POLITEKNIK MALAYSIA

C PROGRAMMING *e-notes*

WRITTEN BY:
HABIBAH BINTI REMELI
MOHD ZAHARI BIN PUTEH
ABDUL MUADZ BIN ABDUL RAHIM

Politeknik Sultan Mizan Zainal Abidin
Km 8, Jalan Paka,
23000 Dungun
Terengganu Darul Iman
Telephone: +609-8400800
Fax: +609-8458781
Website: http://www.psmza.edu.my

HABIBAH BINTI REMELI
MOHD ZAHARI BIN PUTEH
ABDUL MUADZ BIN ABDUL RAHIM

PREFACE

Praise our gratitude to the presence of God Almighty.By His grace and guidance, the author was able to complete a scientific work entitled C PROGRAMMING. The author would like to thank our editorial team very warmly. WE also have worked extremely hard and with a lot of dedication to make this e-book a success. It was a pleasure working with our teams.

Any information that enables the publisher to correct any errors or submit any materials in future is welcome.

Thank you again.

ABSTRACT

C Programming course provides an introduction to programme design and development. Student will learn to design, code, debug, test and document wellstructured programs based on technical and engineering problem. Topic covered; software development principle, programming language basic, data types and input and output operation.

TABLE OF CONTENT

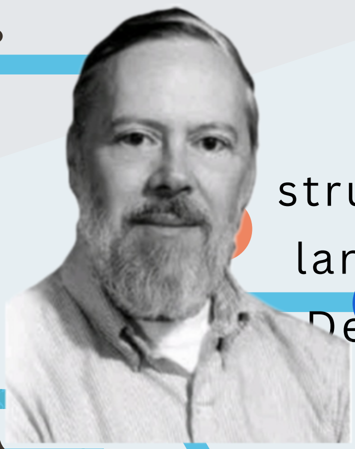
OVERVIEW 6-8

C FUNDAMENTAL 9-23

DATA INPUT AND OUTPUT 24 - 32

EXERCISES 33 - 39

OVERVIEW OF C LANGUAGE



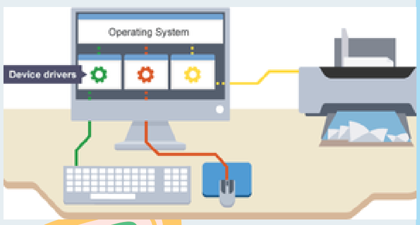
C LANGUAGE is a structured programming language developed by Dennis Ritchie in 1973

Features of C language

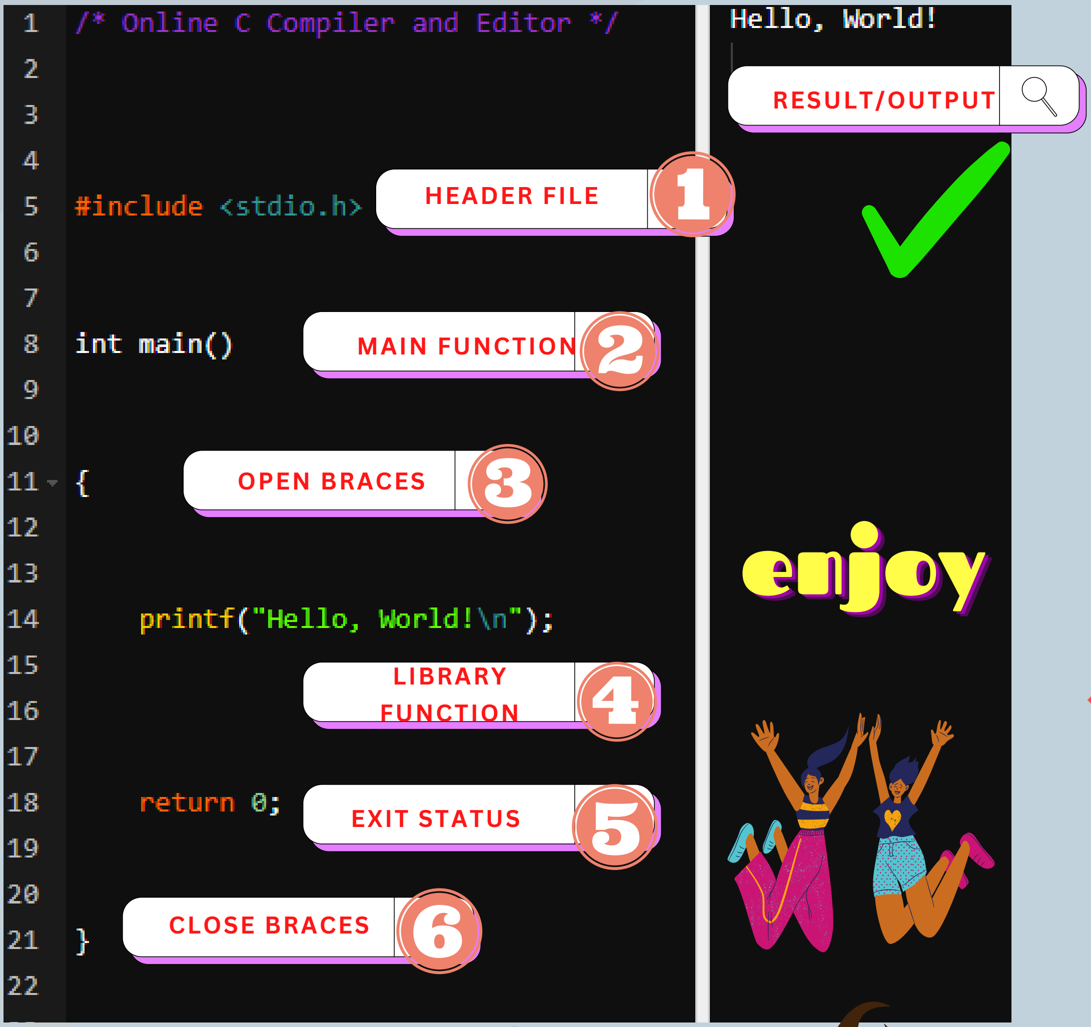
1. Fast & Efficient
2. Portable
3. Function Rich Library
4. Modularity
5. Easy to Extend
6. Variety data types

Advantages of C language over other programming languages

1. Easy to learn
2. Memory Management
3. Produces efficient programs
4. Powerful programming language
5. Structured programming language



LETS SEE HOW TO WRITE A SIMPLE AND MOST BASIC C PROGRAMMING.



```c
/* Online C Compiler and Editor */


#include <stdio.h>


int main()


{


    printf("Hello, World!\n");



    return 0;


}

```

```
Hello, World!
```

LETS KNOWING ME.
I'M C LANGUAGE

SYMBOL IN C: ;
semicolon

LETS SEE HOW TO WRITE A SIMPLE AND MOST BASIC C PROGRAMMING.

1

WHAT IS HEADER FILE IN C?

A Header file is a Standard header files are provided with each compiler, and covers a range of areas like string handling, mathematical functions, data conversion, printing and reading of variables.

2

WHAT IS Pre-Processor?

1. #include is the first word of any C program.
2. It is also known as a pre-processor.
3. #include<stdio.h>, it is to inform the compiler to include the stdio.h header file to the program before executing it.

3

WHAT IS A main ( )FUNCTION

main( ) function in a C program will be executed.

4

WHAT IS THE CURLY BRACES?

signify the start and end of a series of statements

5

HOW TO COMPILE & RUN C?

To compile and run a C language program, you need a C compiler.

7

SYMBOL IN C

\ BACKSLASH

/ SLASH

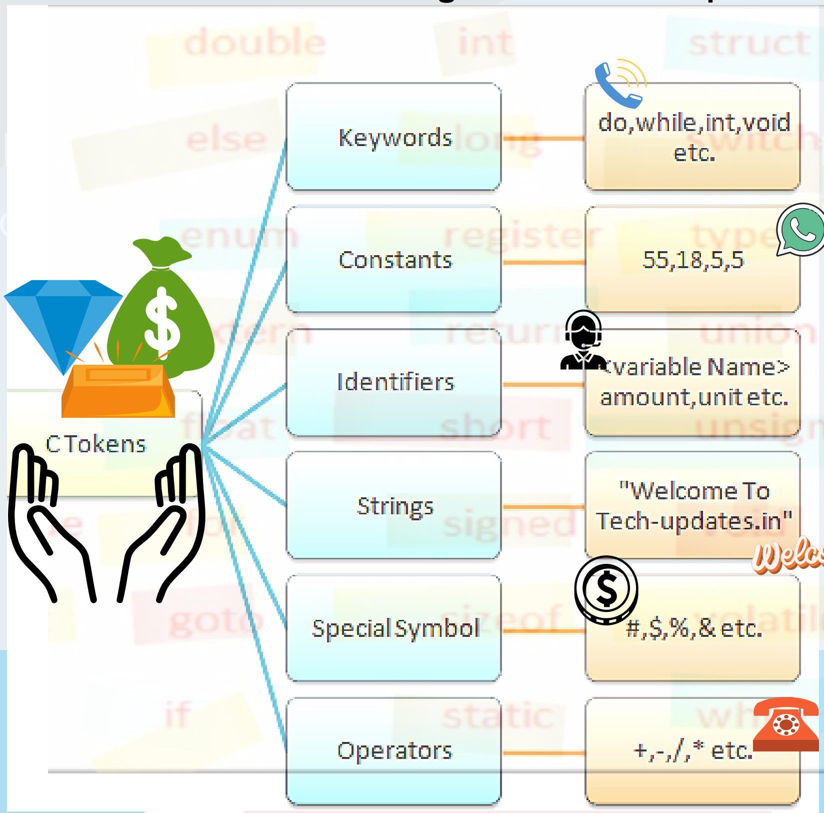
/ SLASH

C FUNDAMENTAL

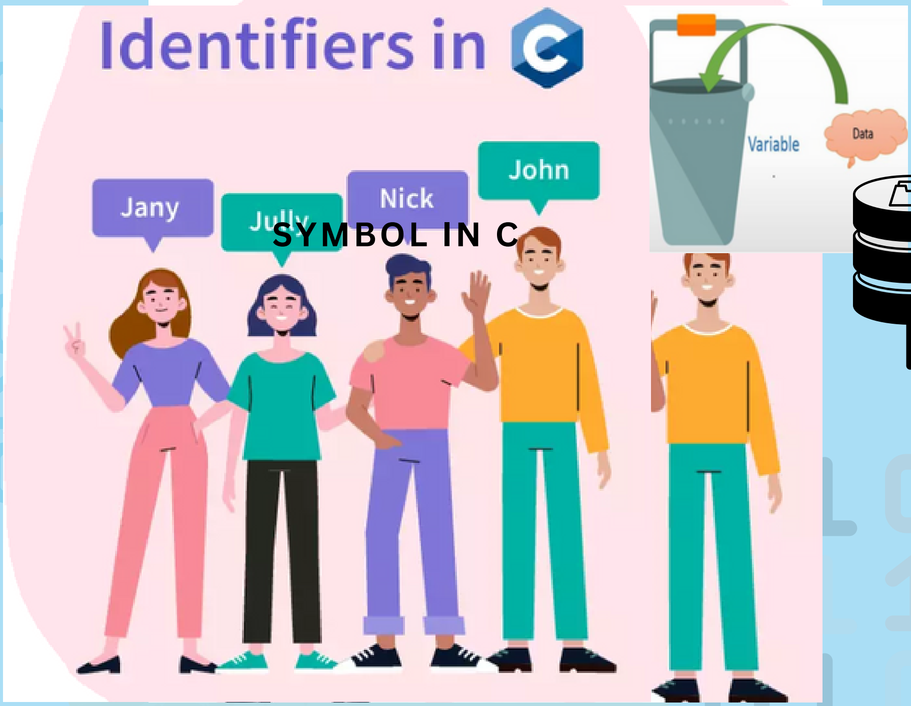
8

TOKEN C LANGUAGE

Tokens are the smallest elements of a program, which are meaningful to the compiler.



Keywords	do,while,int,void etc.
Constants	55,18,5,5
Identifiers	<variable Name> amount,unit etc.
Strings	"Welcome To Tech-updates.in"
Special Symbol	#,$,%,& etc.
Operators	+,-,/,* etc.



SYMBOL IN C

@

SYMBOL IN C

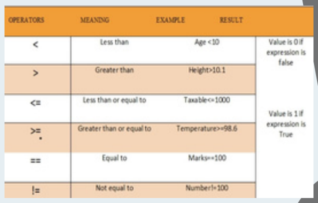
[ ]brackets @at sign

OPERATOR IN C LANGUAGE

Operators are used to perform operations on variables and values.

Relational operators

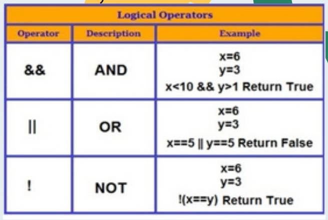
The following table shows all relation operators supported by C.



OPERATORS	MEANING	EXAMPLE	RESULT
<	Less than	Age <10	Value is 0 if expression is false
>	Greater than	Height>10.1	
<=	Less than or equal to	Taxable<=1000	Value is 1 if expression is True
>=	Greater than or equal to	Temperature>=98.6	
==	Equal to	Marks==100	
!=	Not equal to	Number!=100	

Logical operators

C language supports following 3 logical operators. Suppose a = 1 and b = 0,



Logical Operators		
Operator	Description	Example

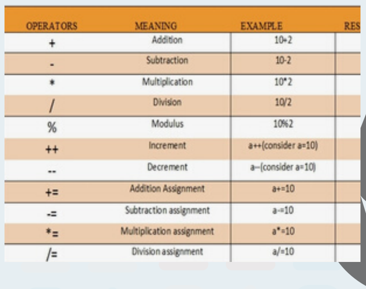
&&	AND	x=6
		y=3
		x<10 && y>1 Return True

||	OR	x=6
		y=3
		x==5 || y==5 Return False

!	NOT	x=6
		y=3
		!(x==y) Return True

Arithmetic operators

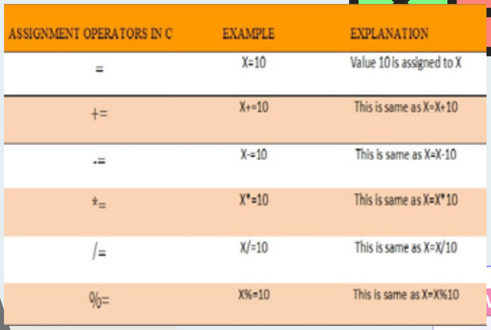
C supports all the basic arithmetic operators. The following table shows all the basic arithmetic operators.



OPERATORS	MEANING	EXAMPLE	RES
+	Addition	10+2	
-	Subtraction	10-2	
*	Multiplication	10*2	
/	Division	10/2	
%	Modulus	10%2	
++	Increment	a++(consider a=10)	
--	Decrement	a--(consider a=10)	
+=	Addition Assignment	a+=10	
-=	Subtraction assignment	a-=10	
*=	Multiplication assignment	a*=10	
/=	Division assignment	a/=10	

Assignment operator

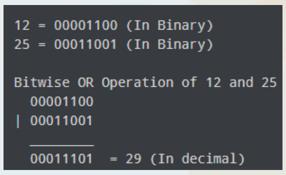
Assignment operators supported by C language are as follows.



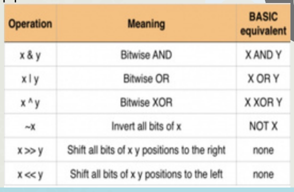
ASSIGNMENT OPERATORS IN C	EXAMPLE	EXPLANATION
=	X=10	Value 10 is assigned to X
+=	X+=10	This is same as X=X+10
-=	X-=10	This is same as X=X-10
*=	X*=10	This is same as X=X*10
/=	X/=10	This is same as X=X/10
%=	X%=10	This is same as X=X%10

Bitwise operators

Bitwise operators perform manipulations of data at bit level. These operators also perform shifting of bits from right to left. Bitwise operators are not applied to float or double.



```
12 = 00001100 (In Binary)
25 = 00011001 (In Binary)

Bitwise OR Operation of 12 and 25
  00001100
| 00011001
  ________
  00011101  = 29 (In decimal)
```



Operation	Meaning	BASIC equivalent
x & y	Bitwise AND	X AND Y
x | y	Bitwise OR	X OR Y
x ^ y	Bitwise XOR	X XOR Y
~x	Invert all bits of x	NOT X
x >> y	Shift all bits of x y positions to the right	none
x << y	Shift all bits of x y positions to the left	none

SYMBOL IN C

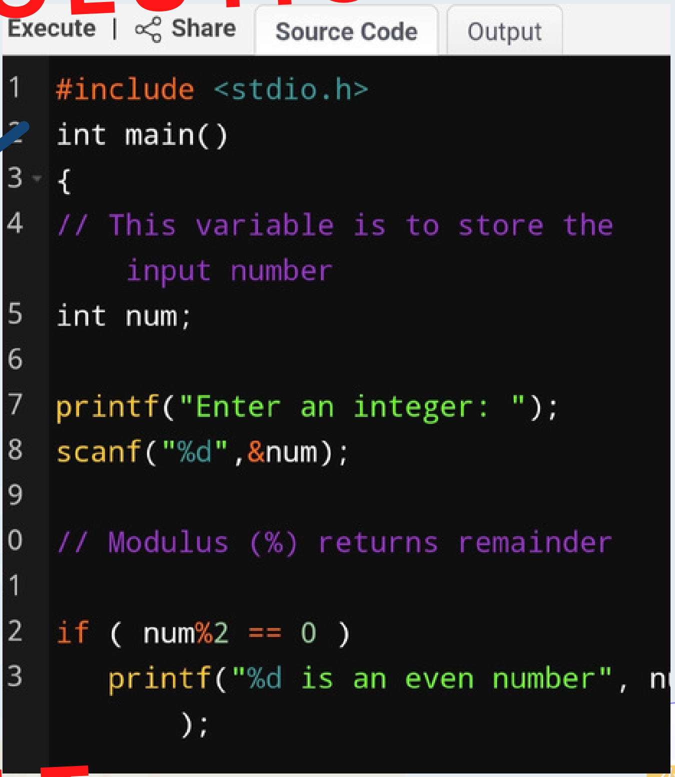
*	asterisk
●	dot

REFER EXERCISE BELOW WITH OPERATOR IN C

PROBLEM 2:

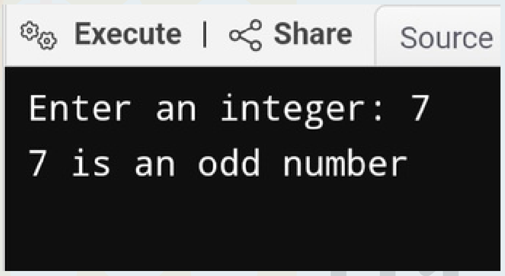
How do you check two numbers is even or odd using relational operator?

SOLUTION



```
#include <stdio.h>
int main()
{
// This variable is to store the
    input number
int num;

printf("Enter an integer: ");
scanf("%d",&num);

// Modulus (%) returns remainder

if ( num%2 == 0 )
   printf("%d is an even number", n
      );
```

RESULT



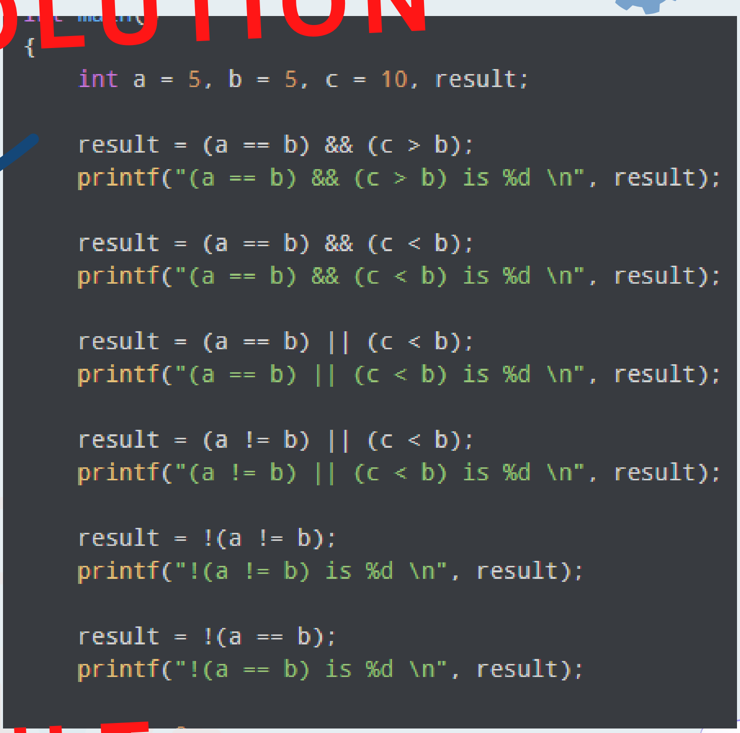
```
Enter an integer: 7
7 is an odd number
```

REFER EXERCISE BELOW WITH OPERATOR IN C

PROBLEM 3:

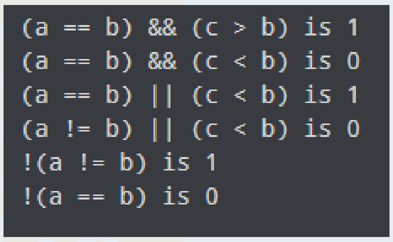
How do you check to compare three numbers using logical operator?

SOLUTION



```
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);
```

RESULT



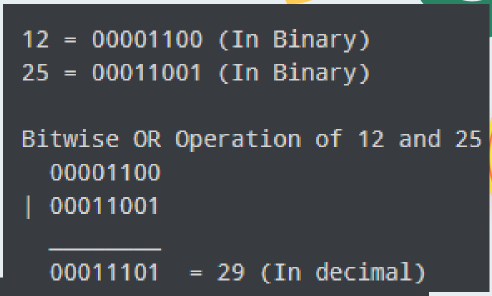
```
(a == b) && (c > b) is 1
(a == b) && (c < b) is 0
(a == b) || (c < b) is 1
(a != b) || (c < b) is 0
!(a != b) is 1
!(a == b) is 0
```

REFER EXERCISE BELOW WITH OPERATOR IN C

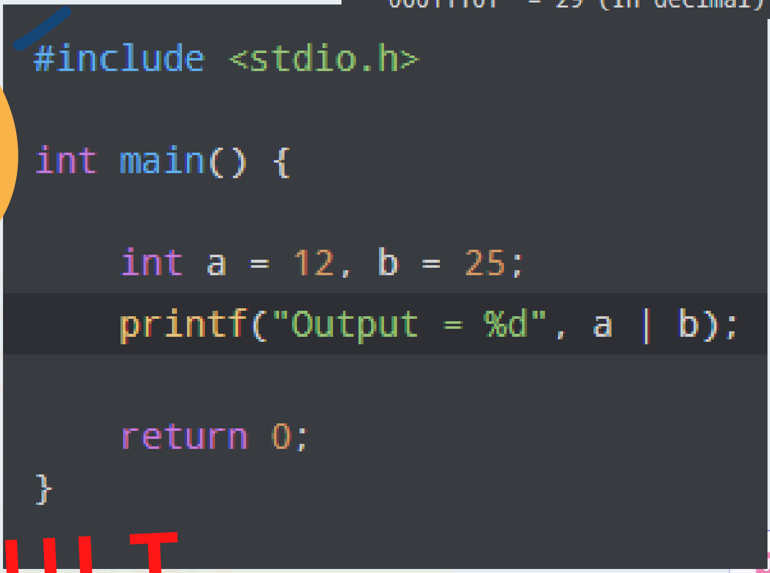
PROBLEM 4:

Apply bitwise OR operation to calculate of two integers 12 and 25.

SOLUTION

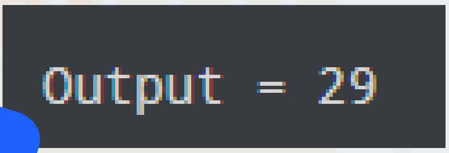


```
12 = 00001100 (In Binary)
25 = 00011001 (In Binary)

Bitwise OR Operation of 12 and 25
  00001100
| 00011001
  ________
  00011101  = 29 (In decimal)
```



```c
#include <stdio.h>

int main() {

    int a = 12, b = 25;
    printf("Output = %d", a | b);

    return 0;
}
```

RESULT

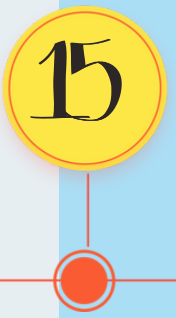


```
Output = 29
```

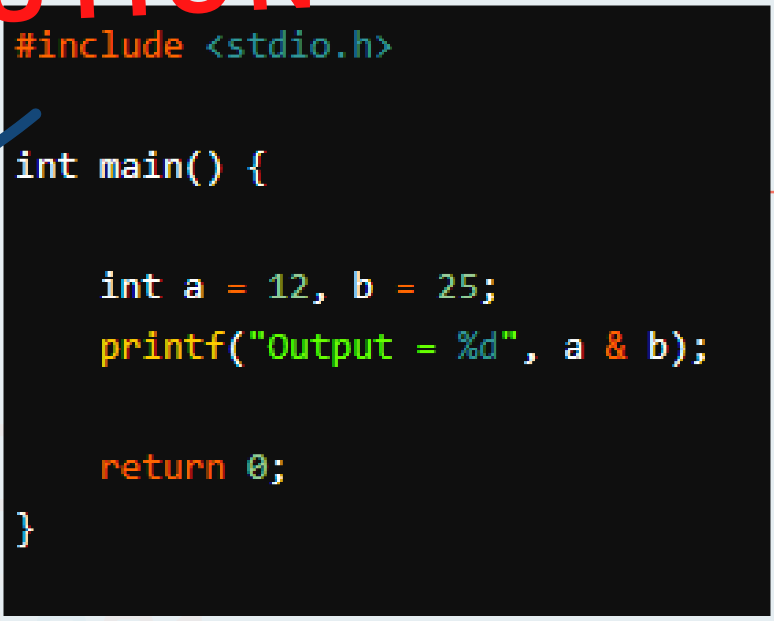
REFER EXERCISE BELOW WITH OPERATOR IN C

PROBLEM 5:

Apply bitwise OR operation to calculate of two integers 12 and 25.

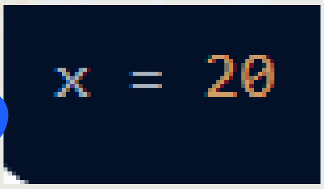


SOLUTION



```c
#include <stdio.h>

int main() {

    int a = 12, b = 25;
    printf("Output = %d", a & b);

    return 0;
}
```

RESULT



```
x = 20
```

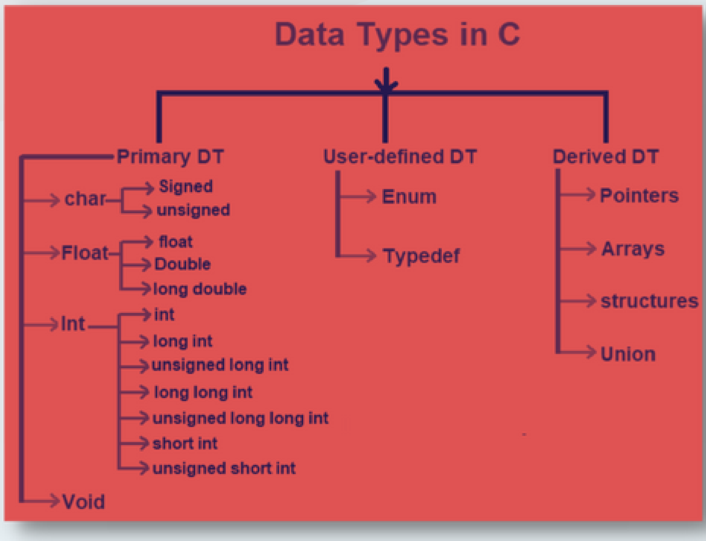
Next.......

LETS

LEARN

TOGETHER

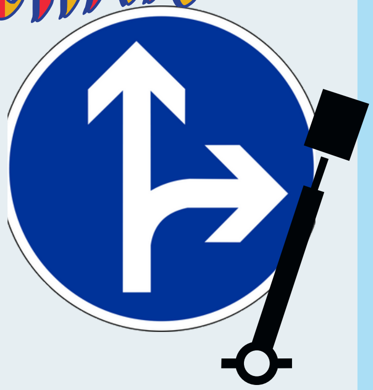
DATA TYPES IN C LANGUAGES

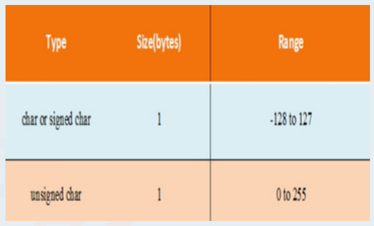


CHARACTER TYPE

Character types are used to store characters value.

CHAR

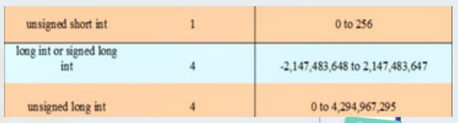




Type	Size(bytes)	Range
char or signed char	1	-128 to 127
unsigned char	1	0 to 255

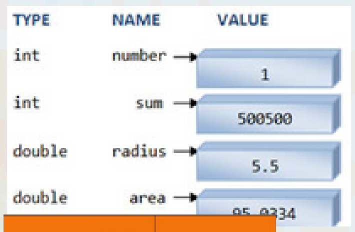
Size and range of Integer type on 16-bit machine

INTEGER TYPE



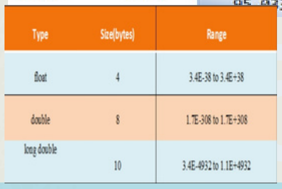
unsigned short int	1	0 to 256
long int or signed long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295

Floating types are used to store real numbers.



TYPE	NAME	VALUE
int	number	1
int	sum	500500
double	radius	5.5
double	area	95.0334

Size and range of Integer type on 16-bit machine



Type	Size(bytes)	Range
float	4	3.4E-38 to 3.4E+38
double	8	1.7E-308 to 1.7E+308
long double	10	3.4E-4932 to 1.1E+4932

SYMBOL IN C

" "
quotation mark

VARIABLE IN C LANGUAGE

When we want to store any information(data) on our computer/laptop, we store it in the computer's memory space.

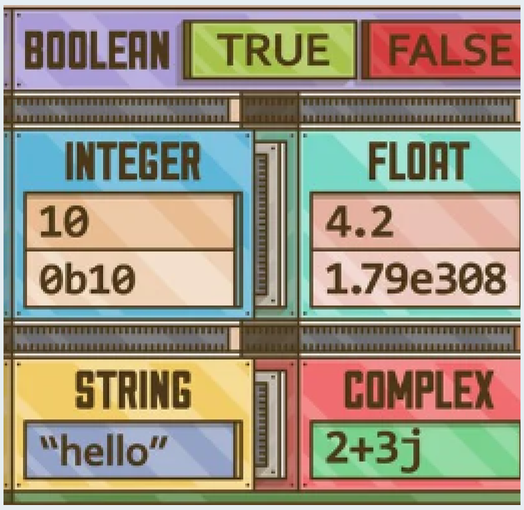
The naming of an address is known as variable. Variable is the name of memory location. Unlike constant, variables are changeable, we can change value of a variable during execution of a program.

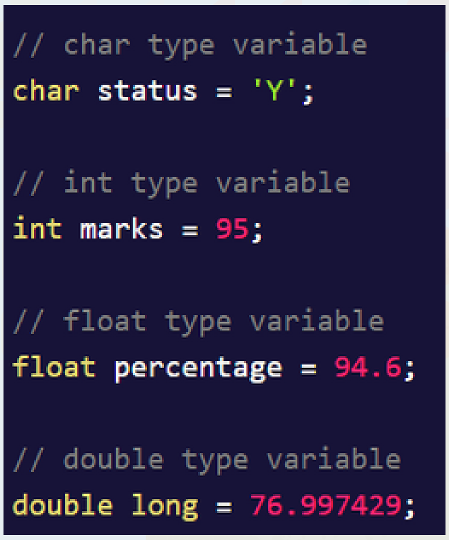
Similarly, in C language, when we want to use some data value in our program, we can store it in a memory space and name the memory space so that it becomes easier to access it.

A variable in C language must be given a type, which defines what type of data the variable will hold.

It can be:

- char: Can hold/store a character in it.
- int: Used to hold an integer.
- float: Used to hold a float value.





```
// char type variable
char status = 'Y';

// int type variable
int marks = 95;

// float type variable
float percentage = 94.6;

// double type variable
double long = 76.997429;
```

RULES TO NAME A VARIABLES

1. 1.Variable name must not start with a digit
2. 2.Variable name can consist of alphabets, digits and special symbols like underscore _.
3. Blank or spaces are not allowed in variable name

Keywords are not allowed as variable name.

5.Upper and lower case names are treated as different, as C is case-sensitive, so it is suggested to keep the variable names in lower case.

SYMBOL IN C

apostrophe `

_ underscore

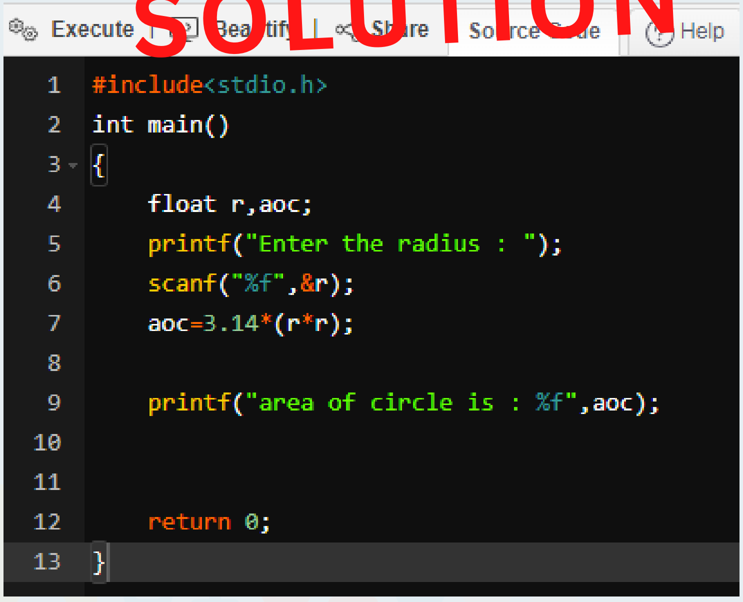
~ tilde

REFER EXERCISE BELOW WITH DATA TYPES IN C

PROBLEM 1:

Find Area of Circle.

SOLUTION

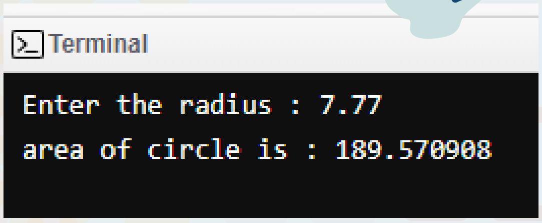


```c
#include<stdio.h>
int main()
{
    float r,aoc;
    printf("Enter the radius : ");
    scanf("%f",&r);
    aoc=3.14*(r*r);

    printf("area of circle is : %f",aoc);


    return 0;
}
```

Algorithm

1. Program Start
2. Declare Variables
3. Input Radius From User
4. Calculate the Area of the Circle
5. Display Result
6. Program End

RESULT



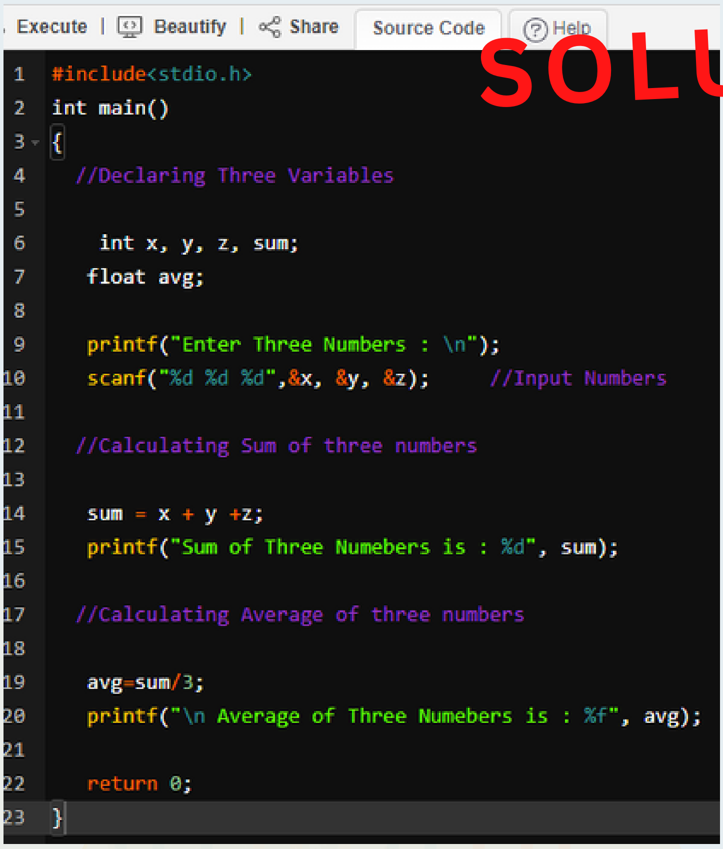
```
Enter the radius : 7.77
area of circle is : 189.570908
```

REFER EXERCISE BELOW WITH DATA TYPES IN C

PROBLEM 2:

Find the Sum and Average of Three Numbers

SOLUTION

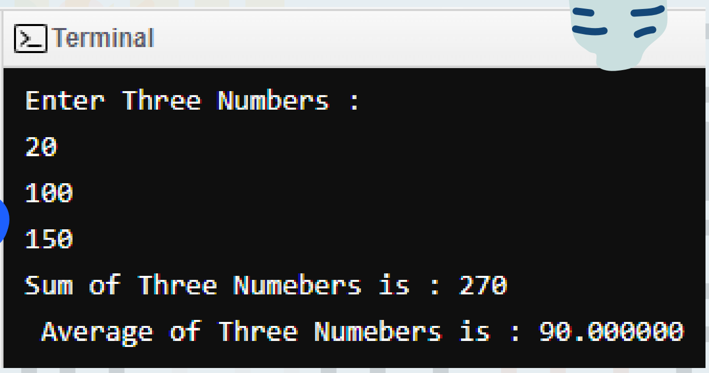


```c
#include<stdio.h>
int main()
{
  //Declaring Three Variables

    int x, y, z, sum;
   float avg;

   printf("Enter Three Numbers : \n");
   scanf("%d %d %d",&x, &y, &z);      //Input Numbers

  //Calculating Sum of three numbers

   sum = x + y +z;
   printf("Sum of Three Numebers is : %d", sum);

  //Calculating Average of three numbers

   avg=sum/3;
   printf("\n Average of Three Numebers is : %f", avg);

   return 0;
}
```

Algorithm

1. Program Start
2. Declaring Variables
3. Input Three Numbers from User
4. Calculating Sum of Three Numbers (sum = x + y + z)
5. Displaying the Sum of Three Numbers
6. Calculating Average of Three Numbers (average = sum/count)
7. Displaying the Average of Three Numbers
8. Program End

RESULT



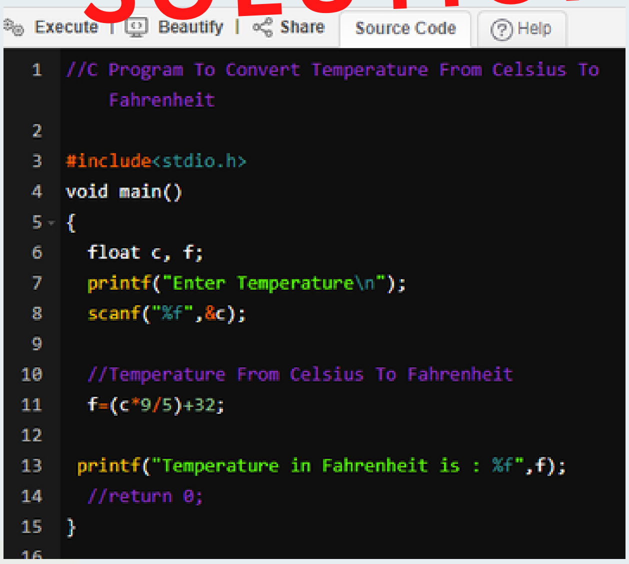
```
Enter Three Numbers :
20
100
150
Sum of Three Numebers is : 270
 Average of Three Numebers is : 90.000000
```

REFER EXERCISES BELOW WITH DATA TYPES IN C

PROBLEM 3:

To Convert Temperature From Celsius To Fahrenheit

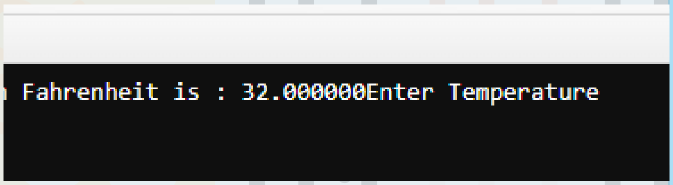
SOLUTION



```c
//C Program To Convert Temperature From Celsius To
    Fahrenheit

#include<stdio.h>
void main()
{
  float c, f;
  printf("Enter Temperature\n");
  scanf("%f",&c);

  //Temperature From Celsius To Fahrenheit
  f=(c*9/5)+32;

 printf("Temperature in Fahrenheit is : %f",f);
  //return 0;
}
```

1. Algorithm-:
2. Program Start
3. Declaration of variable
4. Enter temperature in Celsius
5. Fahrenheit conversion formula
6. Print result
7. Program End

RESULT



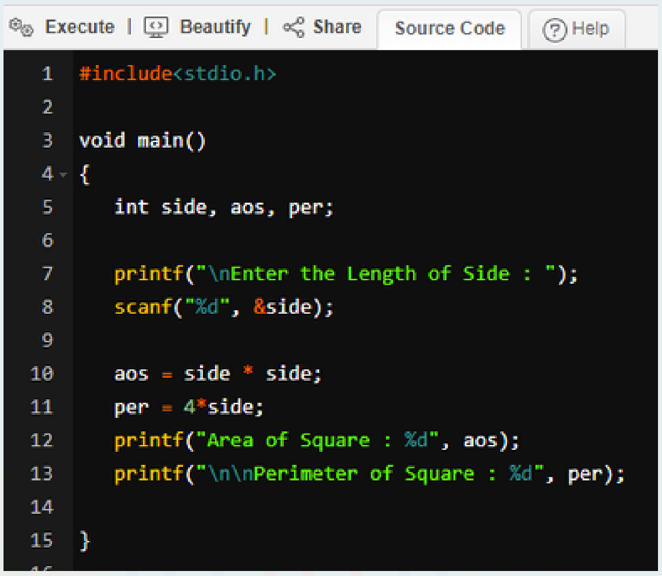
```
 Fahrenheit is : 32.000000Enter Temperature
```

REFER EXERCISES BELOW WITH DATA TYPES IN C

PROBLEM 4:

Find Area and Perimeter of Square

SOLUTION

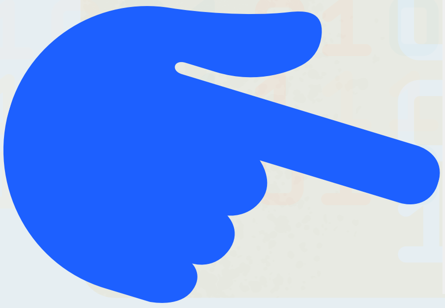


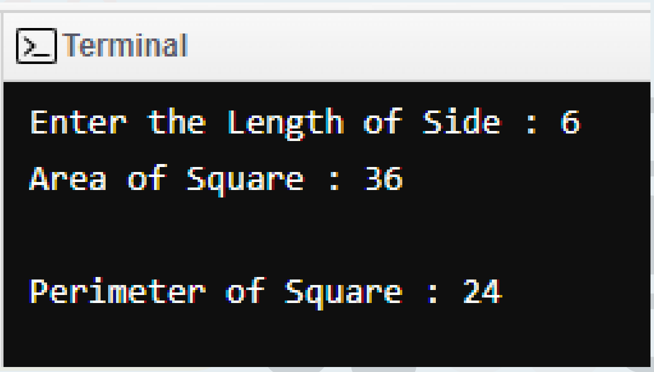
```c
#include<stdio.h>

void main()
{
    int side, aos, per;

    printf("\nEnter the Length of Side : ");
    scanf("%d", &side);

    aos = side * side;
    per = 4*side;
    printf("Area of Square : %d", aos);
    printf("\n\nPerimeter of Square : %d", per);

}
```

Algorithm

1. Program Start
2. Declare Variables
3. Input side from the User
4. Calculating Area of square
5. Calculating Perimeter of square
6. Display Result
7. Program End

RESULT





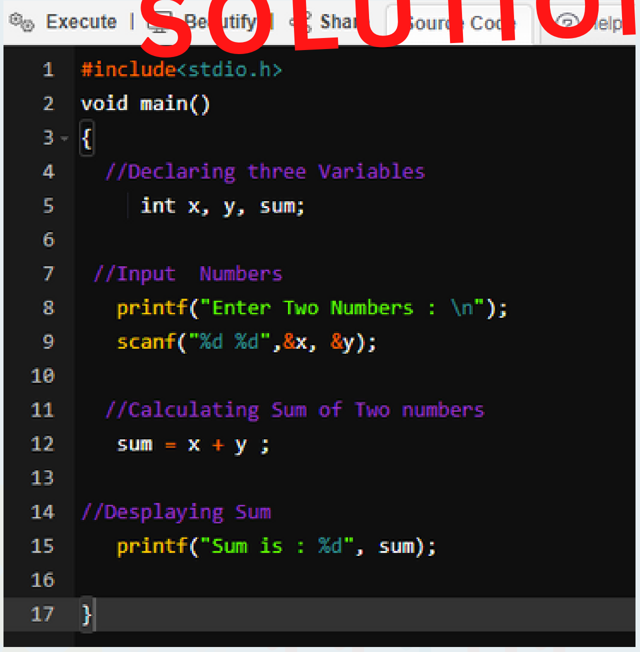
```
Enter the Length of Side : 6
Area of Square : 36

Perimeter of Square : 24
```

REFER EXERCISES BELOW WITH DATA TYPES IN C

PROBLEM 5:

Formula to calculate the Addition of two numbers

SOLUTION

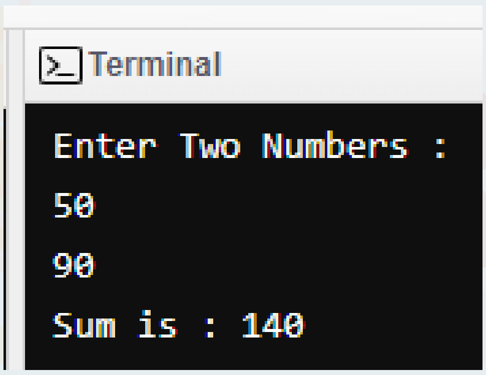


```c
#include<stdio.h>
void main()
{
  //Declaring three Variables
     int x, y, sum;

 //Input  Numbers
   printf("Enter Two Numbers : \n");
   scanf("%d %d",&x, &y);

  //Calculating Sum of Two numbers
   sum = x + y ;

//Desplaying Sum
   printf("Sum is : %d", sum);

}
```

Algorithm

1. Program Start
2. Declaring Variables (x,y,sum)
3. Input Two Numbers
4. Calculating Sum of Two Numbers
5. Displaying the Sum of Two Numbers
6. Program End

RESULT



```
Enter Two Numbers :
50
90
Sum is : 140
```

DATA INPUT OUTPUT

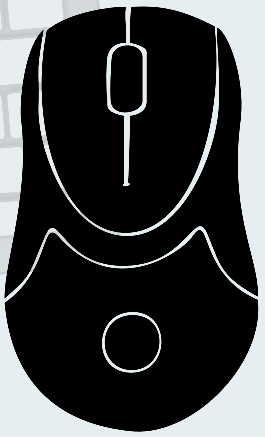
LETS

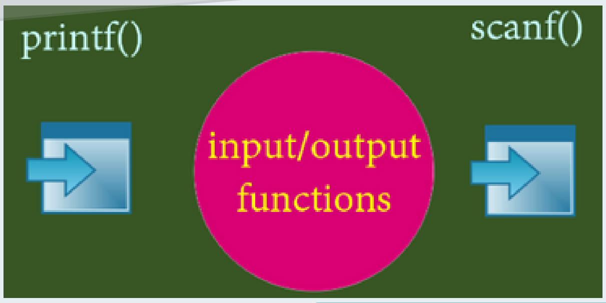
LEARN

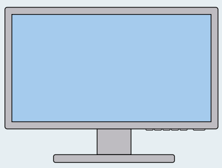
TOGETHER

DATA INPUT OUTPUT

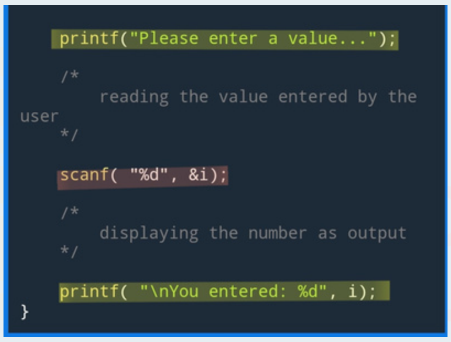
Input means to provide the program with some data to be used in the program



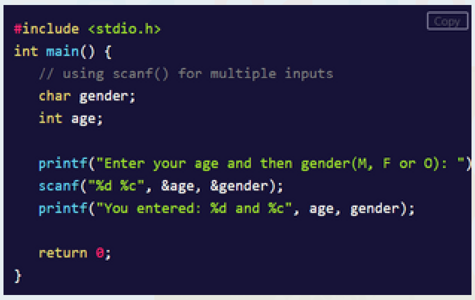




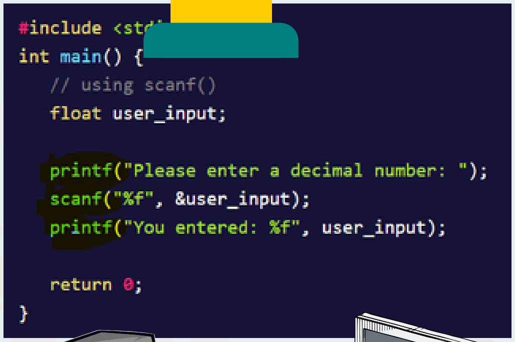
Output means to display data on screen or write the data to a printer or a file.



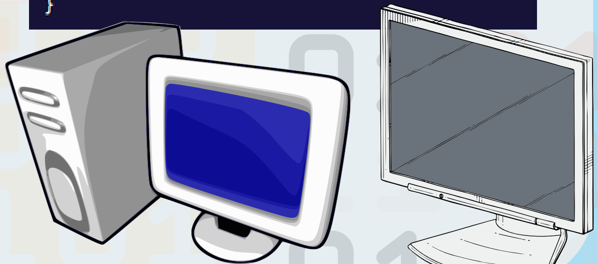
```
    printf("Please enter a value...");

    /*
        reading the value entered by the
user
    */

    scanf( "%d", &i);

    /*
        displaying the number as output
    */

    printf( "\nYou entered: %d", i);
}
```



```
#include <stdio.h>
int main() {
   // using scanf() for multiple inputs
   char gender;
   int age;

   printf("Enter your age and then gender(M, F or O): ")
   scanf("%d %c", &age, &gender);
   printf("You entered: %d and %c", age, gender);

   return 0;
}
```



```
#include <std
int main() {
   // using scanf()
   float user_input;

   printf("Please enter a decimal number: ");
   scanf("%f", &user_input);
   printf("You entered: %f", user_input);

   return 0;
}
```



SYMBOL IN C

! exclamation mark

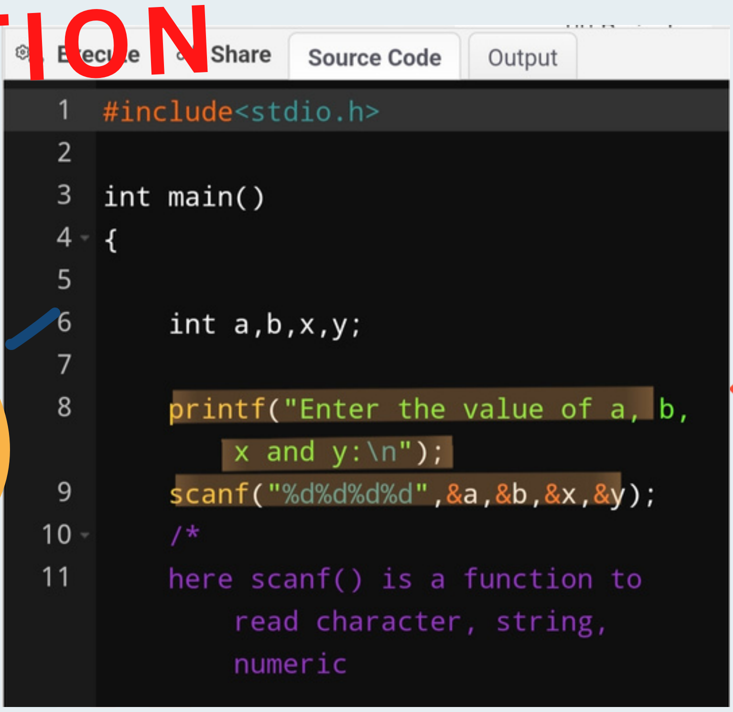
REFER EXERCISES BELOW WITH INPUT OUTPUT IN C

PROBLEM 1:

Take four integer variables a, b, x and y. Scan the values of the variables from user using scanf() function. Now print the output of the following equation:

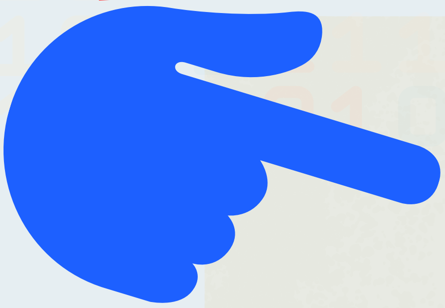
(a*b) + (x*y)

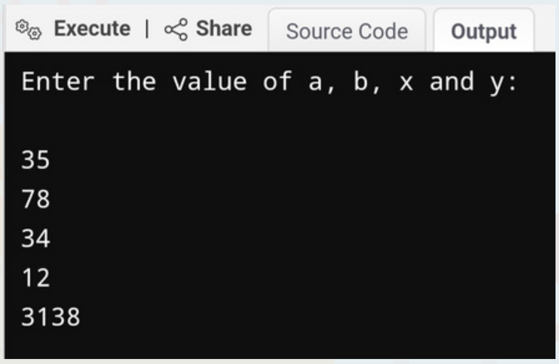
SOLUTION



```c
#include<stdio.h>

int main()
{

    int a,b,x,y;

    printf("Enter the value of a, b, x and y:\n");
    scanf("%d%d%d%d",&a,&b,&x,&y);
    /*
    here scanf() is a function to read character, string, numeric
```

RESULT





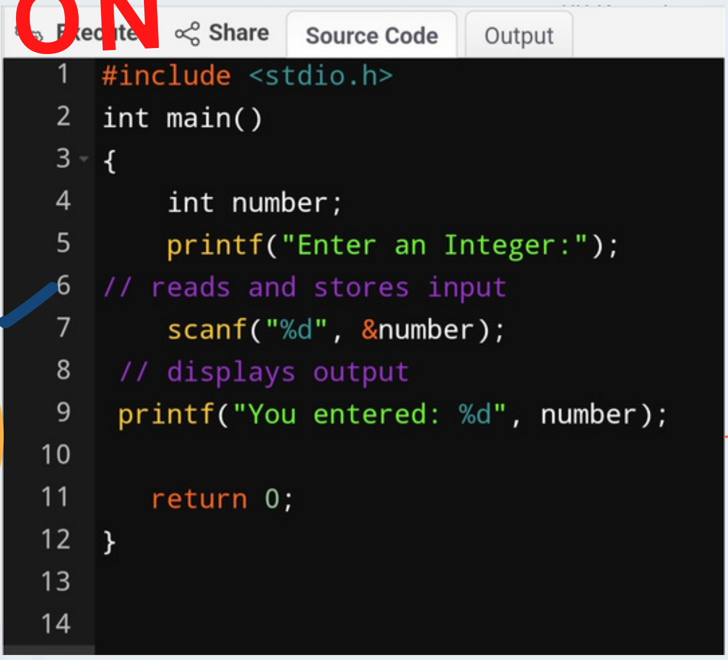
```
Enter the value of a, b, x and y:

35
78
34
12
3138
```

REFER EXERCISES BELOW WITH INPUT OUTPUT IN C

PROBLEM 2:

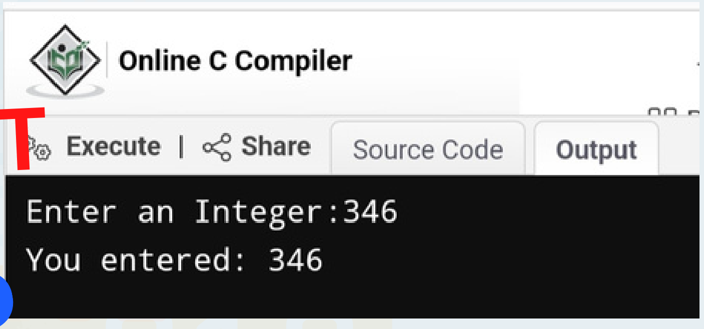
Print an integers that you have entered

SOLUTION



```c
#include <stdio.h>
int main()
{
     int number;
     printf("Enter an Integer:");
// reads and stores input
     scanf("%d", &number);
 // displays output
 printf("You entered: %d", number);

    return 0;
}
```

RESULT



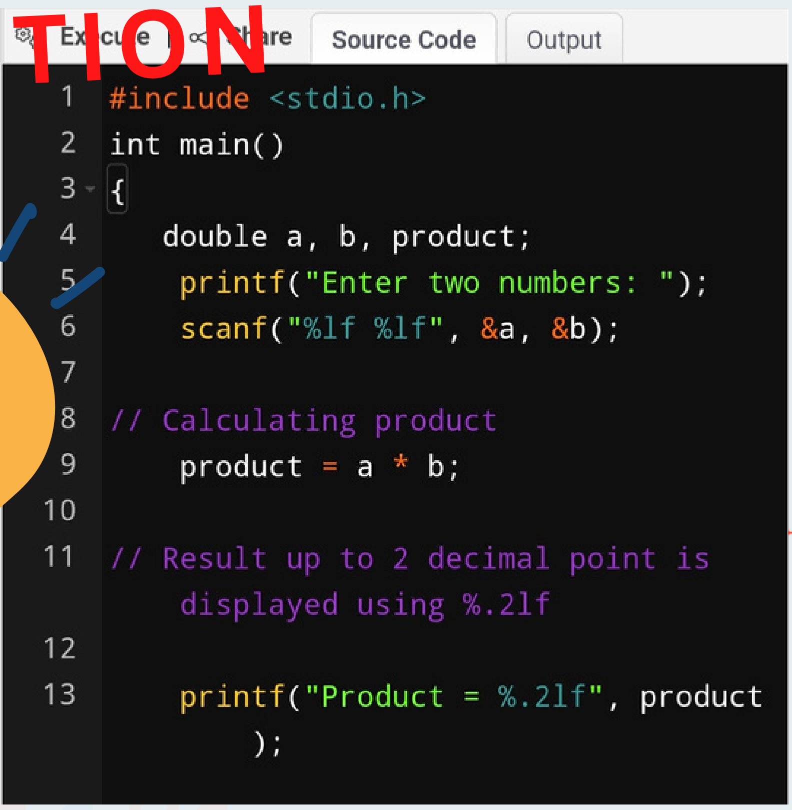
```
Enter an Integer:346
You entered: 346
```

REFER EXERCISES BELOW WITH INPUT OUTPUT IN C

PROBLEM 3:

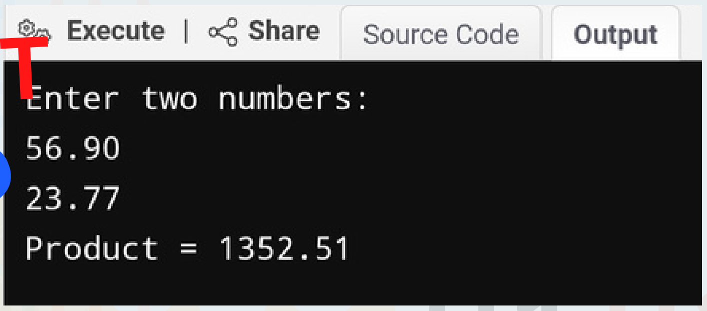
Multiply Two Floating-Point Numbers

SOLUTION



```c
#include <stdio.h>
int main()
{
    double a, b, product;
     printf("Enter two numbers: ");
     scanf("%lf %lf", &a, &b);

// Calculating product
     product = a * b;

// Result up to 2 decimal point is
     displayed using %.2lf

     printf("Product = %.2lf", product
         );
```

RESULT



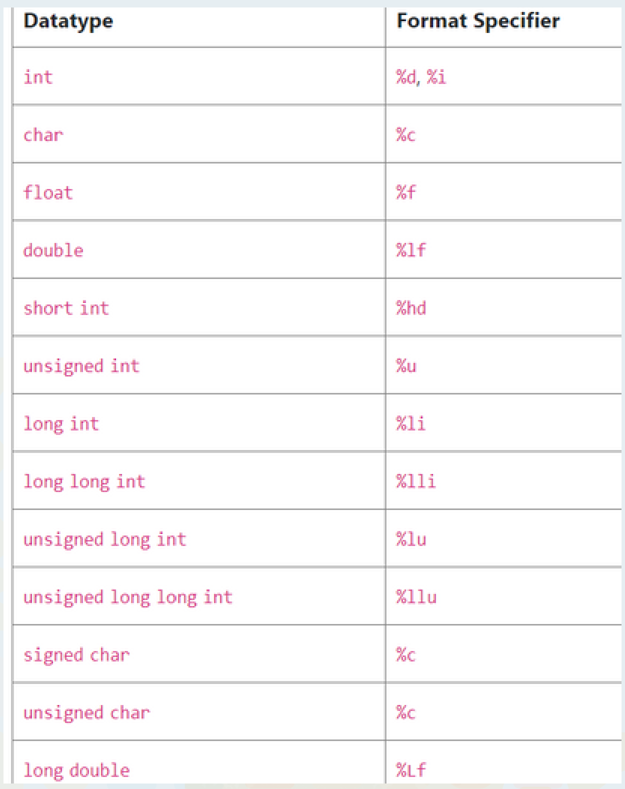
```
Enter two numbers:
56.90
23.77
Product = 1352.51
```

STRING AND CHARACTER

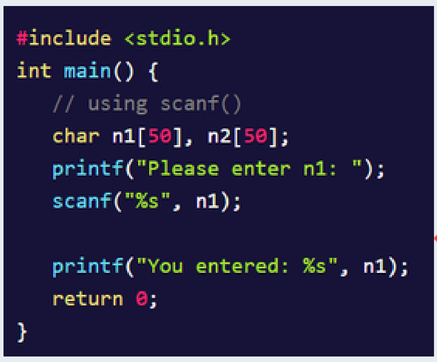
braces

String Input and Output
Input function scanf() can be used with %s format specifier to read a string input from the terminal.

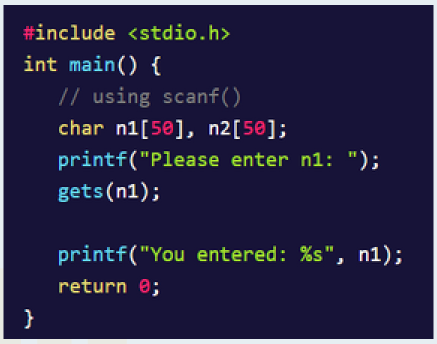
String is a sequence of characters that is treated as a single data item and terminated by null character '\0'.



Datatype	Format Specifier
int	%d, %i
char	%c
float	%f
double	%lf
short int	%hd
unsigned int	%u
long int	%li
long long int	%lli
unsigned long int	%lu
unsigned long long int	%llu
signed char	%c
unsigned char	%c
long double	%Lf



```c
#include <stdio.h>
int main() {
   // using scanf()
   char n1[50], n2[50];
   printf("Please enter n1: ");
   scanf("%s", n1);

   printf("You entered: %s", n1);
   return 0;
}
```



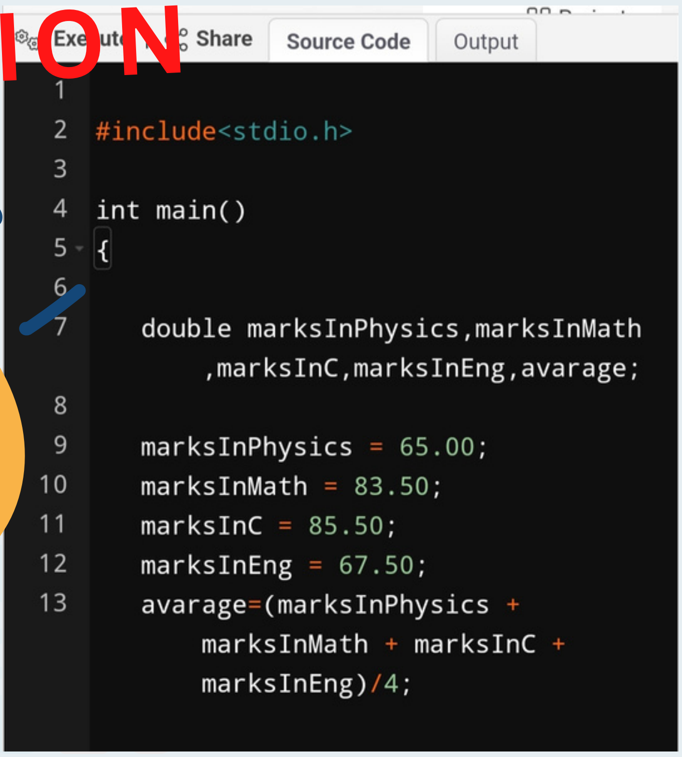
```c
#include <stdio.h>
int main() {
   // using scanf()
   char n1[50], n2[50];
   printf("Please enter n1: ");
   gets(n1);

   printf("You entered: %s", n1);
   return 0;
}
```

REFER EXERCISES BELOW WITH STRING & CHARACTER IN C

PROBLEM 1:

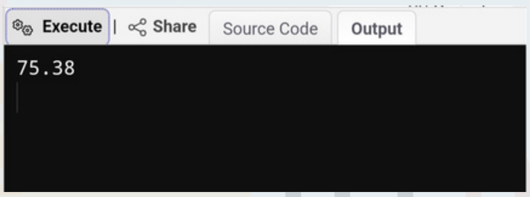
Ali got 65.00 on physics, 83.50 on mathematics, 85.75 on C programming and 67.50 on English. Now write a program to calculate the average of his marks on 4 subjects and print it up to 2 digit after the decimal point. [The result should look like: XX.XX]

SOLUTION



```c

#include<stdio.h>

int main()
{

    double marksInPhysics,marksInMath
        ,marksInC,marksInEng,avarage;

    marksInPhysics = 65.00;
    marksInMath = 83.50;
    marksInC = 85.50;
    marksInEng = 67.50;
    avarage=(marksInPhysics +
        marksInMath + marksInC +
        marksInEng)/4;
```

RESULT



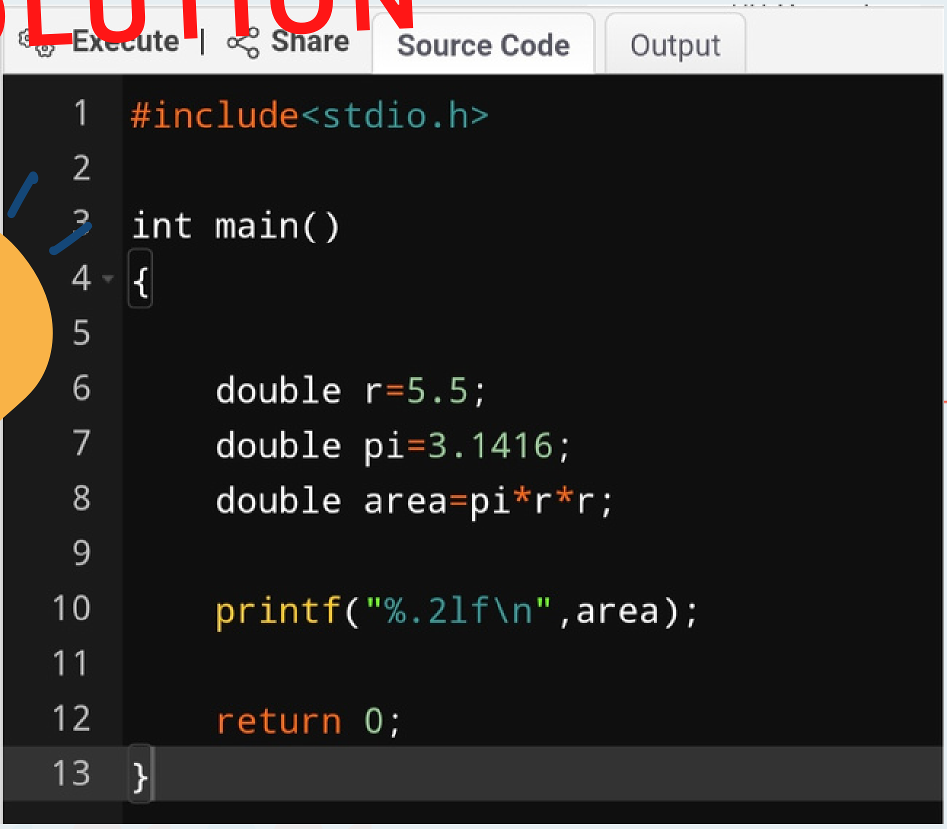
```
75.38
```

REFER EXERCISES BELOW WITH STRING & CHARACTER IN C

PROBLEM 2:

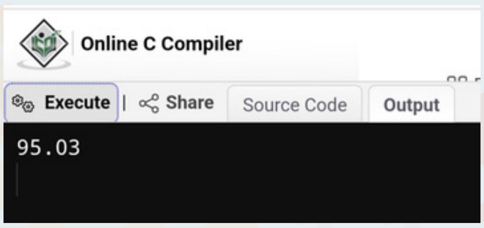
You are given the radius of a circle, r = 5.5. We know that, pi=3.1416. Now write a program to calculate the area of the given circle and print it up to 2 digit after the decimal point. [The result should look like: XX.XX]

SOLUTION



```c
#include<stdio.h>

int main()
{

    double r=5.5;
    double pi=3.1416;
    double area=pi*r*r;

    printf("%.2lf\n",area);

    return 0;
}
```

RESULT



```
95.03
```

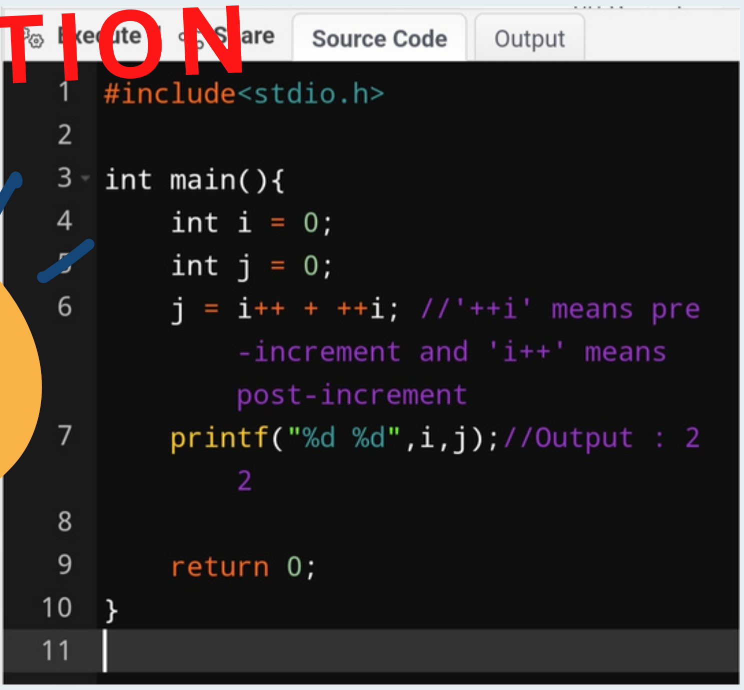
REFER EXERCISES BELOW WITH STRING & CHARACTER IN C

PROBLEM 3:

Take two integer variables i = 0 and j = 0. Now write the output of the following program without running the code.

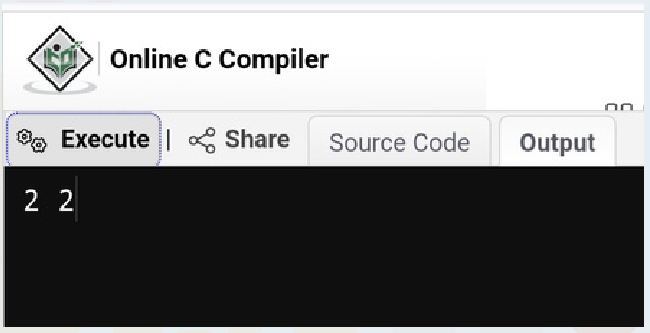
```
int main()
int i = 0;
int j = 0;
j = i++ + ++i;
printf("%d %d",i,j);
}
```

SOLUTION



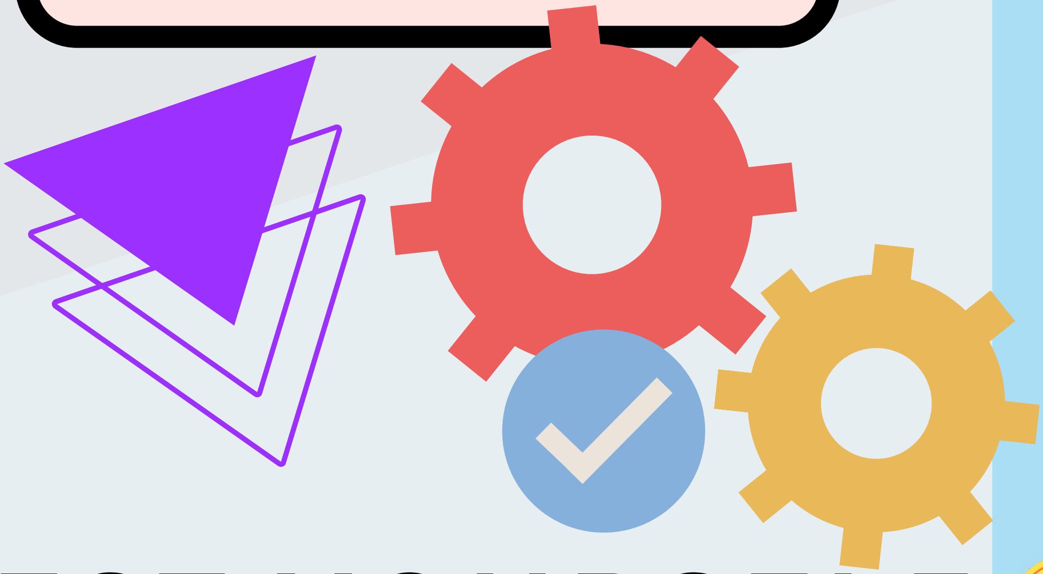
```
#include<stdio.h>

int main(){
    int i = 0;
    int j = 0;
    j = i++ + ++i; //'++i' means pre
        -increment and 'i++' means
        post-increment
    printf("%d %d",i,j);//Output : 2
        2

    return 0;
}

```

RESULT

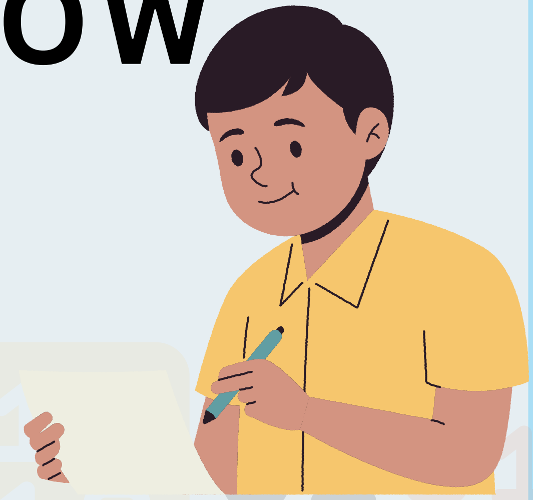


```
2 2
```

QUIZ TIME



TEST YOURSELF WITH EXERCISE BELOW



TEST YOURSELF WITH EXERCISE BELOW

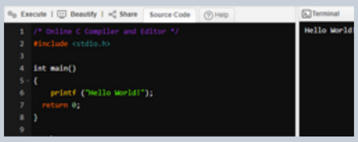
QUESTION 1

Insert the missing part of the code below to output "Hello World!".

```
int       () {
         ("Hello World!");
  return 0;
}
```

ANSWER 1

```
int main() {
  printf("Hello World!");
  return 0;
}
```



```
/* Online C Compiler and Editor */
#include <stdio.h>

int main()
{
    printf ("Hello World!");
  return 0;
}
```

```
Hello World!
```

TEST YOURSELF WITH EXERCISE BELOW

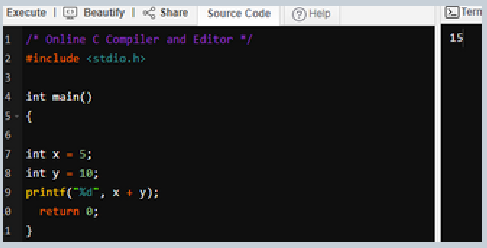
QUESTION 2

Display the sum of 5 + 10, using two variables: x and y.

```
          = ;
int y = 10;
printf("%d", x + y);
```

ANSWER 2

```
int x = 5;
int y = 10;
printf("%d", x + y);
```



```
/* Online C Compiler and Editor */
#include <stdio.h>

int main()
{

int x = 5;
int y = 10;
printf("%d", x + y);
  return 0;
}
```

```
15
```

TEST YOURSELF WITH EXERCISE BELOW

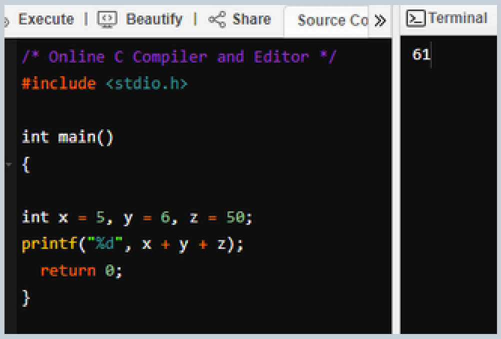
QUESTION 3

Fill in the missing parts to create three variables of the same type, using a comma-separated list:

```
     x = 5  y = 6  z = 50;
printf("%d", x + y + z);
```

ANSWER 3

```
int x = 5, y = 6, z = 50;
printf("%d", x + y + z);
```



```
/* Online C Compiler and Editor */
#include <stdio.h>

int main()
{

int x = 5, y = 6, z = 50;
printf("%d", x + y + z);
  return 0;
}
```

```
61
```

TEST YOURSELF WITH EXERCISE BELOW

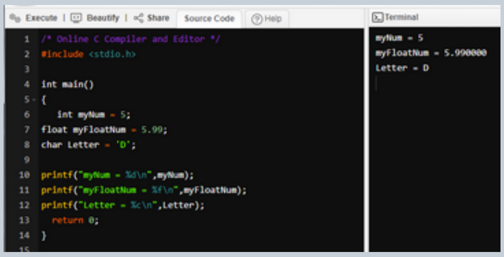
QUESTION 4

Add the correct data type for the following variables:

```
        myNum = 5;
          myFloatNum = 5.99;
         myLetter = 'D';
```

ANSWER 4

```
int myNum = 5;
float myFloatNum = 5.99;
char myLetter = 'D';
```



```
/* Online C Compiler and Editor */
#include <stdio.h>

int main()
{
   int myNum = 5;
float myFloatNum = 5.99;
char Letter = 'D';

printf("myNum = %d\n",myNum);
printf("myFloatNum = %f\n",myFloatNum);
printf("Letter = %c\n",Letter);
  return 0;
}
```

```
myNum = 5
myFloatNum = 5.990000
Letter = D
```

TEST YOURSELF WITH EXERCISE BELOW

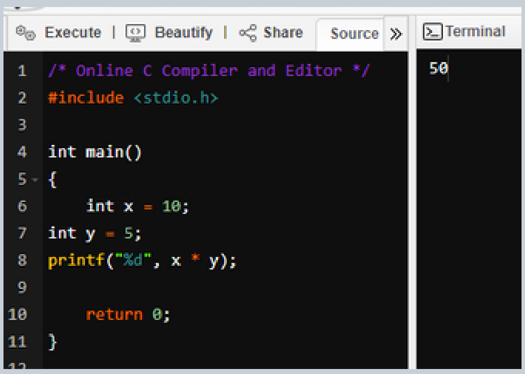
QUESTION 5

Fill in the blanks to multiply 10 with 5, and print the result.

```
int x = 10;
int y = 5;
printf("   ", x   y);
```

ANSWER 5

```
int x = 10;
int y = 5;
printf("%d", x * y);
```



```
/* Online C Compiler and Editor */
#include <stdio.h>

int main()
{
    int x = 10;
int y = 5;
printf("%d", x * y);

    return 0;
}
```

```
50
```

RECALL YOUR MEMORY

WRITE AND NAME THE SYMBOL BELOW.

! ●●●●●●●●●●●●●●

[ ] ●●●●● ●●●●●

? ●●●●● ●●●●●

{ } ●●●●● ●●●●●

~ ●●●●● ●●●●●

* ●●●●● ●●●●●

REFERENCES

1. Syamsul Halim bin Wahab (2009). Asas Pengaturcaraan C Bagi Beginner. Jilid 1, (55-80).
2. Norizan Mohamad (2003). C++ Programming :Exercise Book. Jilid 1, (30-230).
3. Brian W.Kernighan Dennis M.Ritchie. The C Programming Language. 2nd Edition.(90-123).
4. C Programming Absolute Beginner's Guide. 3rd Edition. (78-232).
5. Noor Hasrina Bakar (2019). Programming in C For Foundation. (60-200).
6. Learn C Programming – Programiz. https://www.programiz.com › c-programming.
7. C Tutorial – Tutorialspoint. https://www.tutorialspoint.com › cprogramming

e ISBN 978-967-0047-23-2

9 7 8 9 6 7 0 0 4 7 2 3 2