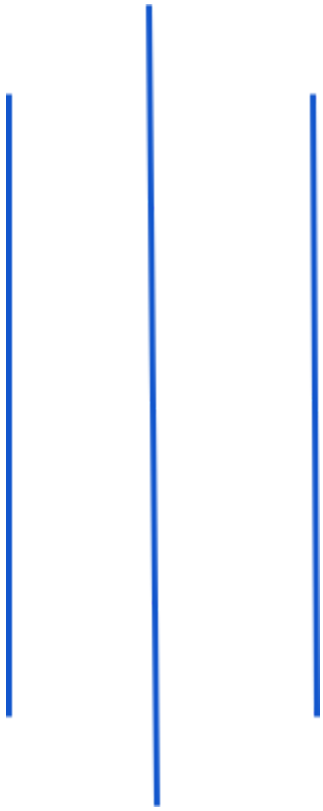
Tribhuvan University

Institute of Engineering

Thapathali Campus,Thapathali

LAB SHEET #1



Submitted by:

Name: Kishan Adhikari

Roll No. :THA077BCT021

Submitted to:

Department of Electronics and Computer Engineering

Date : 10th July 2021

Objective(s):

- To be familiar with syntax and structure of C-Programming
- To learn problem solving techniques using C-Programming
- To be familiar with different data types, Operators and Expressions in C.
- To make algorithms and flowchart for different problems.

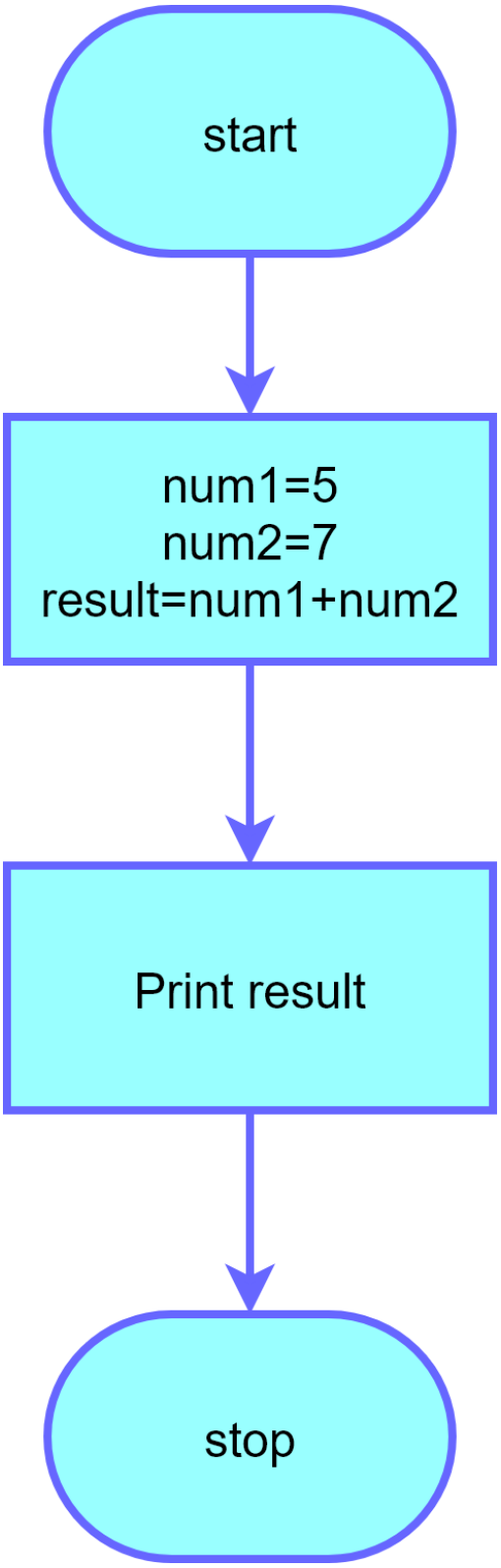
Task:

Write a program to add two numbers (5&7) and display its sum.

Algorithm:

1. Start
2. Assign Value to variables: (num1=5) (num2=7)
3. Calculate result as : result=num1+num2
4. Display result
5. stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: sum.c
 @CreatedDate: 2021/07/09

 program to add two numbers (5 and 7) and print its sum.

*/

#include <stdio.h> //header file (standard input output)
int main()          //main function
{
 int num1 = 5;                          //set variable num1 to 5
 int num2 = 7;                         //set variable num2 to 7
 int result = num1 + num2;       //add num1 and num2 and store it to result
 printf("The sum of %d and %d is: %d\n", num1, num2, result); //print
result
 return 0;                                 //return statement of int type
}
```

Output:

The sum of 5 and 7 is: 12

Discussion and conclusion:

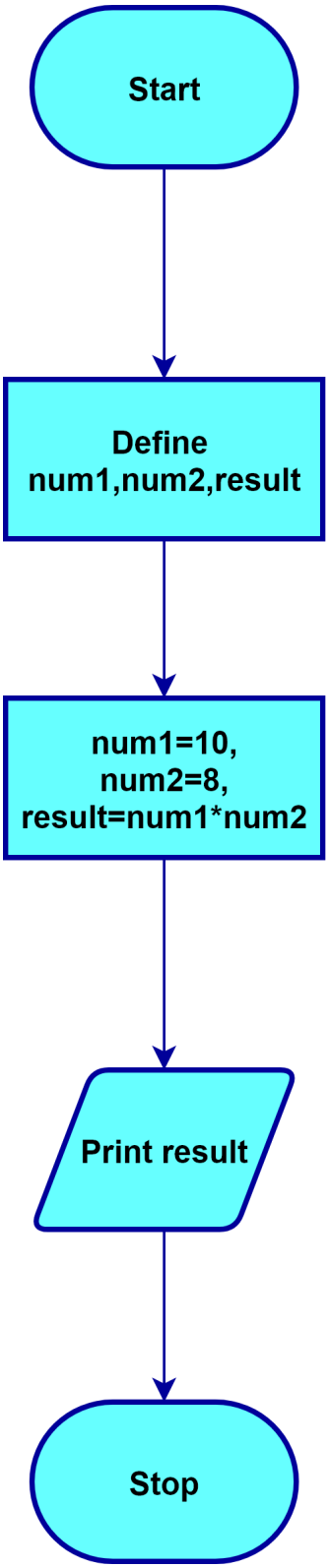
This problem focuses on adding two numbers (5 and 7) and printing its result.From this lab, I understood the basic structure of C programming including the meaning of header files and steps of problem solving as well as drawing flowchart.Hence sum of 5 and 7 were calculated.

Task:
Write a program to multiply two numbers (10&8) and display its product.

Algorithm:

1. Start
2. Define Variables: num1(int) ,num2(int),result(int)
3. Assign values to variables: (num1=10),(num2=8)
4. Calculate result as : result=num1*num2
5. Display result
6. Stop

Flowchart:


Start
Define
num1,num2,result
num1=10,
num2=8,
result=num1*num2
Print result
Stop

Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: multiply.c
 @CreatedDate: 2021/07/09

 Write a program to multiply two numbers (10&8)

*/

#include <stdio.h> //header file (standard input output)
int main()          //main function
{
 int num1, num2, result;      //Declare variable
 num1 = 10;                  //set variable num1 to 10
 num2 = 8;                  //set variable num2 to 8
 result = num1 * num2;     //multiply num1 and num2 and store it to result
 printf("The Product of %d and %d is: %d\n", num1, num2, result); //print
the result
 return 0;                //return statement of int type
}
```

Output:

The Product of 10 and 8 is: 80

Discussion and conclusion:

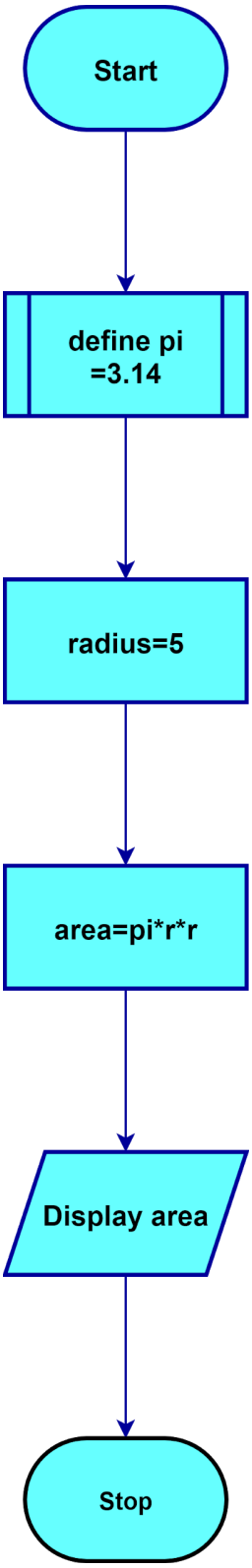
This problem focuses on multiplying two numbers (10 and 8) and printing its result.From this lab, I understood the basic structure of C programming including the meaning of header files and steps of problem solving as well as drawing flowchart.Hence products of 10 and 8 were calculated.

Task:
Write a program to calculate the area of a circle having its radius (r=5).

Algorithm:

1. Start
2. Define pi =3.14
3. Assign r=5
4. Calculate area as : Area=pi*r*r
5. Print area
6. stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: areaofcircle.c
 @CreatedDate: 2021/07/09

 Write a program to calculate the area of a circle having its radius
(r=5).

*/

#include <stdio.h> //header file (standard input output)
#define pi 3.14    //Preprocessor directive (pi is declared and save
before compilation)

int main() //main function
{
 int r = 5;                          //assign r to 5
 float area = pi * r * r;           //calculate area and store in float
 printf("The area of a given circle is %.2f.\n", area); //print area upto
two decimal place
 return 0;                         //exit status
}
```

Output:

The area of a given circle is 78.50.

Discussion and conclusion:

This problem focuses on finding an area of circle having radius 5 and printing its result.From this lab, I understood the basic structure of C programming including the meaning of header files ,use of preprocessor directive,and steps of problem solving as well as drawing flowchart.Hence area of circle was calculated.

Task:

Write a program to calculate the area of an ellipse having its axes (minor=4cm, major=6cm).

Problem Analysis:

The problem needs to calculate the area of ellipse having minor axis(4) and major axis(6).Area of ellipse is given as Area=$\pi * a * b$ (a=major) ,(b=minor).So we declare two variables of integer minor,major and area as float value.And we give value to minor as 4,major as 6 and calculate area using this formula:

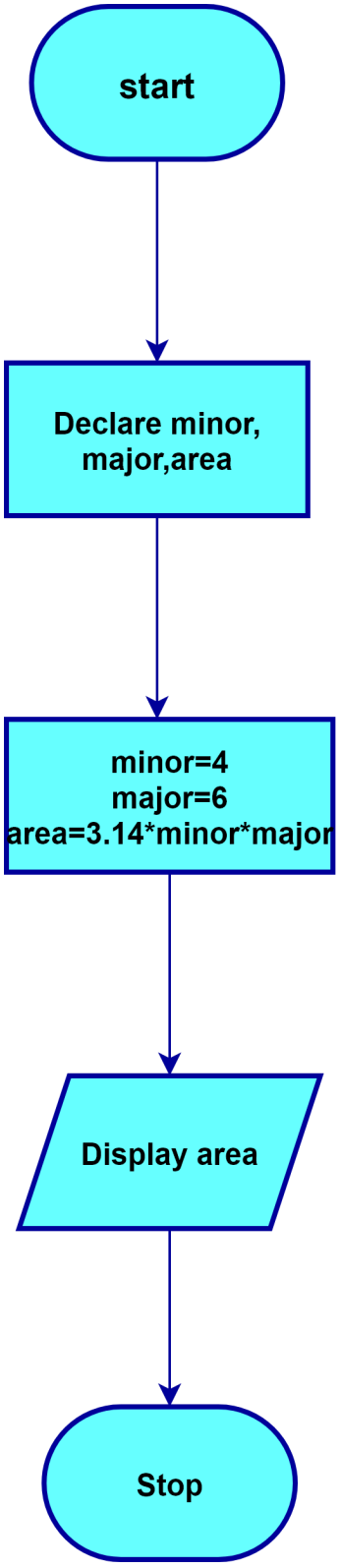
$$area = 3.14 * minor * major$$

Input variables	**Processing variables/ calculations**	**Output variables**	**Necessary header files**
minor(int) major(int) area(float)	$area$ =3.14*minor*major	area(float)	stdio.h

Algorithm:

1. Start
2. Define Variables :minor ,major,area (int)
3. Assign values as : minor = 4 ,major =6
4. Calculate area as :area=$3.14 * a * b$
5. Print area
6. Stop

Flowchart:


start
Declare minor, major,area
minor=4
major=6
area=3.14*minor*major
Display area
Stop

Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: areaofcircle.c
 @CreatedDate: 2021/07/09

  Write a program to calculate the area of an ellipse having its axes
(minor=4cm, major=6cm).
 */
#include <stdio.h> //header file
int main()
{
 int minor, major;
 float area;
 minor = 4;
 major = 6;
 area = 3.14 * minor * major;
 printf("The area of a given ellipse is %.2f.\n", area); //print the area
value upto two decimal point
 return 0;                                              //exit status code
}
```

Output:

The area of a given ellipse is 75.36.

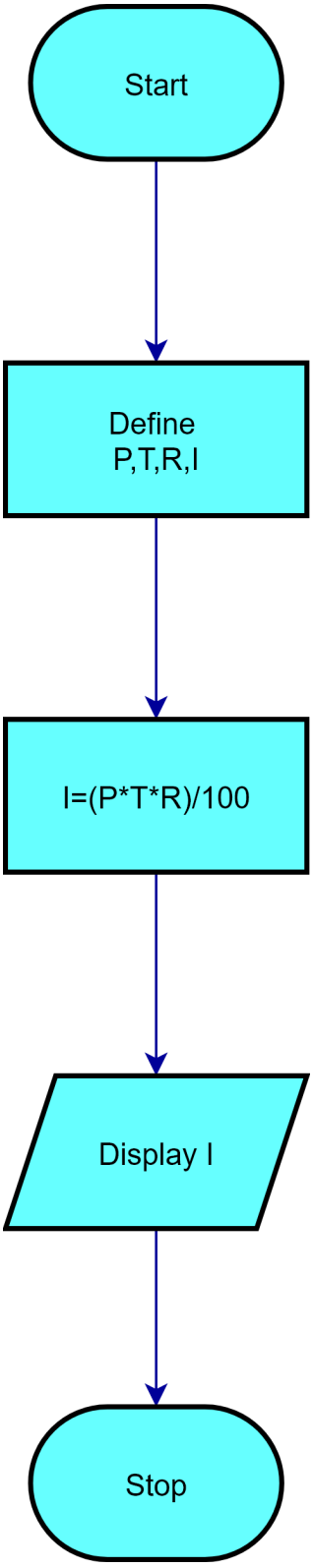
Task:

Write a program to calculate simple interest for a given P=4000, T=2, R=5.5. ($I = P * T * R/100$)

Algorithm:

1. Start
2. Declare variable as : P(int),T(int),R(float),Interest(float)
3. Assign value to variables as : P=4000,T=2,R=5.5
4. Calculate simple interest as ($I = P * T * R/100$)
5. Display simple interest
6. Stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename:simpleinterest.c
 @CreatedDate: 2021/07/09

 Write a program to calculate simple interest for a given P=4000,
T=2, R=5.5. (I =P*T*R/100)

*/

#include <stdio.h>
int main()
{
 int P, T;
 float R, I;
 P = 4000;
 T = 2;
 R = 5.5;
 I = (P * T * R) / 100.0;
 printf("Simple Interest is : %.2f\n", I);
 return 0;
}
```

Output:

Simple Interest is : 440.00

Task:

Write a program to declare two integers and one float variable then initialize them to 10,15, and 12.6. Also print the variable values on the screen.

Problem Analysis:

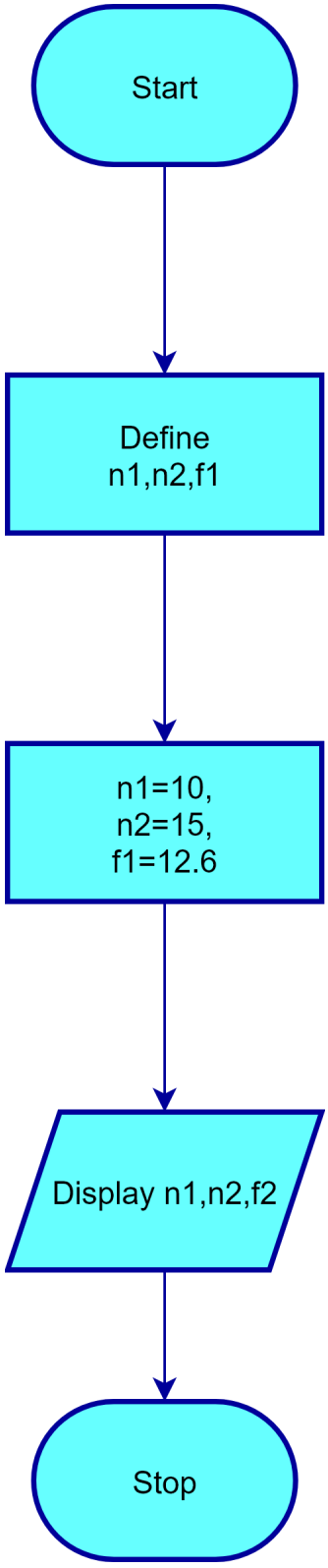
The problem is to declare two integers and one float variable and initialize them to 10,15,12.6.We declare two integer variables n1 ,n2 and f1 as float variable and store their value to 10,15,12.6 respectively. The output of the program is displayed on screen by printf function.

Input variables	Output variables	Header files
n1(int), n2(int), f1(float)	n1,n2,f1	stdio.h

Algorithm:

1. Start
2. Declare variable as:n1(int),n2(int),f1(float)
3. Assign values as : n1=10,n2=15,f1=12.6
4. Display n1,n2,f1
5. stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: declare.c
 @CreatedDate: 2021/07/09

 Write a program to declare two integers and one float variable then
initialize them to 10,15, and 12.6. Also print the variable values
on the screen.
*/

#include <stdio.h> //header file
int main()
{
 int n1, n2;
 float f1;
 n1 = 10;
 n2 = 15;
 f1 = 12.6;
 printf("The number are:\n n1=%d\n n2=%d\n f1=%.2f\n", n1, n2, f1);
//printing variables
 return 0;
//exit status
}
```

Output:

The number are:

n1=10

n2=15

f1=12.60

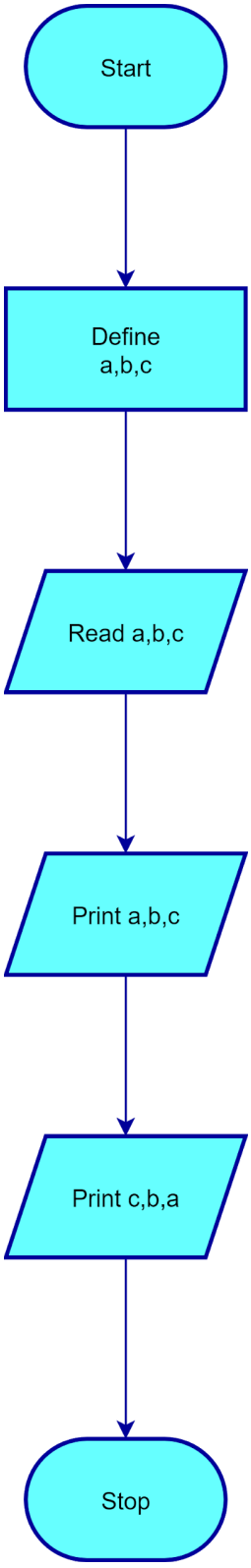
Task:

Write a C program to prompt the user to input 3 integer values and print these values in forward and reversed order.

Algorithm:

1. Start
2. Declare variable as : a(int),b(int),c(int)
3. Get input from user to a,b,c
4. Print a,b,c
5. Print c,b,a
6. Stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: initialize.c
 @CreatedDate: 2021/07/09

 Write a C program to prompt the user to input 3 integer values and
print these values in
 forward and reversed order.

*/

#include <stdio.h>
int main()
{
 int a, b, c;
 scanf("%d %d %d", &a, &b, &c); //scanf function to get user input
and & to save memory address
 printf("The number entered in order are : %d,%d ,%d.\n", a, b, c);
//print in order
 printf("The number entered in reversed order are : %d,%d ,%d.\n",
c, b, a); //in reverse order
 return 0;
}
```

Output:

15

16

17

The number entered in order are : 15,16 ,17.

The number entered in reversed order are : 17,16 ,15.

Task:

Write a program to calculate simple and compound interest.

Problem Analysis:

The problem is to find simple interest and compound interest for any Principle(p),time(t),number of period(n) and rate(r).We define input parameter as: p(int),t(int),r(float),n(int) and output variable as: SI(float), CI(float) and calculate as:

SI=$(P * T * R)/100$

CI=$P * (1 + r/n)^{nt} - P$

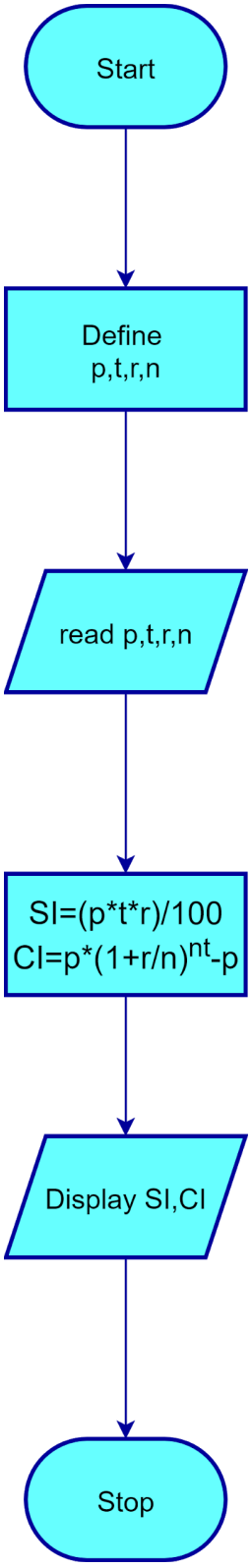
We need a math.h header file to perform exponential calculation using pow function.

Input variables	**Processing variables/ calculations**	**Output variables**	**Necessary header files**
p(int) t(int) n(int) r(float)	SI=$(P * T * R)/100$ CI=$P * (1 + r/n)^{nt} - P$	SIfloat) CI(float)	stdio.h math.h

Algorithm:

1. Start
2. Declare variable as : p(int),t(int),r(float),n(int),SI(float),CI(float)
3. Read p,t,r
4. SI=$(P * T * R)/100$
5. CI=$P * (1 + r/n)^{nt} - P$
6. Display SI
7. Display CI
8. Stop

Flowchart:


Start
Define
p,t,r,n
read p,t,r,n
SI=(p*t*r)/100
CI=p*(1+r/n)^nt-p
Display SI,CI
Stop

Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename:interest.c
 @CreatedDate: 2021/07/09

 Write a program to calculate simple and compound interest.

*/

#include <stdio.h> //header file
#include <math.h>  //math.h file to calculate power
int main()
{
 int p, t, n; //initialize variable
 float r, SI, CI;

 printf("Enter principle:\n");
 scanf("%d", &p); //getting input from user

 printf("Enter time:\n");
 scanf("%d", &t);

 printf("Enter number of periods:\n");
 scanf("%d", &n);

 printf("Enter rate:\n");
 scanf("%f", &r);

 SI = (p * t * r) / 100.0; //calculating simple interest
 CI = p * pow((1 + r / (100 * n)), n * t) - p; //calculating Compound
interest

 printf("Simple interest is %.2f and compound interest is %.2f.\n", SI,
CI);

 return 0;
}
```

Output:

Enter principle:
1000
Enter time:
10
Enter number of periods:
2
Enter rate:
10
Simple interest is 1000.00 and compound interest is 1653.30.

Discussion and conclusion:

This problem focuses on solving problem to find simple interest and compound interest for a period of time.From this lab ,I understood header files,linking external header file for compilation,using Pow function and solving mathematical problem to find simple and compound interest.Hence ,simple interest and compound interest is calculated and displayed in screen.

Task:

Write a program to swap two variables values with and without using third variables.

Problem Analysis:

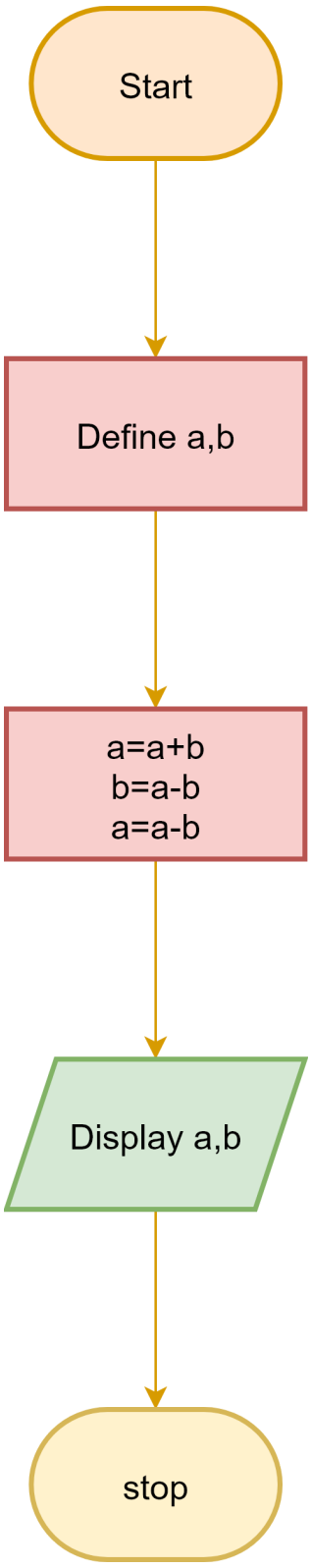
The problem can be break down into two parts:

1. Swap two variables without third variables
   First we declare two variables Say(a and b) and then we do some math as:
   A = A operation B (operation=+,inverse of + is -)
   B = A inverse-operation B (B=A-B)
   A = A inverse-operation B (A=A-B)
   We can solve it with multiplication and division operation if both are non zero.

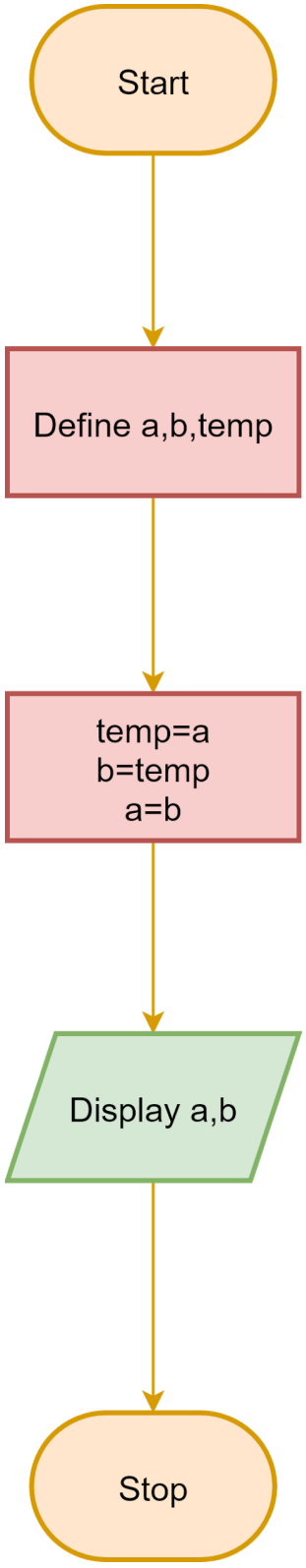
2. Swap two variables with third variables
   First we declare two variables Say(a and b) and then declare the third variable say temp and temp to a,a to b,and b=temp.so ,temp is store as a,a is stored as b,b is stored as temp (which is a).

Problem	**Input variables**	**Calculations**	**Output variables**	**Necessary headers file**
first	a,b	a=a+b b=a-b a=a-b	a,b	stdio.h
second	a,b,temp	temp=a a=b b=temp	a,b	stdio.h

Flowchart#1:



Flowchart#2:



Source code#1:

```
/*
 @Author : Kishan Adhikari
 @Filename: swipe.c
 @CreatedDate: 2021/07/09



 Write a program to swap two variables values without using third
variables
*/

#include <stdio.h>
int main()
{
 char a, b;


 printf("Enter value of a:\n");
 scanf(" %c", &a);   // whitespace before %c is required to read
next line


 printf("Enter value of b\n");
 scanf(" %c", &b);


 printf("Value before swapping is: a=%c,b=%c\n", a, b);


 a = a + b; //a=4,b=5,a=9
 b = a - b; //b=9-5:b=4
 a = a - b; //b=4:a=9-4:a=5

 printf("value after swapping is: a=%c,b=%c\n", a, b);
 return 0;
}
```

Source code#2:

```c
/*
 @Author : Kishan Adhikari
 @Filename: swipewith.c
 @CreatedDate: 2021/07/09


 Write a program to swap two variables values with using third
variables
*/

int main()
{

 char a, b, temp;

 printf("Enter value of a:\n");
 scanf(" %c", &a); // whitespace before %c is required to read next
line
 printf("Enter value of b\n");
 scanf(" %c", &b);
 printf("Value before swapping is: a=%c,b=%c\n", a, b);

 temp = a;
 a = b;
 b = temp;

 printf("Value after swapping is a=%c,b=%c\n", a, b);
}
```

Output#1:

Enter value of a:

4

Enter value of b

5

Value before swapping is: a=4,b=5

value after swapping is: a=5,b=4

Output#2:

Enter value of a:

a

Enter value of b

b

Value before swapping is: a=a,b=b

Value after swapping is a=b,b=a

Task:

Write a program to check odd or even number

(a) using modulus operator

(b) using bitwise operator

(c) without using bitwise and modulus operator

(d) using conditional operator.

Problem analysis:

Odd numbers are those that cannot be divided by 2 evenly but even numbers can be.for example 1,3,5,7,9… are odd numbers and 2,4,6,8,.... are even numbers.

(a).To check odd or even using modulus operator

Modulus operator(%) gives us the remainder.So,performing any number N modulo 2 gives us its remainder.If it's evenly divided then its remainder is zero and the number is odd otherwise even.

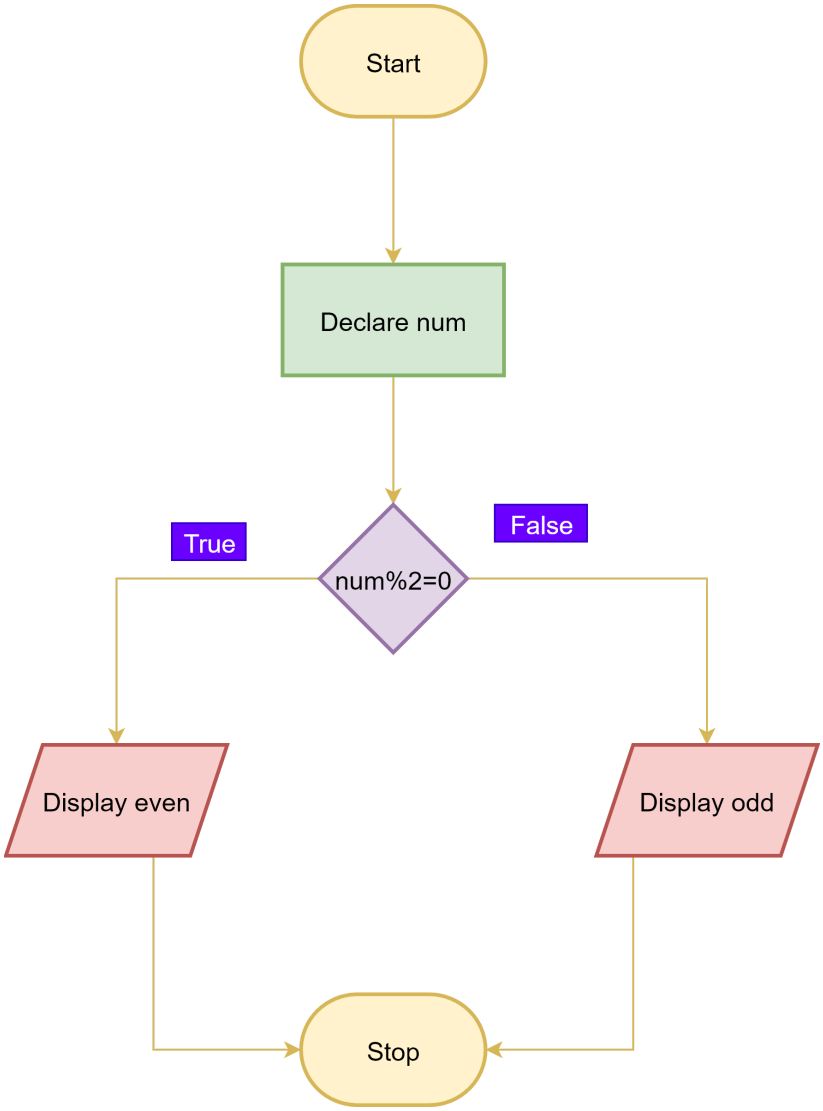
Input variables	Calculations	Output variables	Necessary headers file
Num(int)	Num%2		stdio.h

Algorithm:

1. Start
2. Define Num
3. Read Num
4. If Num%2=0 :Display even
5. Else Display odd

6. Stop

Flowchart:



Source code:

```c
/*
 @Author : Kishan Adhikari
 @Filename: oddeven.c
 @CreatedDate: 2021/07/11

  Write a program To check odd or even using modulus operator
 */
#include <stdio.h>
int main()
{
 int num;
 printf("Enter any number:\n");
 scanf("%d", &num);  //get input fromuser
 if (num % 2 == 0)  //check condition
 {
   printf("%d is an even number\n", num);
 }
 else
 {
   printf("%d is a odd number\n", num);
 }
 return 0;
}
```

Output:

Enter any number:

14

14 is an even

Enter any number:

17

17 is a odd

(b).To check odd or even using bitwise operator

Problem Analysis:

Bitwise AND (&) gives us the 1 if both numbers are 1 otherwise zero.We know even numbers are divided by 2 leaving zero remainder.So,any even number in binary has last digit as zero and odd has 1.

Let's take few example (num=12 which in binary is 1100)

```
    1100
&   0001
-----------
    0000
```

For num=9 (1001)

```
    1001
&   0001
-----------
    0001
```

So,odd number when performed Bitwise And operation(&) give one and even numbers give zero.

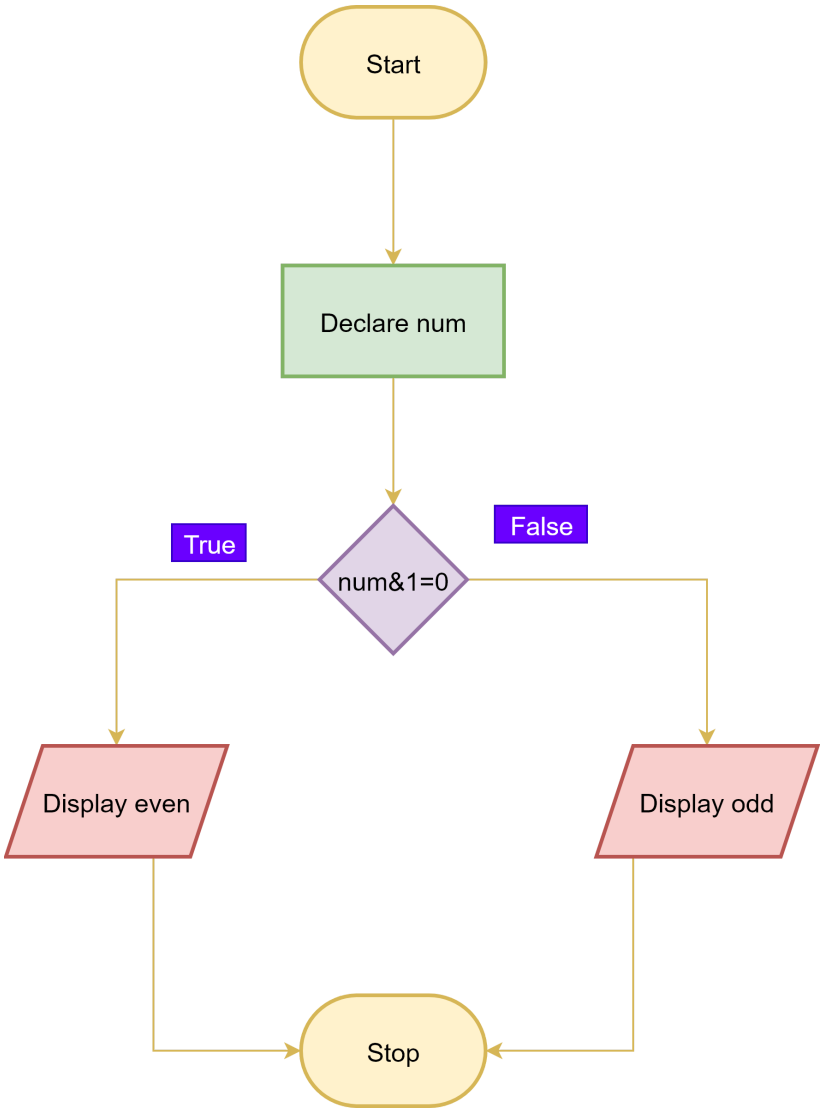
Input variables	Calculations	Output variables	Necessary headers file
Num(int)	Num & 1		stdio.h

Algorithm:

1. Start
2. Define num
3. Read num
4. num&1=0 true display even

5. Otherwise display odd
6. Stop

Flowchart:



Source code:

```c
/*
 @Author : Kishan Adhikari
 @Filename: bitwiseoddeven.c
 @CreatedDate: 2021/07/10

 To check odd or even using bitwise operator
*/

#include <stdio.h>
int main()
{
 int num;
 printf("Enter any number:\n");
 scanf("%d", &num);
 switch (num & 1)    //bitwise and
 {
 case 0:
   printf("%d is an even number.\n", num);
   break;

 case 1:
   printf("%d is a odd number\n", num);
   break;
 }
 return 0;
}
```

Output:

Enter any number:
259
259 is a odd number

Enter any number:
1578
1578 is an even number.

(C). To check odd or even without using bitwise and modulus operator

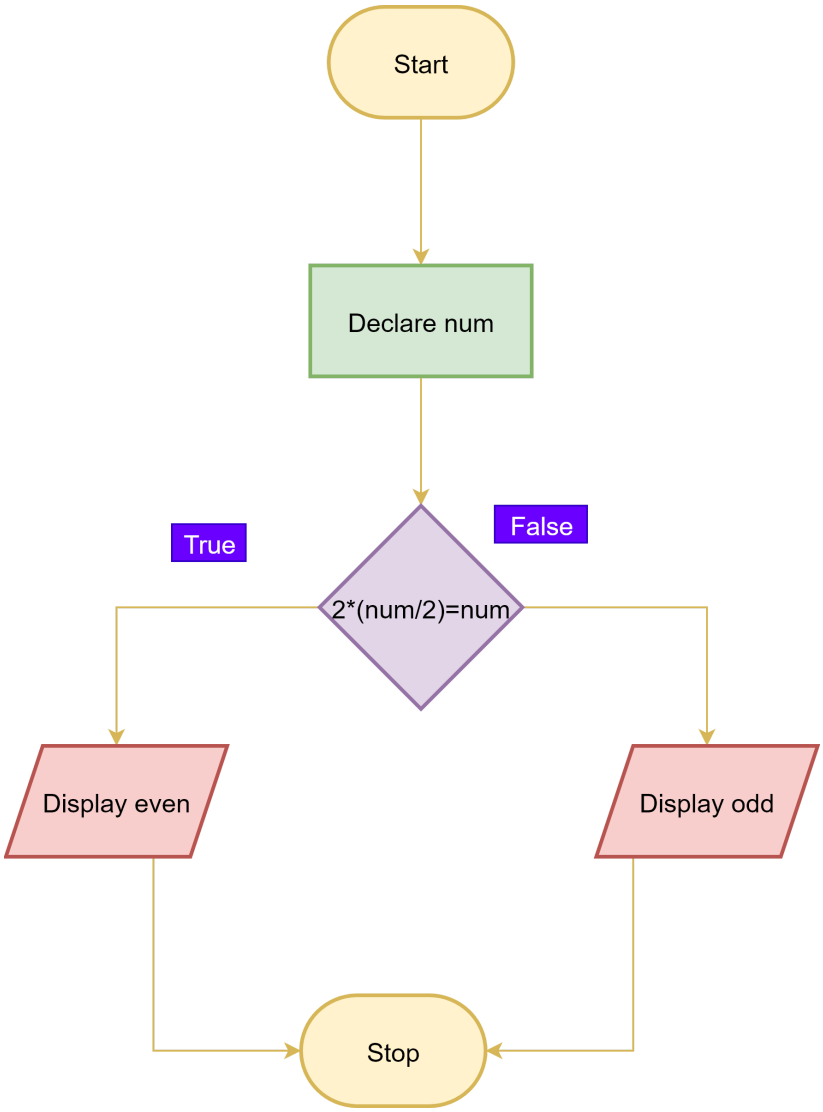
Problem Analysis:
The problem is to check whether a given number is odd or even without using bitwise and modulus operators. We define an integer N and read its value from the user.To accomplish this task we divide N by 2 and multiply it by 2.In C program division (/) operator gives nearest integer value round to zero.So, 25/2 gives 12 which when multiplied by 2 gives 24 which is not equal to 25.But when we take even number say 48 ,which divided by 2 (48/2) gives 24 and then multiplied by 2 gives 48 which is given number. Using this logic ,we solve this problem.

Input variables	**Calculations**	**Output variables**	**Necessary headers file**
Num(int)	2*(Num / 2)		stdio.h

Algorithm:

1. Start
2. Define variable :num(int)
3. Read num
4. 2*(num/2)==num
5. true display even
6. Else display odd
7. stop

Flowchart:


Start
Declare num
True
False
2*(num/2)=num
Display even
Display odd
Stop

Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: divoddeven.c
 @CreatedDate: 2021/07/11

  Write a program To check odd or even without using bitwise and modulus
operator.
 */
#include <stdio.h>
int main()
{
 int num;
 printf("Enter any number:\n");
 scanf("%d", &num);
 if (2 * (num / 2) == num)
 {
   printf("%d is an even number\n", num);
 }
 else
 {
   printf("%d is a odd number\n", num);
 }
}
```

Output:

Enter any number:

17

17 is a odd number

Enter any number:

458

458 is an even number

Task:

(d).To check odd or even using conditional operator.

Problem Analysis

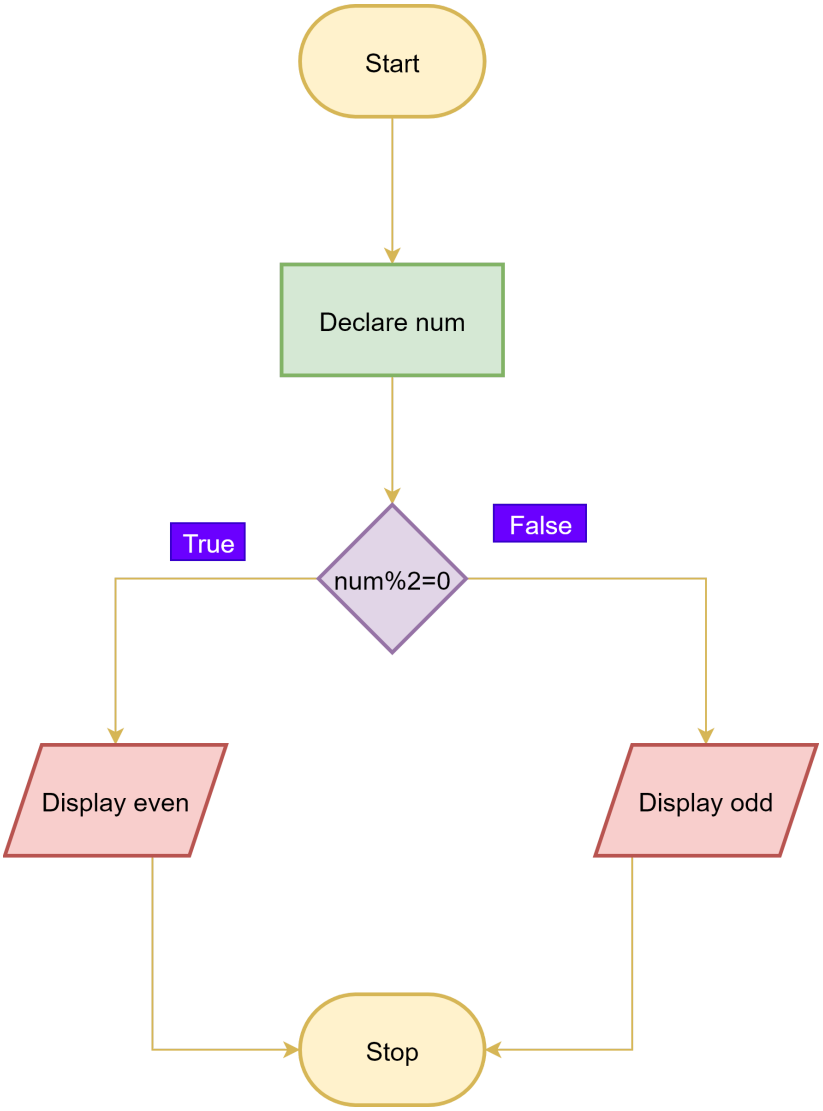
Modulus operator(%) gives us the remainder.So,performing any number N modulo 2 gives us its remainder.If it's evenly divided then its remainder is zero and the number is odd otherwise even.

Input variables	Condition	Output variables	Necessary headers file
Num(int)	Num%2=0		stdio.h

Algorithm:

1. Start
2. Define num
3. Read num
4. num%2=0
5. True : display even
6. False :display odd
7. stop

Flowchart:


Start
Declare num
num%2=0
True
False
Display even
Display odd
Stop

Source Code:

```c
/*
 @Author : Kishan Adhikari
 @Filename: oddeven.c
 @CreatedDate: 2021/07/10

 To check odd or even using modulus operator
*/

#include <stdio.h>
int main()
{
 int num;
 printf("Enter any number:\n");
 scanf("%d", &num);
 (num % 2 == 0) ? printf("%d is an even\n", num) : printf("%d is a
odd\n", num);  //ternary operator also known as conditional operator
 return 0; //exit status
}
```

Output:

Enter any number:

178

178 is an even number

Enter any number:

19

19 is a odd number

Task:

Print the value of y for given x=2 & z=4and analyze the output.

a. y = x++ + ++x;

b. y=++x + ++x;

c. y= ++x + ++x + ++x;

d. y = x>z;

e. y= x>z? x: z;

f. y = x&z;

g. y= x>>2 + z<<1;

Problem Analysis:

The problem is to analyze different output given by performing operations such as postfix increment,prefix increment,logical and,bit shifting etc. We define three integer variables as x(int),y(int),z(int) and perform the above operation and analyze the output.

Input variables	**Calculations**	**Output variables**	**Necessary headers file**

x(int)
y(int)
z(int)
	a. y = x++ + ++x;
	b. y=++x + ++x;
	c. y= ++x + ++x + ++x;
	d. y = x>z;
	e. y= x>z? x: z;
	f. y = x&z;
	g. y= x>>2 + z<<1;
		y(int)	stdio.h

Algorithm:

1. Start
2. Define variable as :x(int),y(int),z(int)
3. Assign value as x=2,z=4
4. Perform operation as:

   y=x++ + ++x ,

   y=++x + ++x

   y=++x + ++x + ++x;

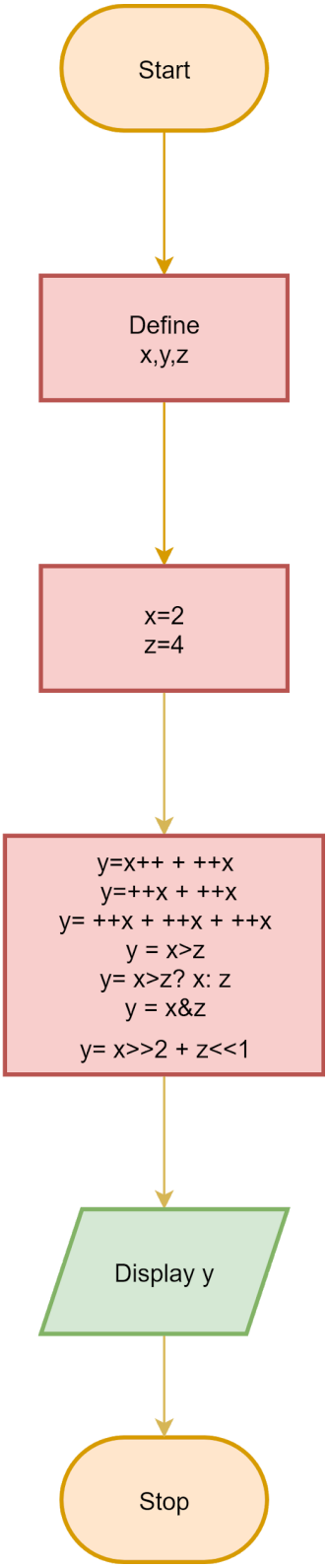
   y=x>z;

   y=x>z?x:z;

   y=x&x;

   y=x>>2+z<<1;
5. Display y in each case
6. Stop

Flowchart:


Start
Define
x,y,z
x=2
z=4
y=x++ + ++x
y=++x + ++x
y= ++x + ++x + ++x
y = x>z
y= x>z? x: z
y = x&z
y= x>>2 + z<<1
Display y
Stop

Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: operator.c
 @CreatedDate: 2021/07/12


 Print the value of y for given x=2 & z=4and analyze the output.
 a. y = x++ + ++x;
 b. y=++x + ++x;
 c. y= ++x + ++x + ++x;
 d. y = x>z;
 e. y= x>z? x: z;
 f. y = x&z;
 g. y= x>>2 + z<<1


*/
#include <stdio.h>
int main()
{
 int x;
 int z;
 int y;
 x = 2, z = 4;
 //  x=2; x then becomes (3)+x becomes 4,x is assigned to 4 ==>
y=2+4
 y = x++ + ++x;       //Assigning values
//x++ is postfix and ++x is prefix operator "++x" -> increments "x"
then uses value "x" "x++" -> uses value "x" then increments "x"
 printf("The value of y after operation( y=x++ + ++x) is %d.\n", y);
 printf("The value of x =%d\n\n", x);
//x=4

 //  5 (5)+ 6(6) : 6+6
 y = ++x + ++x;
 printf("The value of y after operation(y = ++x + ++x) is %d.\n",
```

```
y);
 printf("The value of x =%d\n\n", x); //x=6
 y = ++x + ++x + ++x;
 printf("The value of y after operation(y = ++x + ++x + ++x) is
%d.\n", y);
 printf("The value of x =%d\n\n", x); //x=9

 y = x > z; //greater than (if true gives 1 otherwise gives 0)
 // printf("The value of x =%d\n", x);
 printf("The value of y after operation(x > z) is %d.\n\n", y);

 printf("The value of x =%d\n", x);
 y = x > z ? x : z; //if x is greater than x then x else z
 printf("The value of y after operation(y = x > z ? x : z) is
%d.\n\n", y);

 y = x & z; //bitwise and operation
 printf("The value of y after operation( y = x & z) is %d.\n\n", y);

 y = x >> 2 + z << 1; //left shift and right shift
 printf("The value of y after operation(y = x >> 2 + z << 1) is
%d.\n", y);
 return 0;
}
```

Output:

The value of y after operation(y=x++ + ++x) is 6.
The value of x =4

The value of y after operation(y = ++x + ++x) is 12.
The value of x =6

The value of y after operation(y = ++x + ++x + ++x) is 25.
The value of x =9

The value of y after operation(x > z) is 1.

The value of x =9
The value of y after operation(y = x > z ? x : z) is 9.

The value of y after operation(y = x & z) is 0.

The value of y after operation(y = x >> 2 + z << 1) is 0.

Discussion and conclusion:

(a). y = x++ + ++x

The output of the above program is 6.It is because x++ is a postfix operator which first assigns value and then increments by 1 and ++x is a prefix operator which first increments x by 1 and then assigns values to it.

x++ = 2 (x first becomes 2 and then it increases by 1)
++x = 3+1=4 (x is 3 and is increment by 1 and becomes 4)
y=4+2=6

(b).y=++x + ++x

The output of the above program is 12.

x is 4 (from (a)).
++x =5 (increases 4 by 1 and assign it to x)
++x=6 (increases 5 by 1 and assign it)
So x has become 6.
y=6+6=12

(c). y = ++x + ++x + ++x

The output of the above program is 25. C compiler doesn't do all three operations at once . It operate first two and then operate the third one as:
y=(++x + ++x) + ++x

x is 6.
++x =7(increase x by 1 and assign x to 7) and x becomes 8
++x=8(increases by 1 and assign x to 8)
++x=9(increases by 1)
y=8+8+9=25

(d). x>z

It check whether x is greater than z or not .If condition is true it return 1 else return 0.
Since x is 9 and z is 4.
9>4 which is true .
Thus ,the output is 1.

(e).y = x > z ? x : z

It checks whether x is greater than z or not .It x is greater than z then it returns x otherwise it returns z.

?=if

: else

x>z?x:z

9>4?9:4 (which is true)

Thus,the output is 9.

(f).y = x & z

& is bitwise operator(bitwise and).so it gives convert given number into binary and perform And operation (1 if both are 1 else 0) and convert to integer value.

x(9) in binary =1001

z (4) in binary =100

```
        1001
&       0100
-------------
        0000
```

Thus ,the output is zero.

Note: Logical operator and bitwise operator are different because logical operator gives binary i.e 1 and 0 value but bitwise operator give integer value.

(g).**y = x >> 2 + z << 1**

A bit shift moves each digit in a number's binary representation left or right. “>>” is a right shift whereas “<<” is left bit shift.

x>>2 shift x by 2 bits on the right side.
1001>>2 =0010
z<<1 shifts z by 1 bits on left side.
0100<<1 =1000
0010+1000=0000

Thus ,the output is zero

Hence, from this lab ,I understood the basics of operator , bitwise shift,postfix,prefix etc.

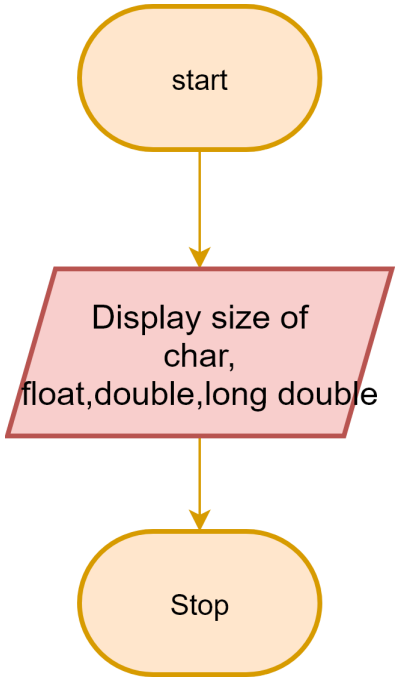
Task:

Write a program to print the size of char, float, double and long double data types in C.

Algorithm:

1. Start
2. Display size of (char),float,double,long double
3. Stop

Flowchart:



Source code:

```
/*
 @Author : Kishan Adhikari
 @Filename: size.c
 @CreatedDate: 2021/07/13

 Write a program to print the size of char, float, double and long double data types in C

*/

#include <stdio.h>
int main()
{
 printf("type\t\t\tbytes\n");
 printf("char\t\t\t%zu\n", sizeof(char)); //sizeof function to get size in byte
 printf("float\t\t\t%zu\n", sizeof(float));
 printf("double\t\t\t%zu\n", sizeof(double));
 printf("longdouble\t\t%zu\n", sizeof(long double));
 return 0;
}
```

Output:

type	bytes
char	1
float	4
double	8
longdouble	16

Discussion and conclusion:

Hence , from the first lab I understood the basic concept of structure of C programming,Using mathematical,logical as well as bitwise operators,Use of ternary condition ,if else condition. Also , I understood the problem solving approach,algorithm writing and flowchart making.

The whole program is written,Debugged and compiled on Vs code terminal in Ubuntu 18.04 Os with gcc version 7.5.0.[1]

[1] The end