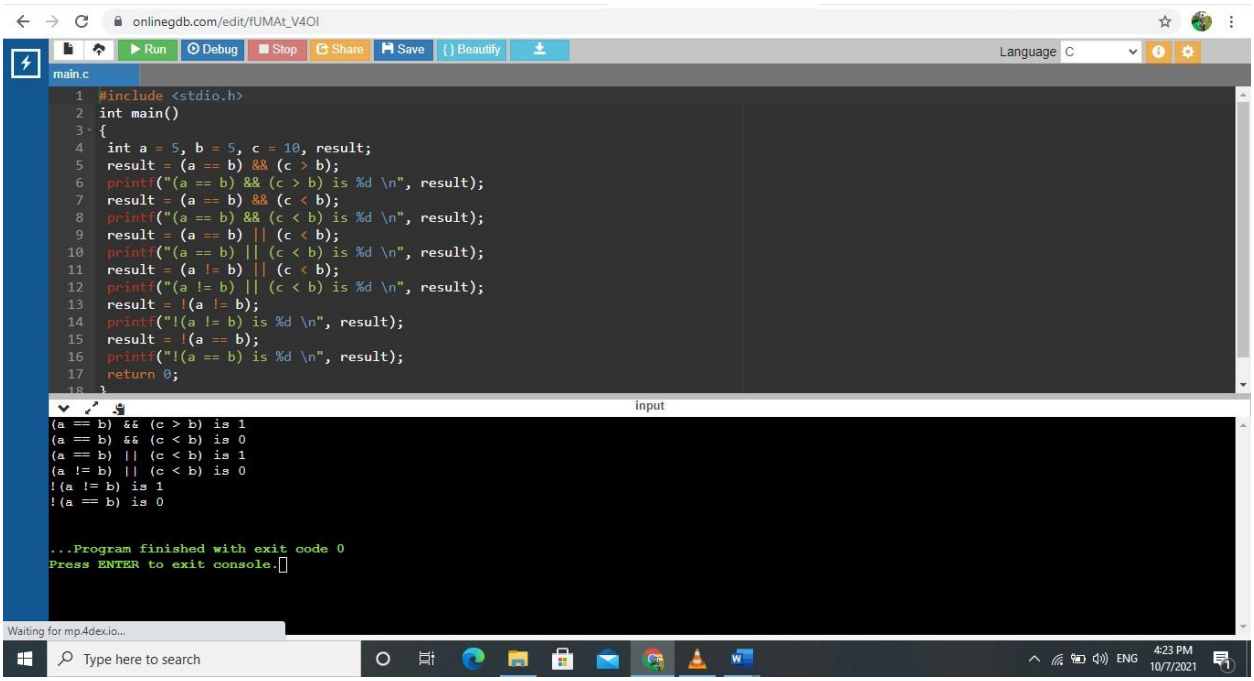


3) Type the following code and observe the output to know the working of logical operators

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

```
int main()
{
 int a = 5, b = 5, c = 10, result;
 result = (a == b) && (c > b);
 printf("(a == b) && (c > b) is %d \n", result);
 result = (a == b) && (c < b);
 printf("(a == b) && (c < b) is %d \n", result);
 result = (a == b) || (c < b);
 printf("(a == b) || (c < b) is %d \n", result);
 result = (a != b) || (c < b);
 printf("(a != b) || (c < b) is %d \n", result);
 result = !(a != b);
 printf("!(a != b) is %d \n", result);
 result = !(a == b);
 printf("!(a == b) is %d \n", result);
 return 0;
}
```

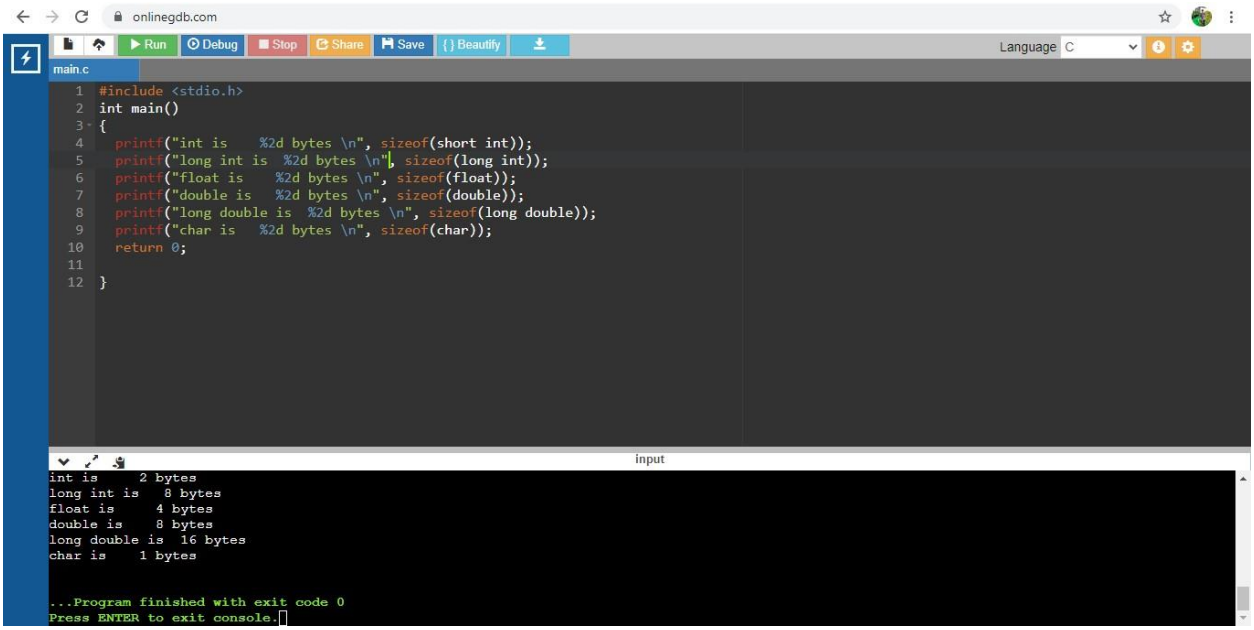
Output:



4) Write a C program that displays the size of all possible data types in C

```
#include <stdio.h>

int main()

{

  printf("int is    %2d bytes \n", sizeof(short int));

  printf("long int is  %2d bytes \n", sizeof(long int));

  printf("float is    %2d bytes \n", sizeof(float));

  printf("double is   %2d bytes \n", sizeof(double));

  printf("long double is  %2d bytes \n", sizeof(long double));

  printf("char is   %2d bytes \n", sizeof(char));

  return 0;


}
```

Output:

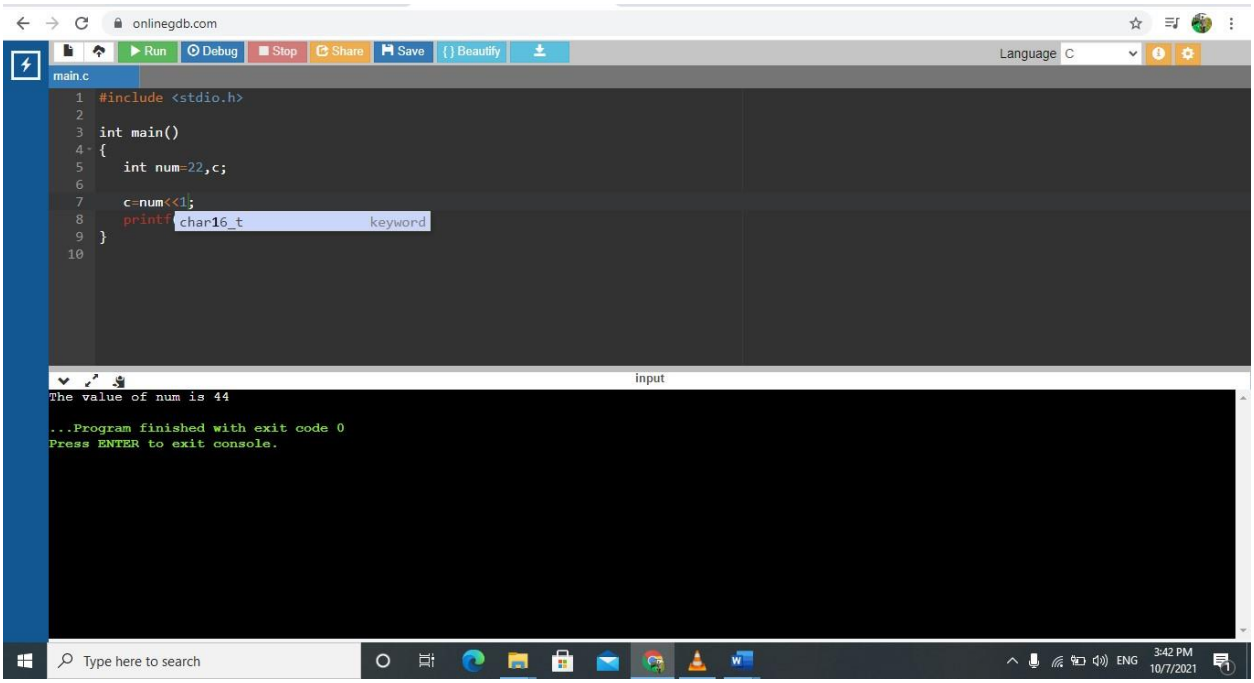


5) Let the variable num=22. Show the working (on paper) of left shift operator on num, i.e. num<<1,num<<2…. Validate the results by writing a program. Repeat the above to see the working of right shift operator as well

num<<1

```
#include <stdio.h>


int main()

{

  int num=22,c;


  c=num<<1;

  printf("The value of num is %d",c);

}
```

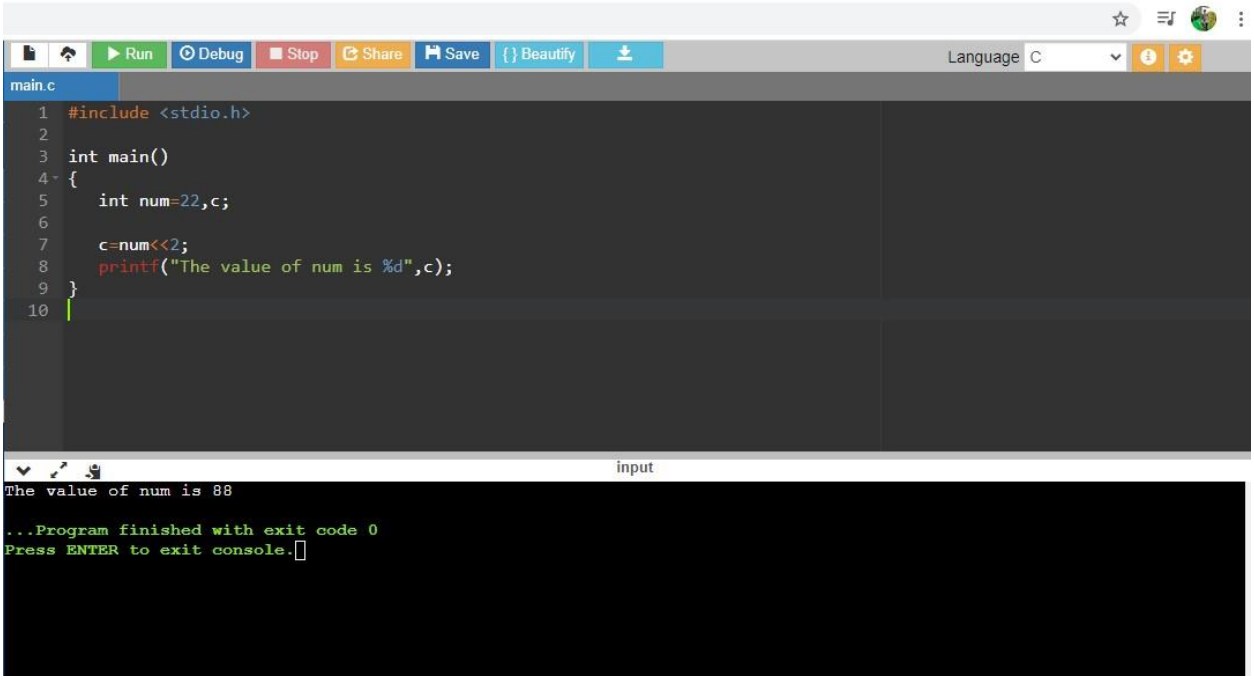
Output:



Num<<2

```
#include <stdio.h>


int main()
{
  int num=22,c;


  c=num<<2;
  printf("The value of num is %d",c);
}
```

Output:



Num>>1

#include <stdio.h>

int main()

{

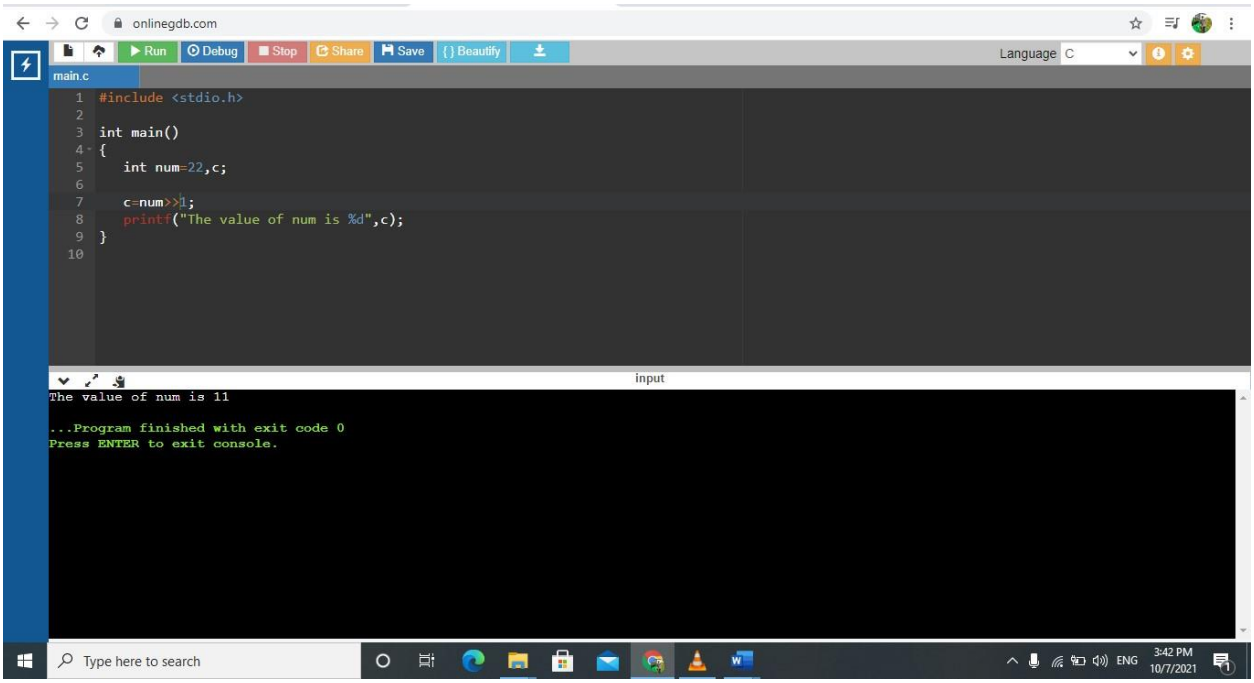
int num=22,c;

c=num>>1;

printf("The value of num is %d",c);

}

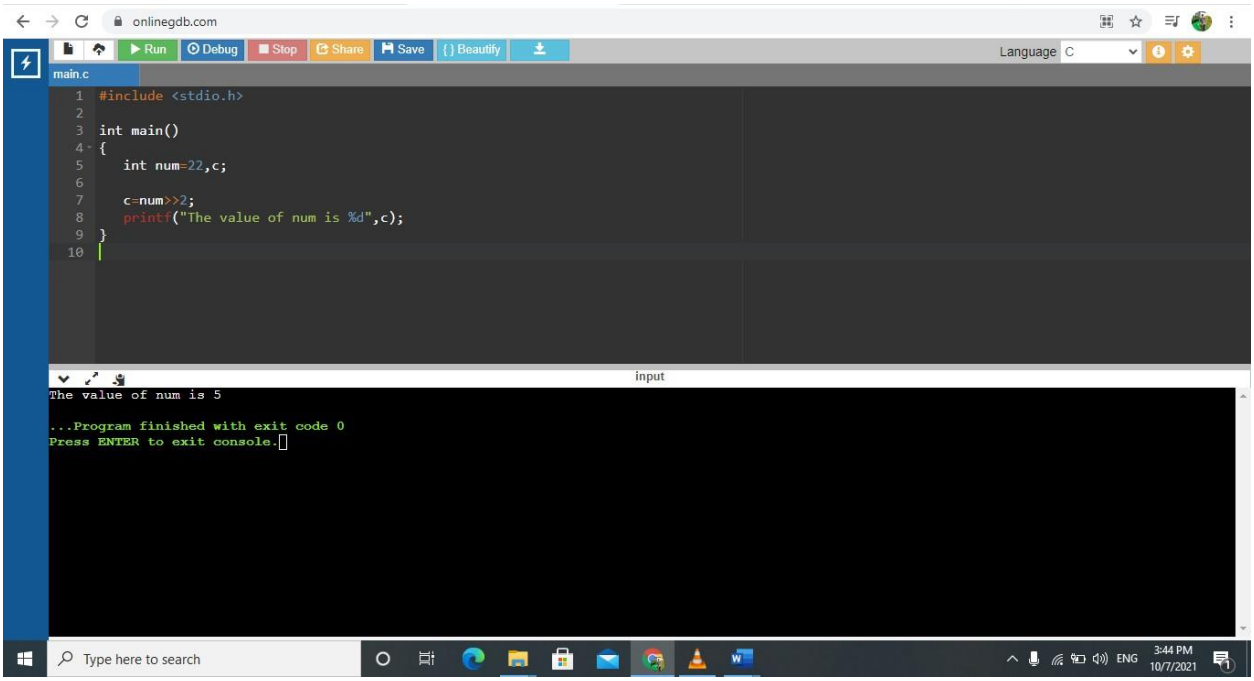
Output:

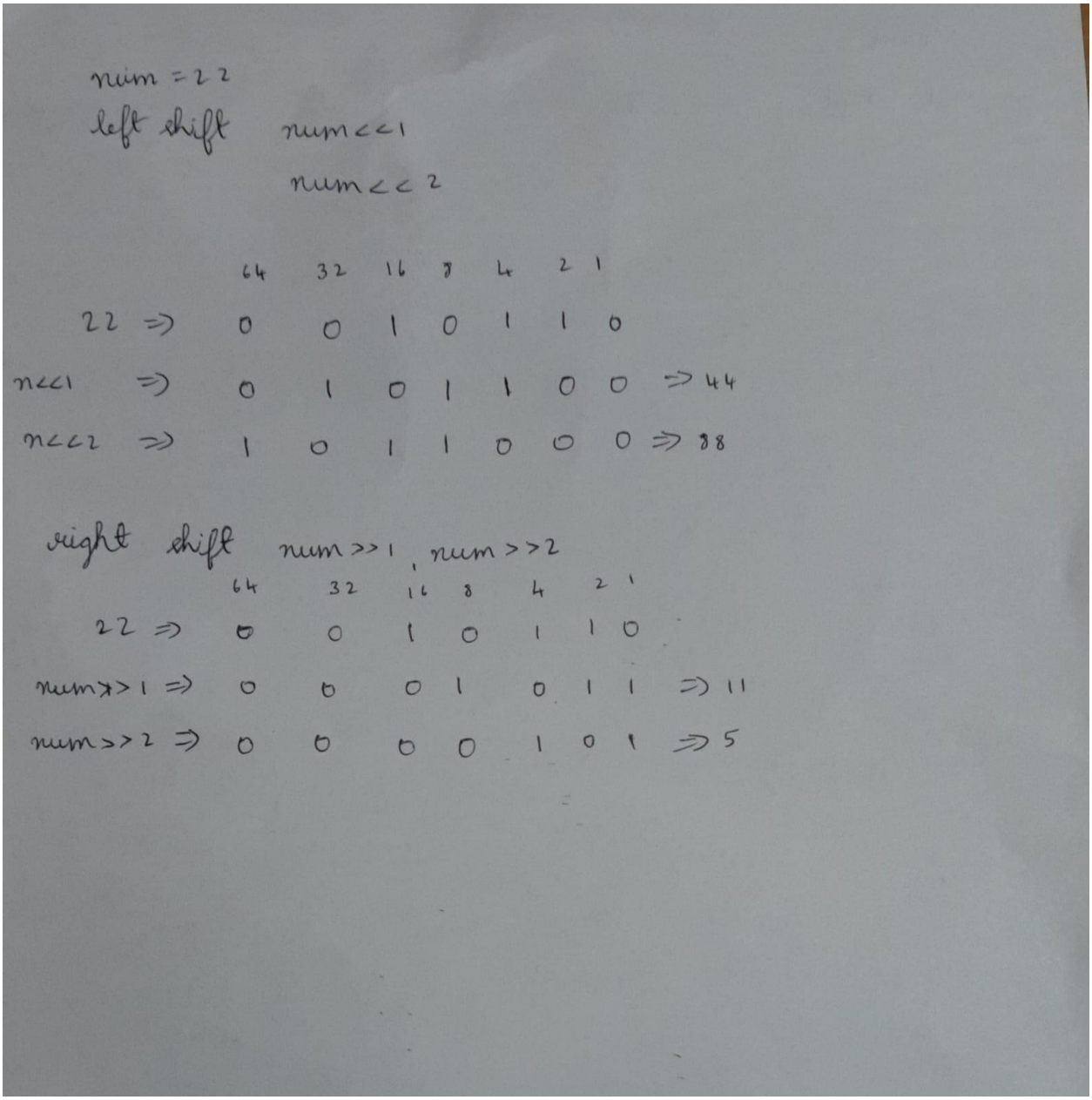


Num>>2

```
#include <stdio.h>


int main()

{

  int num=22,c;


  c=num>>2;

  printf("The value of num is %d",c);

}
```

Output:


onlinegdb.com
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
#include <stdio.h>
int main()
{
int num=22,c;
c=num>>2;
printf("The value of num is %d",c);
}
input
The value of num is 5
...Program finished with exit code 0
Press ENTER to exit console.
Type here to search
ENG
3:44 PM
10/7/2021

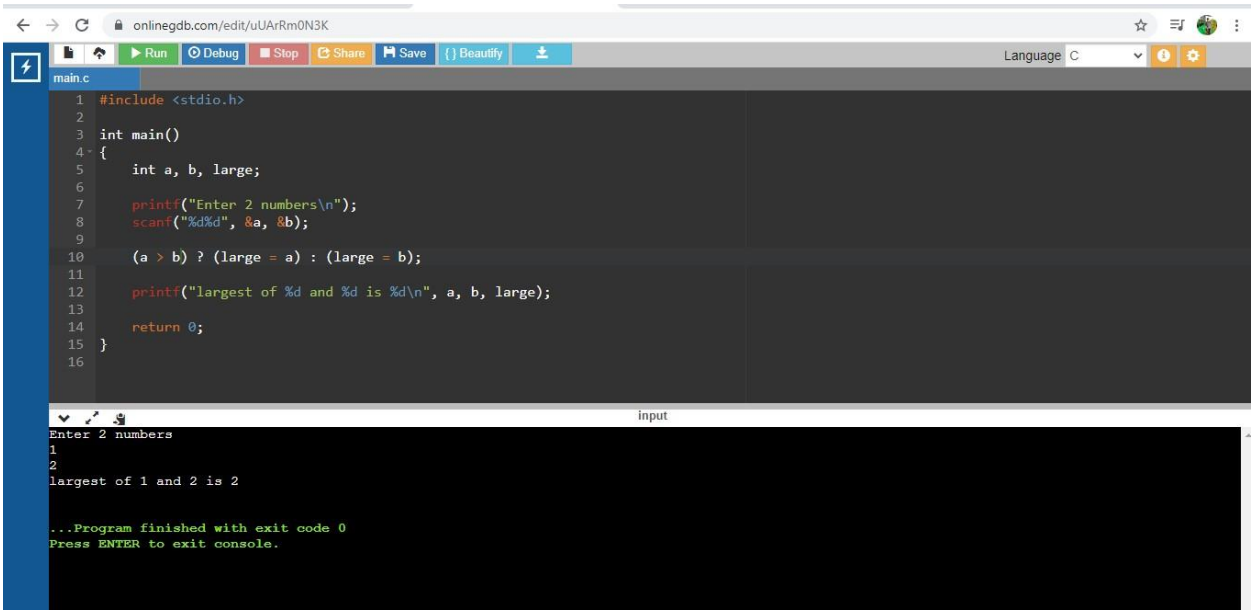


6)Write a program to determine the largest of two numbers using ternary operator. Enhance the code to determine largest of three numbers. (if statements are not allowed)

```
#include <stdio.h>


int main()

{

   int a, b, large;
```

```
    printf("Enter 2 numbers\n");
    scanf("%d%d", &a, &b);

    (a > b) ? (large = a) : (large = b);

    printf("largest of %d and %d is %d\n", a, b, large);

    return 0;
}
```

Output:




```
# include <stdio.h>

void main()
{
    int a, b, c, large ;

    printf("Enter three numbers : ") ;
```

```
    scanf("%d %d %d", &a, &b, &c) ;

    large = a > b ? (a > c ? a : c) : (b > c ? b : c) ;

    printf("\nThe largest number is : %d", large) ;
}
```

Output: