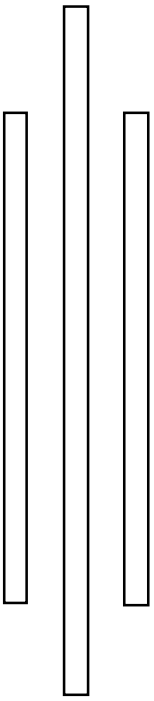
TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Lab Sheet #3



PURWANCHAL CAMPUS

DHARAN-8

Submitted by:

Name: **Arbind Kumar Mehta**

Roll No: **PUR075BCT017**

Faculty: BCT

Group: I/I 'A'

Date: .....

Submitted to:

Department of

Electronics & Computer

Engineering

Checked by: .....

Title:

Get input of two float numbers in to variables x & y. receive the mathematical operator (+, -, *, /) using unformatted I/O into the variable Ch1 and perform operations on x & y and display the result.

Objective:

- To be familiar with different data types, operation, expression and unformatted I/O in C

Problem Analysis:

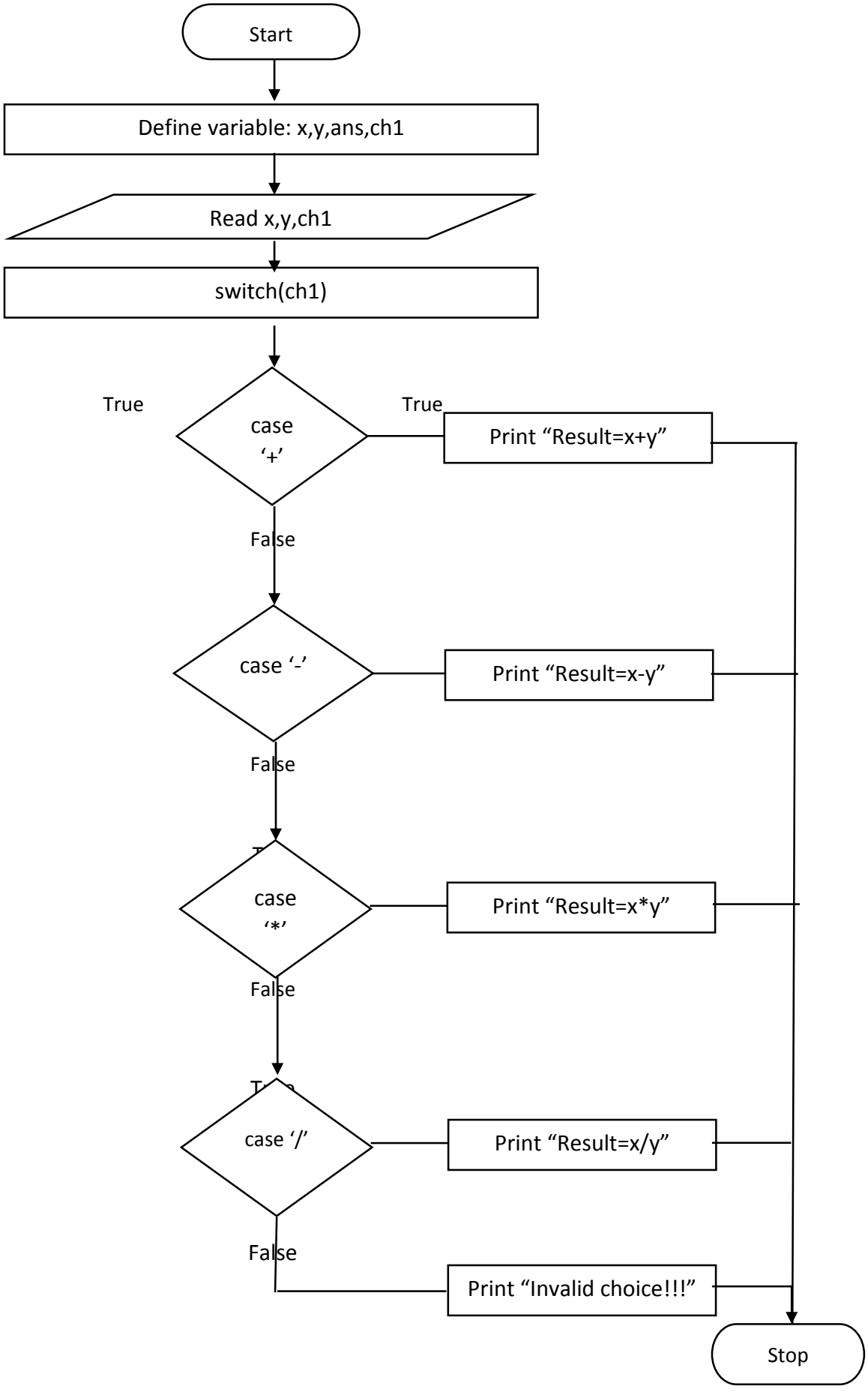
Based on problem, it is required to get the input of two float variables x and y. Different mathamatical operation should be performed by using unformatted I/O.

Input variables	Processing variables/calculations	Output variables	Necessary header files/functions/macros
x, y (float type)	ch1 (char type)	ans(float type)	stdio.h coino.h scanf() printf() switch() getch()

Algorithm:

1. Start
2. Define variables: x,y,ans,ch1
3. Take input from keyboard for all the input variables
4. Use switch statement to get ans
5. Display the ans
6. Stop

Flowchart:

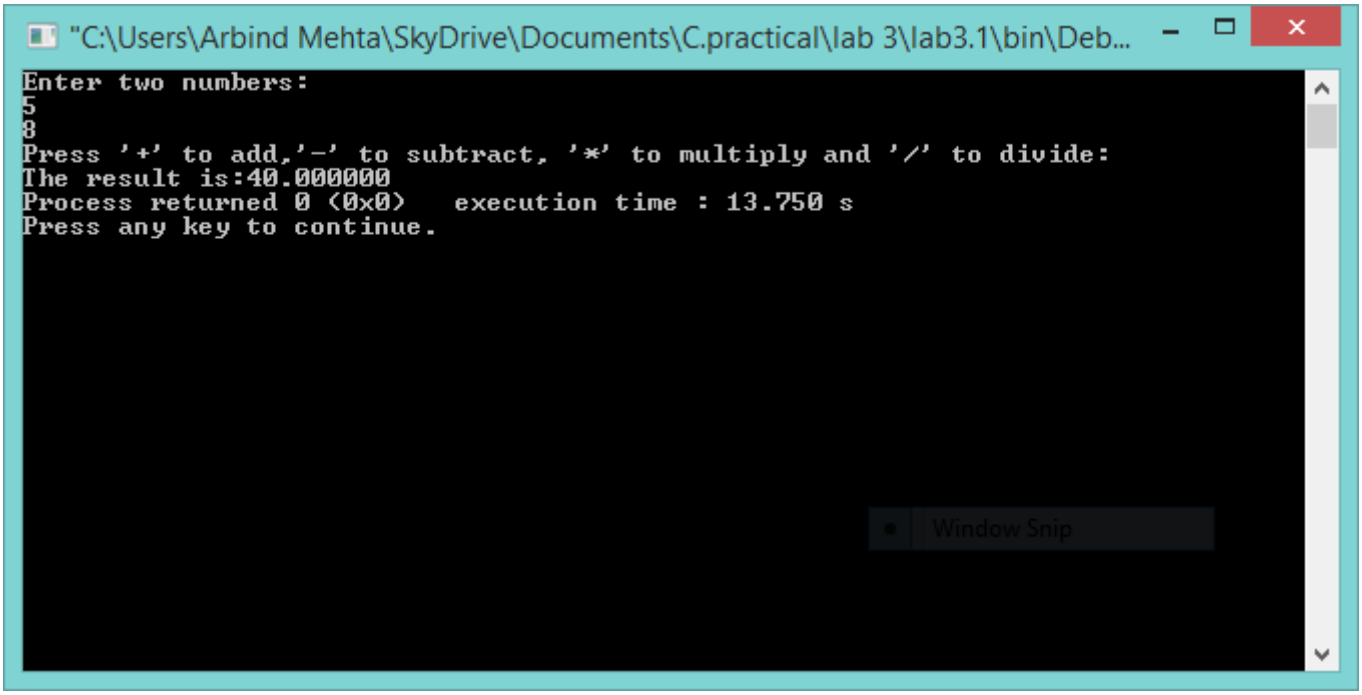

Start
Define variable: x,y,ans,ch1
Read x,y,ch1
switch(ch1)
True
case '+'
True
Print "Result=x+y"
False
case '-'
Print "Result=x-y"
False
case '*'
Print "Result=x*y"
False
case '/'
Print "Result=x/y"
False
Print "Invalid choice!!!"
Stop

Code:

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

int main()
{
   float x,y,ans;
   char ch1;

   printf("Enter two numbers:\n");
   scanf("%f%f",&x,&y);
   printf("Press '+' to add,'-' to subtract, '*' to multiply and '/' to divide:\n");
   ch1=getch();
   switch(ch1)
   {
      case '+':
      ans=x+y;
      break;

      case '-':
      ans=x-y;
      break;

      case '*':
      ans=x*y;
      break;
```

```
        case '/':
        ans=x/y;
        break;

        default:
        printf("Invalid choice!!!");
    }

    printf("The result is:%f",ans);
    getch();
    return 0;
}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types with unformatted input/output functions.

TITLE:

Define the math operator '+' as PLUS, '-' as MINUS, '*' as MULT & '/' as DIVIDE using preprocessor directives and do the operations over variables (x,y) defined on above question like z=x PLUS y.

Objective:

- To know different types of data types, operation,
- To be familier with macros.

Problem analysis:

Based on given problem, our program must define three variables of type float and one of char type. Different operation should be define using macros.

Input variables	**Necessary header files/functions/macros**
x,y,ans(float type) ch1(char type)	stdio.h define plus + coino.h define minus - printf() define mult * scanf() define div /

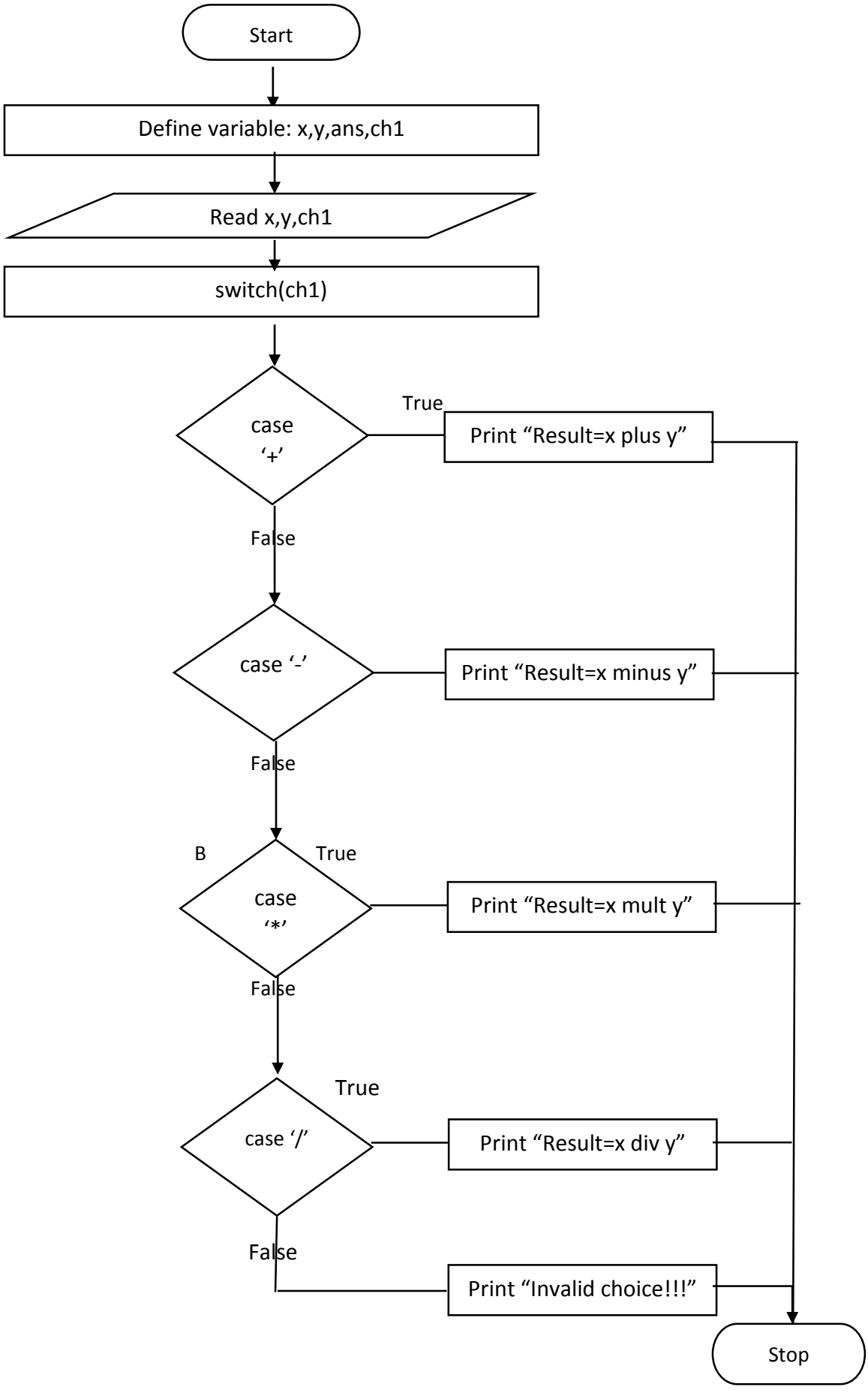
Algorithm:

1. Start
2. Define variable: x,y,ans,ch1
3. Read two number
4. Switch(ch1)

case '+':print "Result is: x plus y
case '-':print "Result is: x minus y
case '*':print "Result is: x mult y
case '/':print "Result is: x div y

5. Stop.

Flowchart:

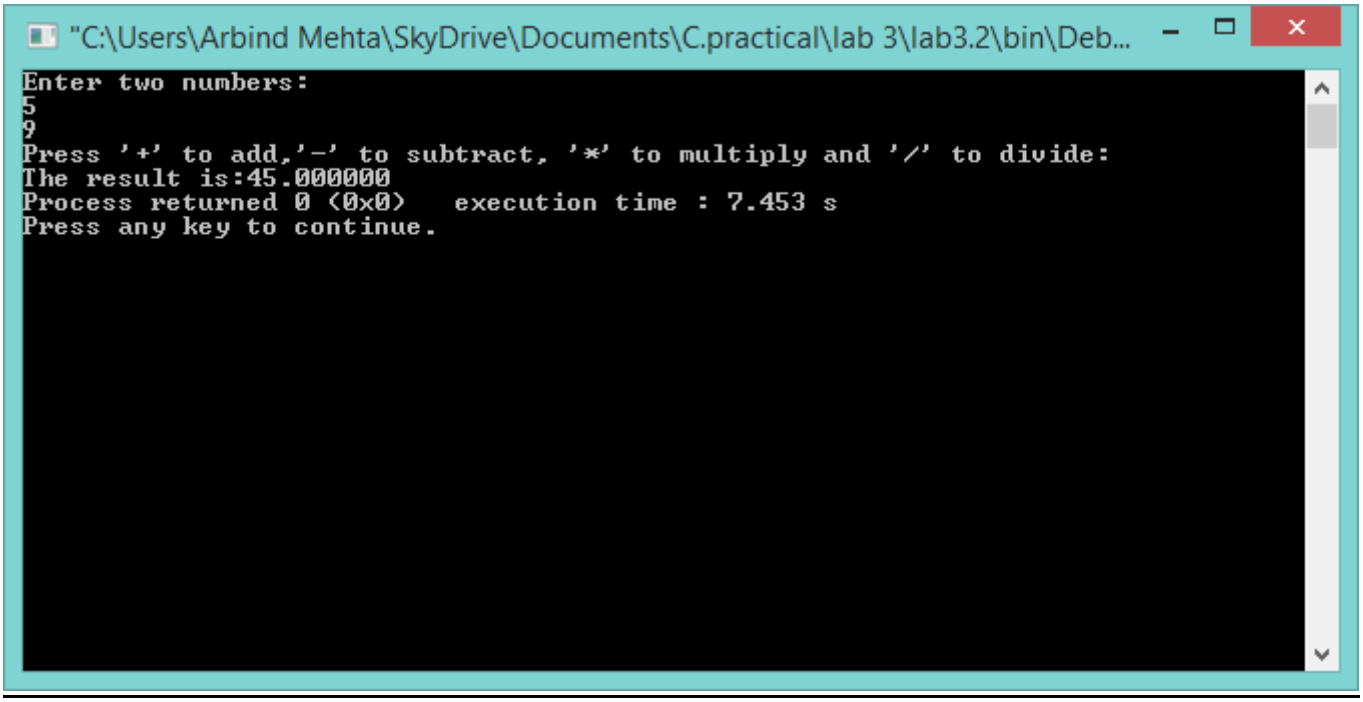

Start
Define variable: x,y,ans,ch1
Read x,y,ch1
switch(ch1)
case '+'
True
Print "Result=x plus y"
False
case '-'
Print "Result=x minus y"
False
B
True
case '*'
Print "Result=x mult y"
False
True
case '/'
Print "Result=x div y"
False
Print "Invalid choice!!!"
Stop

Code:

```
#include <stdio.h>
#define plus +
#define minus -
#define mult *
#define div /

int main()
{
   float x,y,ans;
   char ch1;

   printf("Enter two numbers:\n");
   scanf("%f%f",&x,&y);
   printf("Press '+' to add,'-' to subtract, '*' to multiply and '/' to divide:\n");
   ch1=getch();
   switch(ch1)
   {
      case '+':
      ans=x plus y;
      break;

      case '-':
      ans=x minus y;
      break;

      case '*':
      ans=x mult y;
```

```
        break;

        case '/':
        ans=x div y;
        break;

        default:
        printf("Invalid choice!!!");
    }

    printf("The result is:%f",ans);

    return 0;
}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types with unformatted input/output functions, macros and preprocessor directives in C.

TITLE:

Get input of your name, address, age in years, weight and height from keyboard and display the information using unformatted I/O (String I/O).

Objective:

- To know different types of data types, operation,
- To be familier with unformatted I/O in C.

Problem analysis:

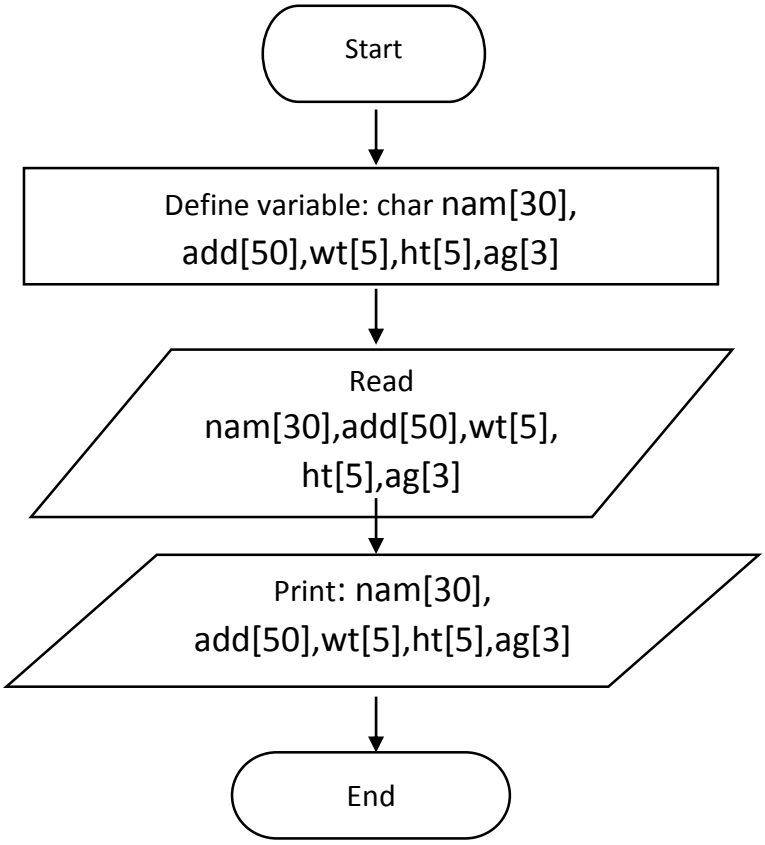
Based on given problem, our program must define five variables of type char. Different operation should be performed using unformatted I/O.

Input variables	**Necessary header files/functions/macros**
nam[30], add[50],wt[5],ht[5],ag[3] (char type)	stdio.h coino.h gets() puts()

Algorithm:

1. Start
2. Define variable: nam[30], add[50],wt[5],ht[5],ag[3]
3. Read variables using gets() function
4. Display values using puts() function
5. Stop.

Flowchart:

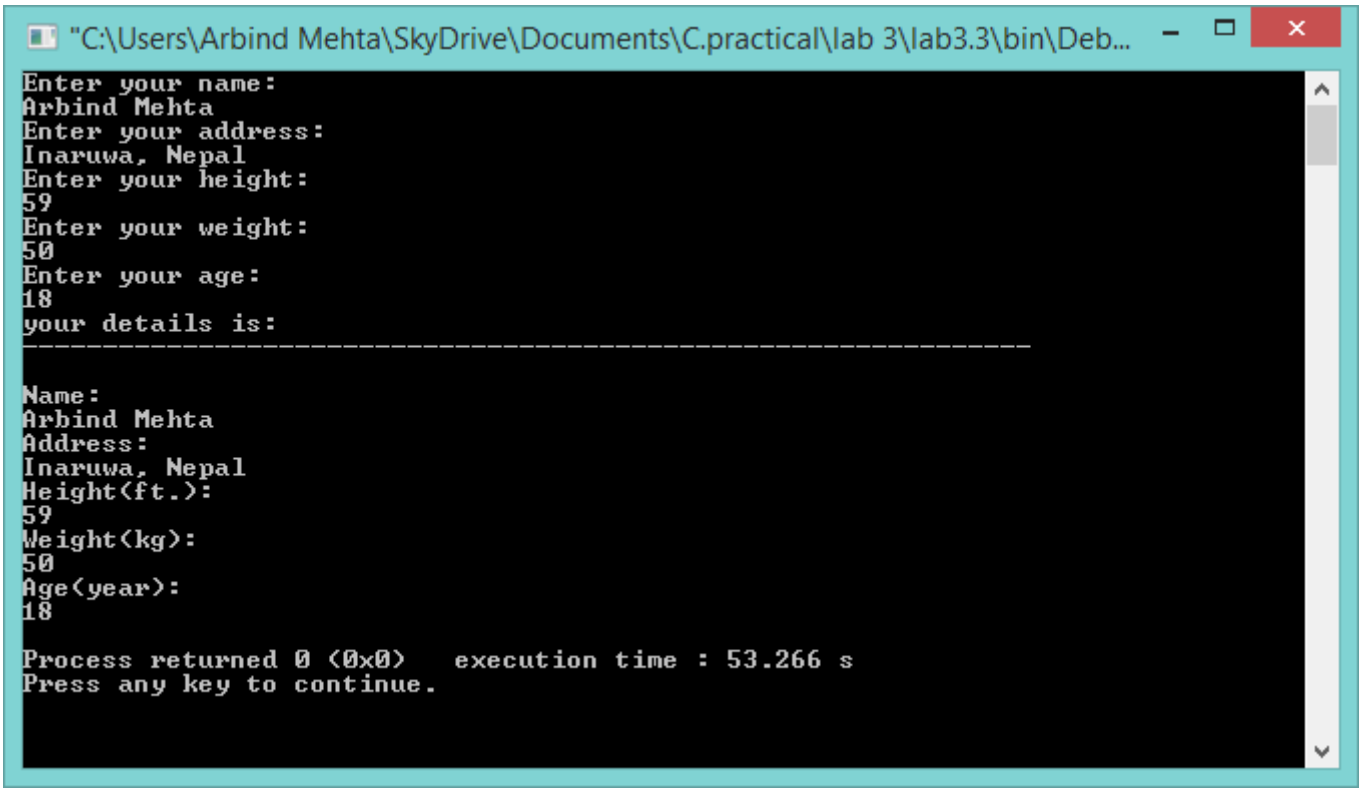


Code:

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


void main()
{
   char nam[30], add[50],wt[5],ht[5],ag[3];


   puts("Enter your name:");
   gets(nam);
```

```
    puts("Enter your address:");
    gets(add);
    puts("Enter your height:");
    gets(ht);
    puts("Enter your weight:");
    gets(wt);
    puts("Enter your age:");
    gets(ag);

    puts("your details is:\n------------------------------------------------------------\n");
    puts("Name:");
    puts(nam);
    puts("Address:");
    puts(add);
    puts("Height(ft.):");
    puts(ht);
    puts("Weight(kg):");
    puts(wt);
    puts("Age(year):");
    puts(ag);


}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types with unformatted input/output in C.

TITLE:

Write a program to produce the output as shown below:

x	y	expressions	results
6	3	x=y+3	x=6
6	3	x=y-2	x=1
6	3	x=y*5	x=15
6	3	x=x/y	x=2
6	3	x=x%y	x=0

Objective:

- To know different types of data types, operation,
- To be familier with formatted I/O in C.

Problem analysis:

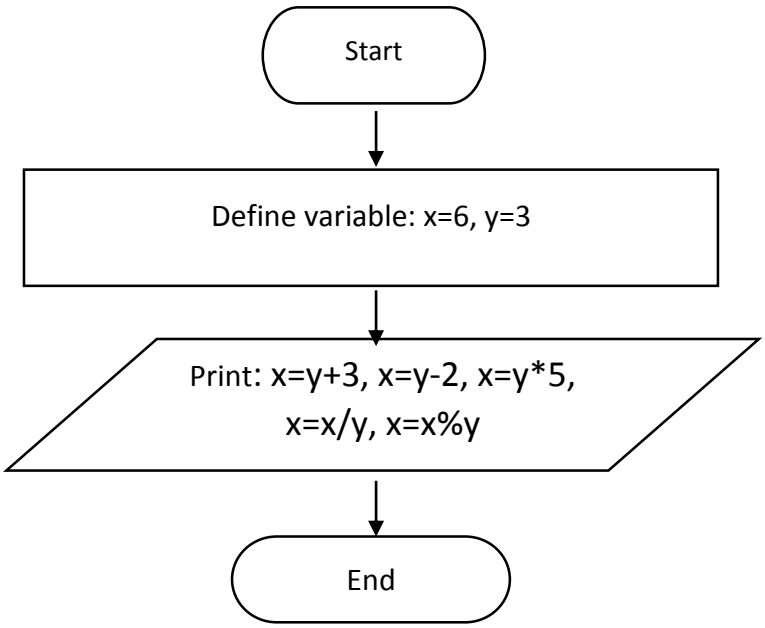
Based on given problem, our program must define two variables of type int. Different operation should be performed using formatted I/O.

Input variables	Necessary header files/functions/macros
x=6, y=3 (int type)	stdio.h coino.h printf()

Algorithm:

1. Start
2. Define variable: x=6, y=3
3. Display: x=y+3, x=y-2, x=y*5, x=x/y, x=x%y using printf() function
4. Stop.

Flowchart:

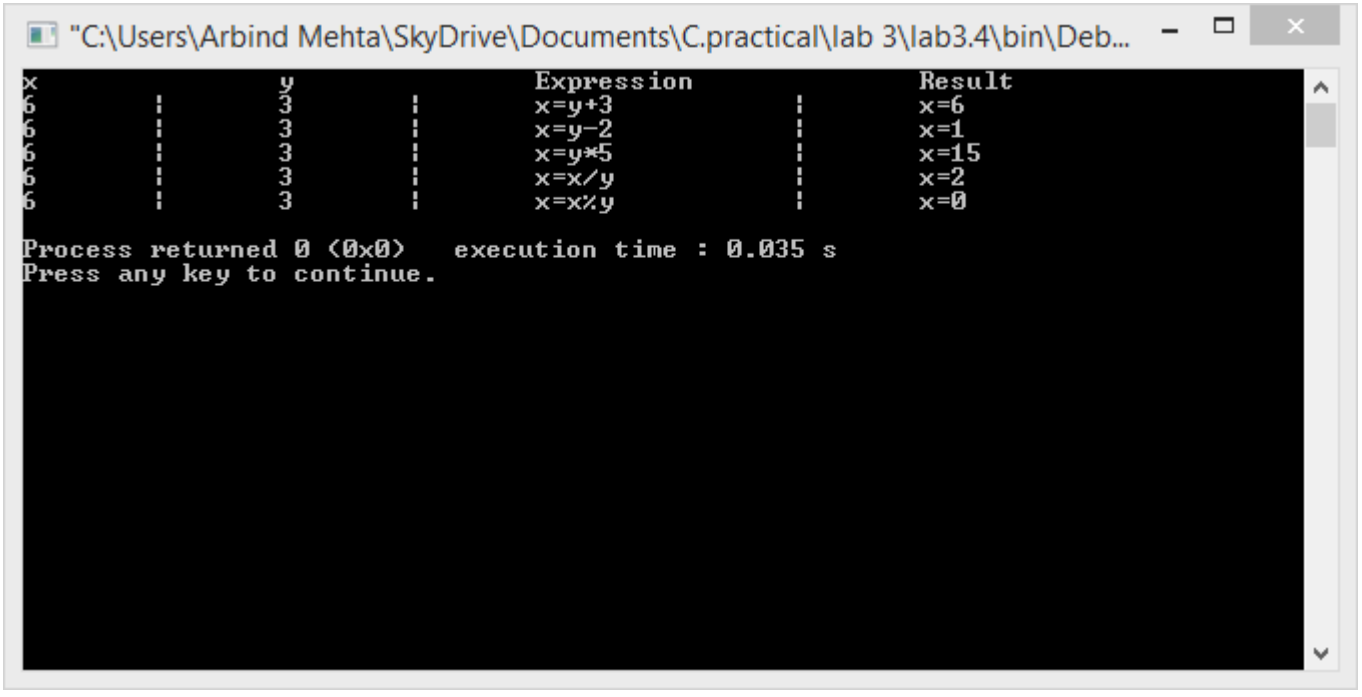


Code:

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

```
int main()
{
   int x=6,y=3;
   printf("x\t\ty\t\tExpression\t\tResult\n");
   printf("%d\t|\t%d\t|\tx=y+3\t\t|\tx=%d\n",x,y,y+3);
   printf("%d\t|\t%d\t|\tx=y-2\t\t|\tx=%d\n",x,y,y-2);
   printf("%d\t|\t%d\t|\tx=y*5\t\t|\tx=%d\n",x,y,y*5);
   printf("%d\t|\t%d\t|\tx=x/y\t\t|\tx=%d\n",x,y,x/y);
   printf("%d\t|\t%d\t|\tx=x%%y\t\t|\tx=%d\n",x,y,x%y);
   return 0;
}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types, different operation, with formatted input/output in C.

TITLE:

Given x=3.0, y=12.5, z= 523.3, A=300.0, B=1200.5, C=5300.3, Write a program to display the following:

X y z= 3.0| 12.5| 523.3| A B C= 300.0| 1200.5| 5300.3|

X y z= |3.00 |12.50 |523.30 A B C= |300.00 |1200.50 |52300.30

Objective:

- To know different types of data types, operation,
- To be familier with formatted I/O in C.

Problem analysis:

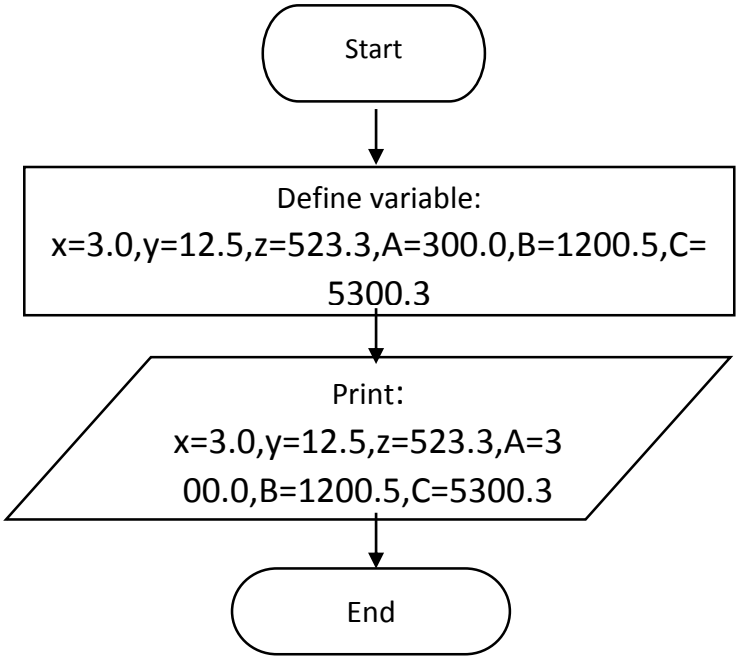
Based on given problem, our program must define six variables of type int. Different operation should be performed using formatted I/O.

Input variables	**Necessary header files/functions/macros**
x=3.0,y=12.5,z=523.3,A=300.0,B=1200.5,C=5300.3 (int type)	stdio.h coino.h printf()

Algorithm:

1. Start
2. Define variable: x=3.0,y=12.5,z=523.3,A=300.0,B=1200.5,C=5300.3
3. Display values using printf() function
4. Stop.

Flowchart:

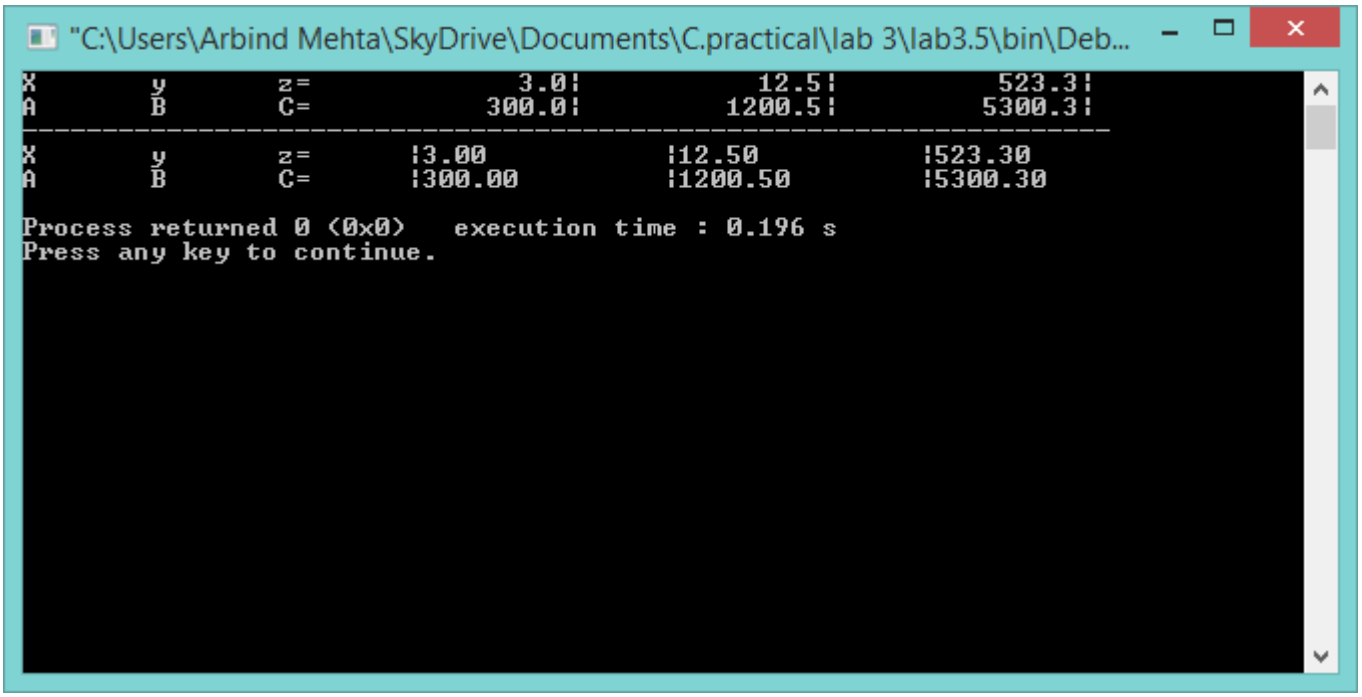


Code:

```
#include <stdio.h>


int main()
{
   float x=3.0,y=12.5,z=523.3,A=300.0,B=1200.5,C=5300.3;

   printf("X\ty\tz=\t%10.1f|\t%10.1f|\t%10.1f|\n",x,y,z);
   printf("A\tB\tC=\t%10.1f|\t%10.1f|\t%10.1f|\n",A,B,C);
   printf("----------------------------------------------------------------\n");
   printf("X\ty\tz=\t|%-10.2f\t|%-10.2f\t|%-10.2f\n",x,y,z);
   printf("A\tB\tC=\t|%-10.2f\t|%-10.2f\t|%-10.2f\n",A,B,C);

   return 0;
```

}

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types, different operation, with formatted input/output in C.

TITLE:

Given the three numbers a(=8), b(=4),c and constant value PI=3.1415, calculate and display the following result using macros (preprocessor directives)

a. c = PI * mult(a,b) //the macro mult(a,b) perform the multiplication of a & b(a*b)

b. c= PI* sum(a,b) //the macro mult(a,b) perform the sum of a & b (a+b)

c. c= PI *sub(a,b) //the macro mult(a,b) perform the subtraction of a & b (a-b)

d. c= PI*div(a,b) //the macro mult(a,b) perform the division of a & b (a/b)

Objective:

- To know different types of data types, operation,
- To be familier with formatted I/O, macros, preprocessor directives in C.

Problem analysis:

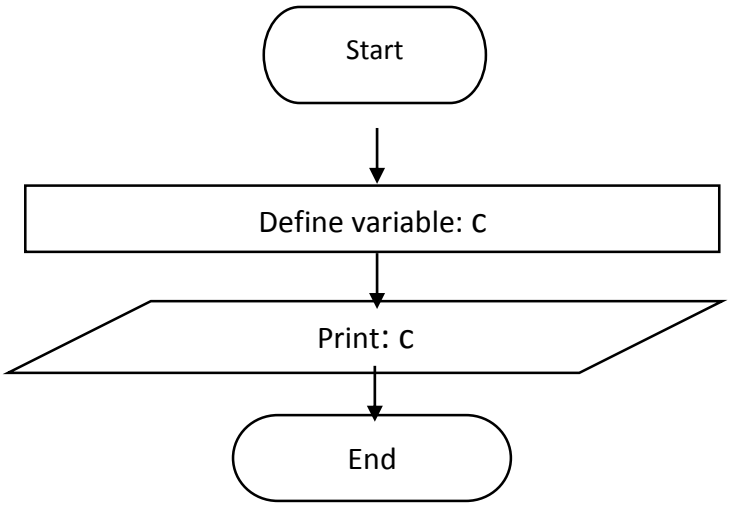
Based on given problem, our program must define two variables of type int and one of type float and define pi=3.1415 as preprocessor directives. Different operation should be performed using formatted I/O.

Input variables	**Necessary header files/functions/macros**
c(float type)	stdio.h coino.h printf() #define pi 3.1415 #define a 8 #define b 4 #define sum (a+b) #define sub (a-b) #define mult (a*b) #define div (a/b)

Algorithm:

1. Start
2. Define variable:c
3. Display values using printf() function
4. Stop.

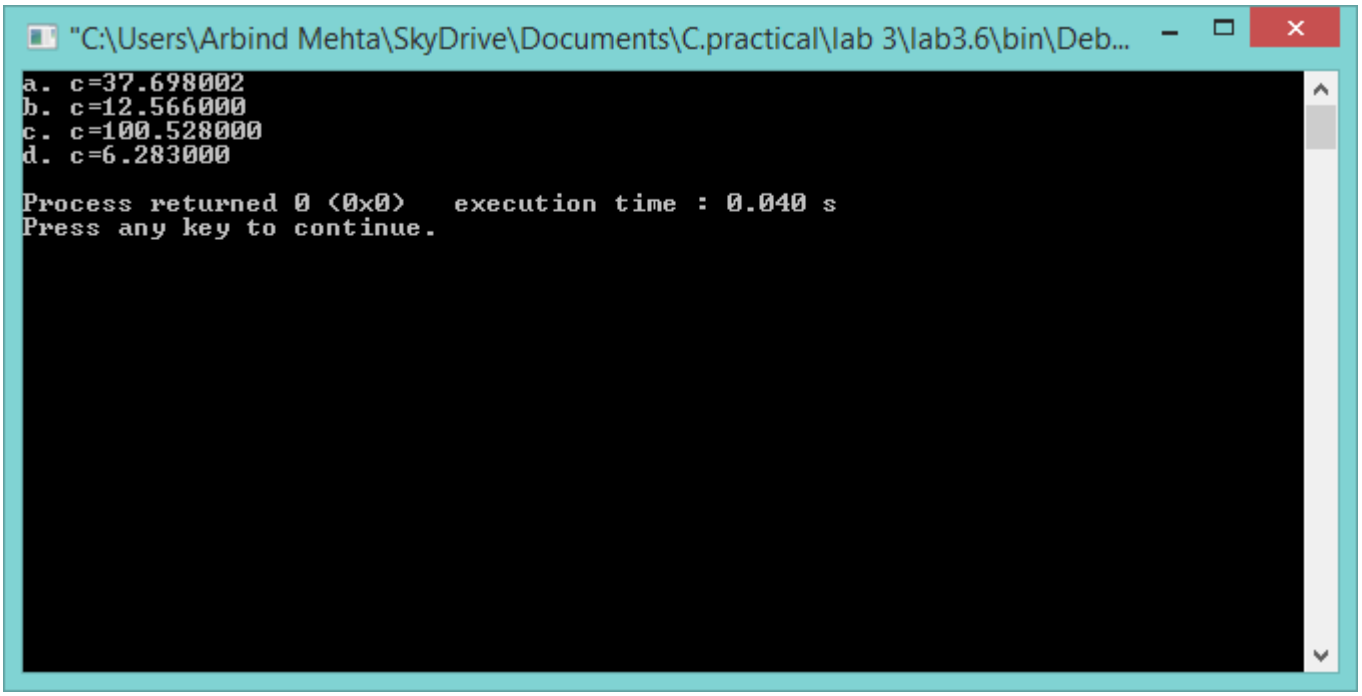
Flowchart:



Code:

```
#include <stdio.h>
#define pi 3.1415
#define a 8
#define b 4
#define sum (a+b)
#define sub (a-b)
#define mult (a*b)
#define div (a/b)

int main()
{
   float c;

   c=pi*sum;
   printf("a. c=%f\n",c);
   c=pi*sub;
   printf("b. c=%f\n",c);
   c=pi*mult;
   printf("c. c=%f\n",c);
   c=pi*div;
   printf("d. c=%f\n",c);
   return 0;
}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types, different operation, macros and preprocessor, with formatted input/output in C.

TITLE:

Write a program to take a character input from keyboard and check if it is a number or alphabet or special character using ASCII CODE. Again check if the character is using character functions below:

a. Alphanumeric => isalnum()

b. Blank character => isblank()

c. Alphabetic => isalpha()

d. Control character => iscntrl()

e. Number-digit => isdigit()

f. Upper case => isupper()

g. Lower case => islower()

h. Hexadecimal digit => ixdigit()

i. Graphical character => isgraph()

Objective:

- ❖ To know different types of data types, operation,
- ❖ To be familier with different inbuilt function, conditional operator in C.

Problem analysis:

Based on given problem, our program must define one variables of type char and one of int type to hold ASCII value of given character. Different operation should be performed using If and conditional operator.

Input variables	**Necessary header files/functions/macros**
ch (char type) a(int type)	stdio.h coino.h #include<ctype.h> #include<stdlib.h>

Algorithm:

1. Start
2. Define variable: ch, a
3. Read variables ch using getchar() function
4. if((a>=48&&a<=57)||(a>=65&&a<=90)||(a>=97&&a<=122))

```
        {
        if((a>=65&&a<=90)||(a>=97&&a<=122))
                printf("%c is an alphabet.\n",ch);
        else
                printf("%c is a number.\n",ch);
    }
    else
    printf("%c is a special character.\n",ch);

    (isalnum(ch)>0)?printf("%c is an alphanumeric.\n",ch):EXIT_SUCCESS;
     (isalpha(ch)>0)?printf("%c is an alphabetic.\n",ch):EXIT_SUCCESS;
```

(isblank(ch)>0)?printf("%c is Blank charactar.\n",ch):EXIT_SUCCESS;

(iscntrl(ch)>0)?printf("%c is control character.\n",ch):EXIT_SUCCESS;

(isdigit(ch)>0)?printf("%c is Number-digit.\n",ch):EXIT_SUCCESS;

(isupper(ch)>0)?printf("%c is Upper case.\n",ch):EXIT_SUCCESS;

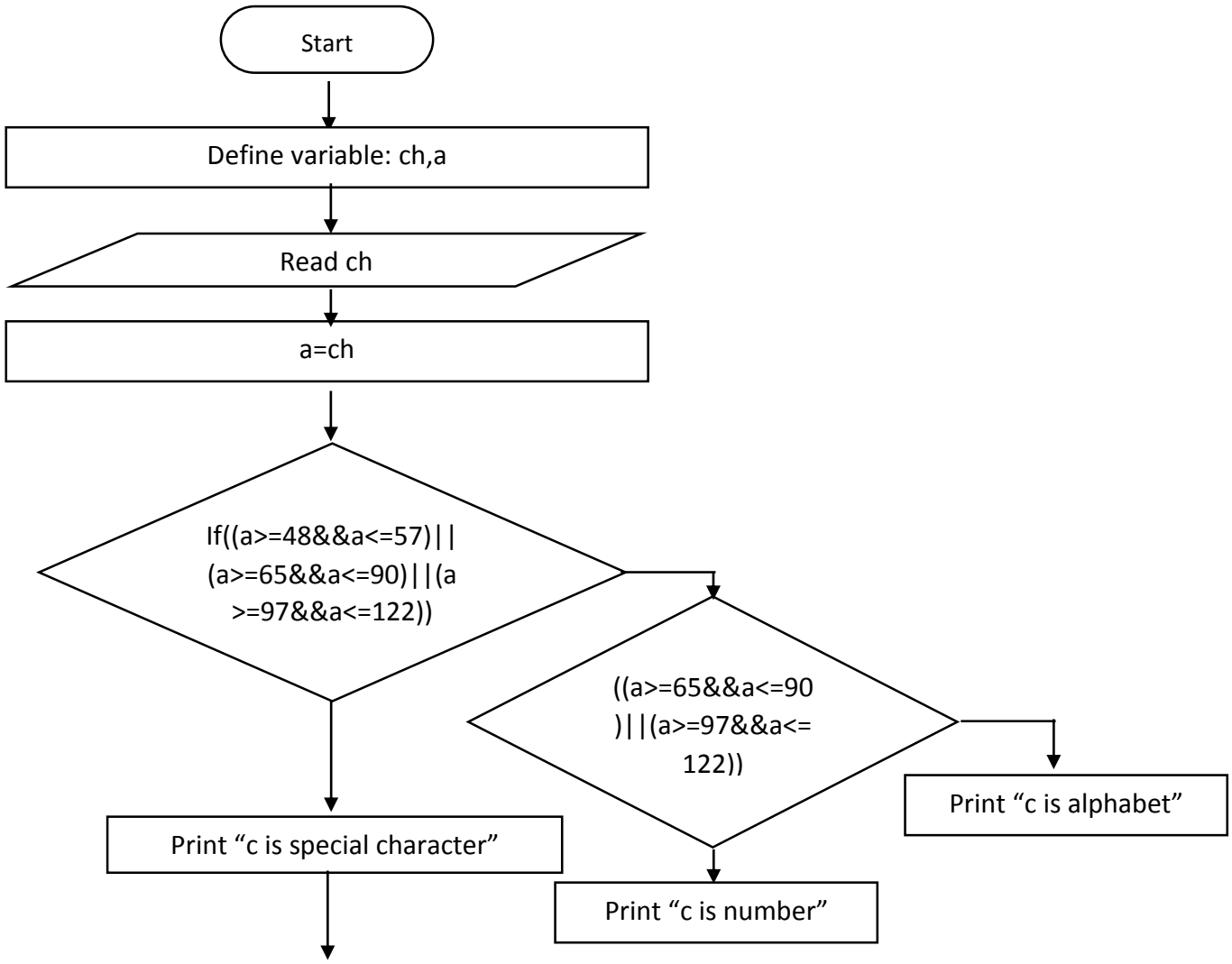
(islower(ch)>0)?printf("%c is Lower case.\n",ch):EXIT_SUCCESS;

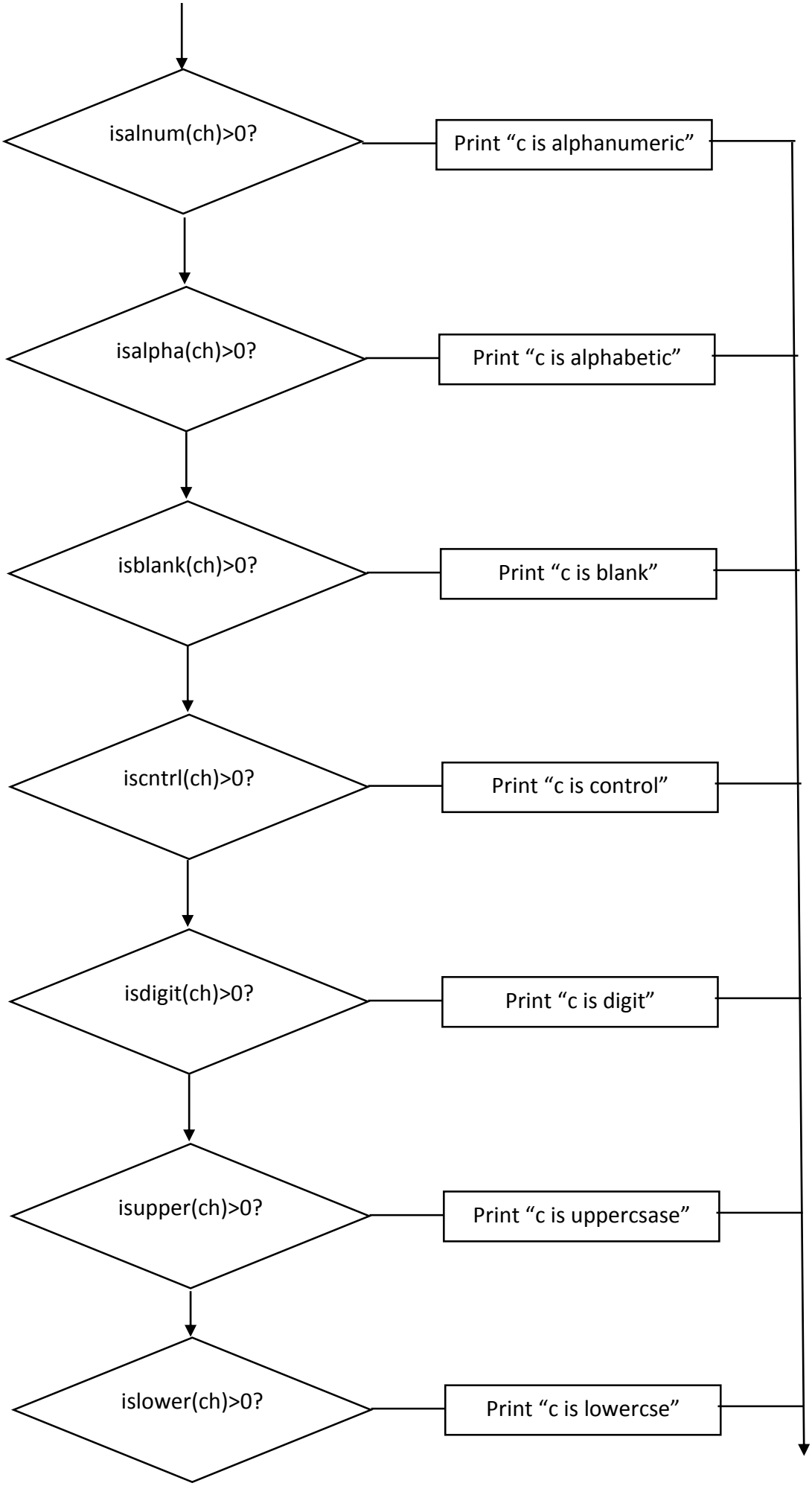
(isxdigit(ch)>0)?printf("%c is Hechadecimal digit.\n",ch):EXIT_SUCCESS;

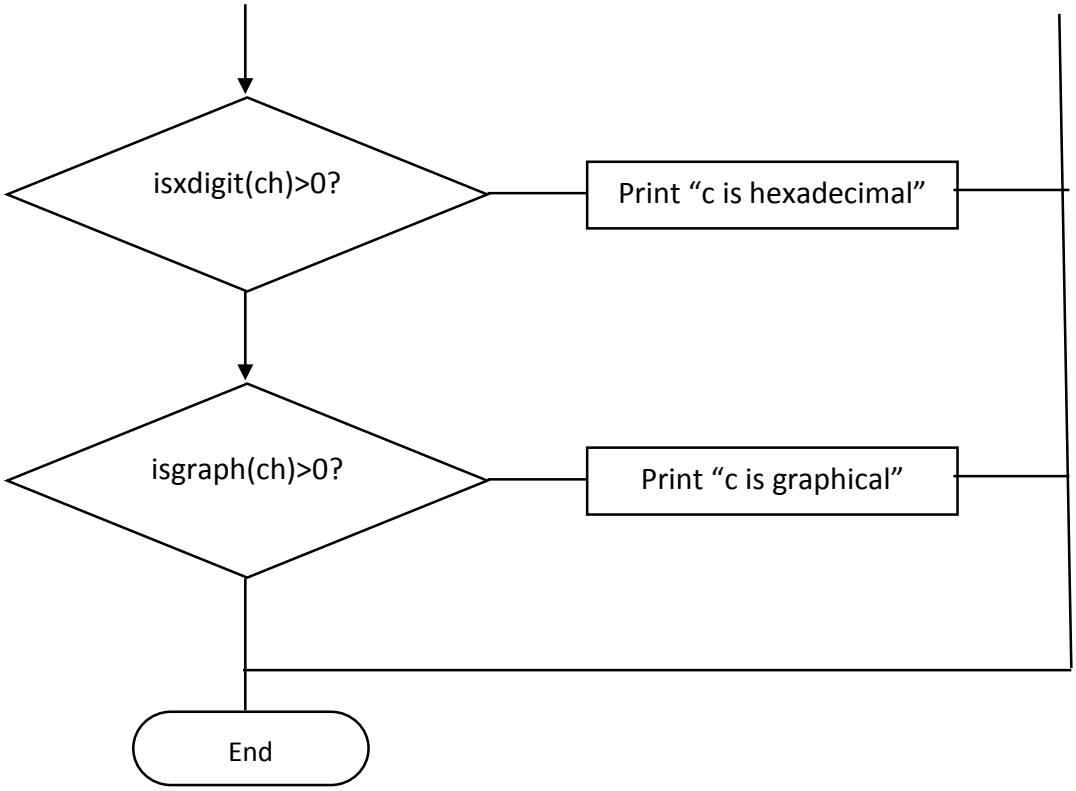
(isgraph(ch)>0)?printf("%c is Graphical character.\n",ch):EXIT_SUCCESS;

5. Stop.

Flowchart:




isalnum(ch)>0?
Print “c is alphanumeric”
isalpha(ch)>0?
Print “c is alphabetic”
isblank(ch)>0?
Print “c is blank”
iscntrl(ch)>0?
Print “c is control”
isdigit(ch)>0?
Print “c is digit”
isupper(ch)>0?
Print “c is uppercsase”
islower(ch)>0?
Print “c is lowercse”

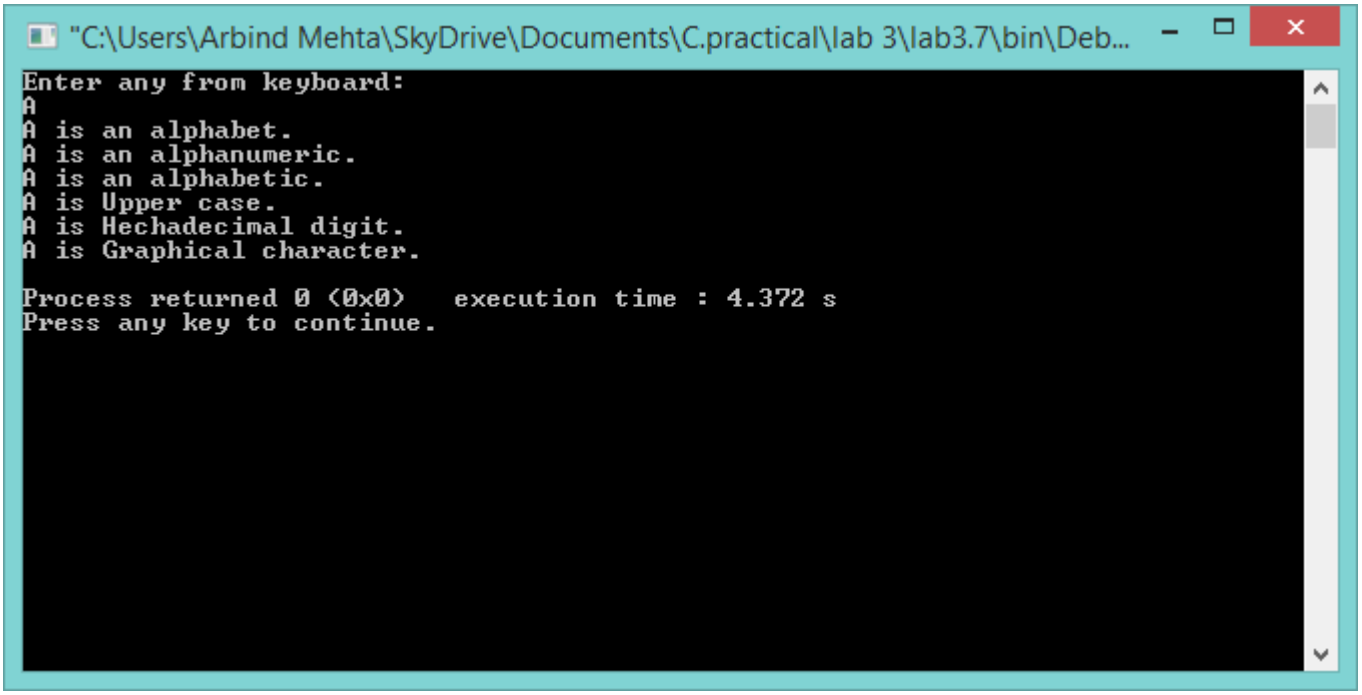


Code:

```
#include <stdio.h>
#include<ctype.h>
#include<stdlib.h>
int main()
char ch;
   int a;
   printf("Enter any from keyboard:\n");
   ch=getchar();
   a=ch;
   if((a>=48&&a<=57)||(a>=65&&a<=90)||(a>=97&&a<=122))    //comparing ASCII code of various characters
   {
      if((a>=65&&a<=90)||(a>=97&&a<=122))
      printf("%c is an alphabet.\n",ch);
 else
```

```
 printf("%c is a number.\n",ch);
   }
   else
   printf("%c is a special character.\n",ch);

   (isalnum(ch)>0)?printf("%c is an alphanumeric.\n",ch):EXIT_SUCCESS;
//EXIT_SUCCESS=>sucessfull termination
   (isalpha(ch)>0)?printf("%c is an alphabetic.\n",ch):EXIT_SUCCESS;
   (isblank(ch)>0)?printf("%c is Blank charactar.\n",ch):EXIT_SUCCESS;
   (iscntrl(ch)>0)?printf("%c is control character.\n",ch):EXIT_SUCCESS;
   (isdigit(ch)>0)?printf("%c is Number-digit.\n",ch):EXIT_SUCCESS;
   (isupper(ch)>0)?printf("%c is Upper case.\n",ch):EXIT_SUCCESS;
   (islower(ch)>0)?printf("%c is Lower case.\n",ch):EXIT_SUCCESS;
   (isxdigit(ch)>0)?printf("%c is Hechadecimal digit.\n",ch):EXIT_SUCCESS;
   (isgraph(ch)>0)?printf("%c is Graphical character.\n",ch):EXIT_SUCCESS;
   return 0;
}
```

Output (Compilation, Debugging and Testing):



Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types, different operation, different inbuilt function, conditional operator , with formatted input/output in C.

TITLE:

Demonstrate the differences among getch(), getche(), getchar(). Demonstrate the difference between scanf() & gets(), printf() & puts().

Objective:

- To know different types of data types, operation,
- To be familier with formatted and unformatted I/O in C.

Problem analysis:

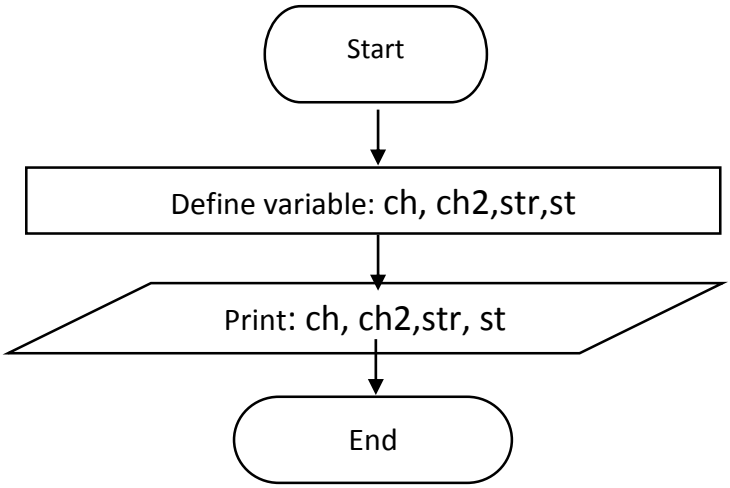
Based on given problem, our program must define four variables of type char. Different operation should be performed using formatted and unformatted I/O.

Input variables	**Necessary header files/functions/macros**
ch,ch2,str[10],st[10] (char type)	stdio.h coino.h printf() scanf() getch(), getche(), getchar() gets(), puts()

Algorithm:

1. Start
2. Define variable: ch, ch2, str[10], st[10]
3. Read value using gets(), getchar(), getc(), scanf() function
4. Display values using printf() , puts(), putc(), putchar() function
5. Stop.

Flowchart:

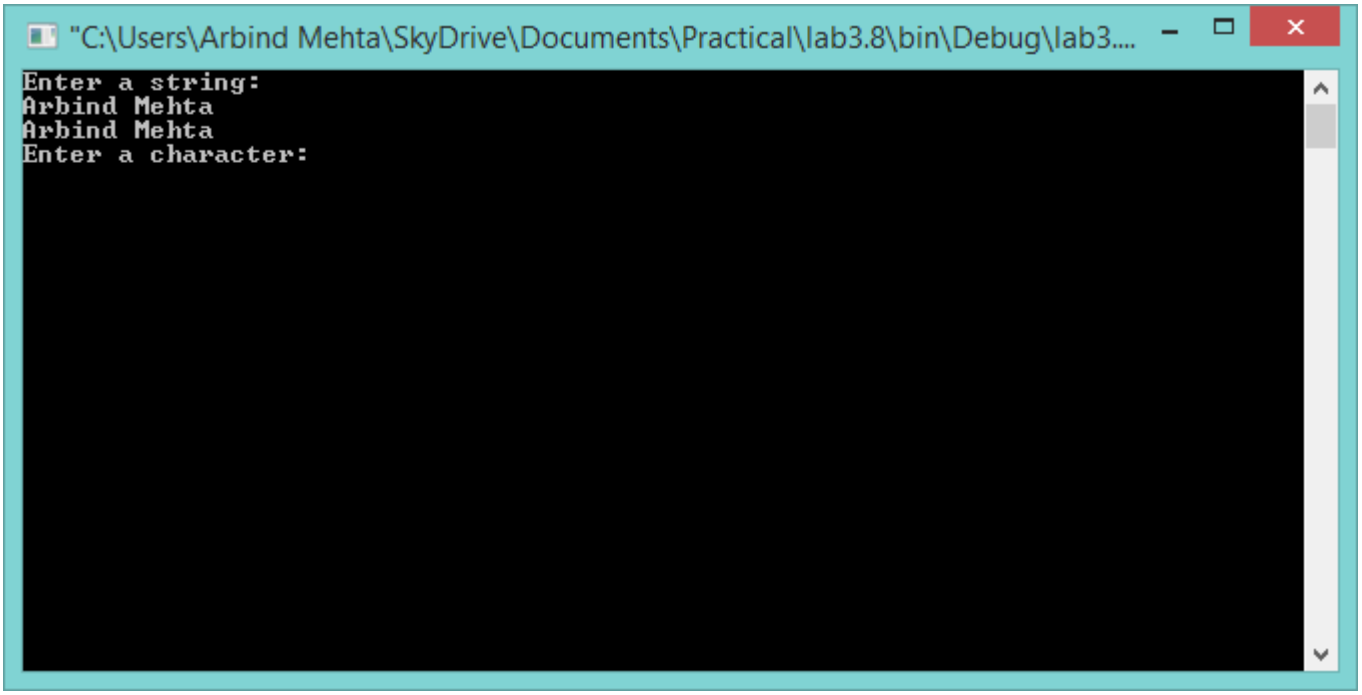


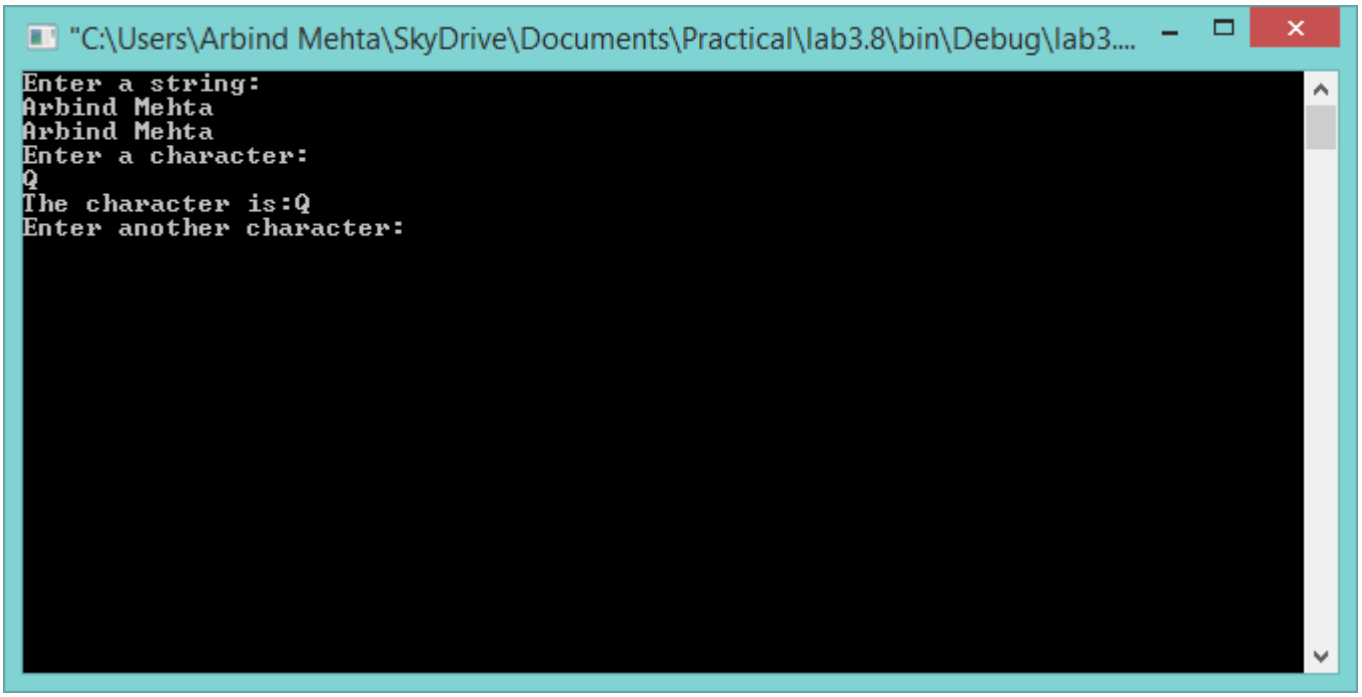
Code:

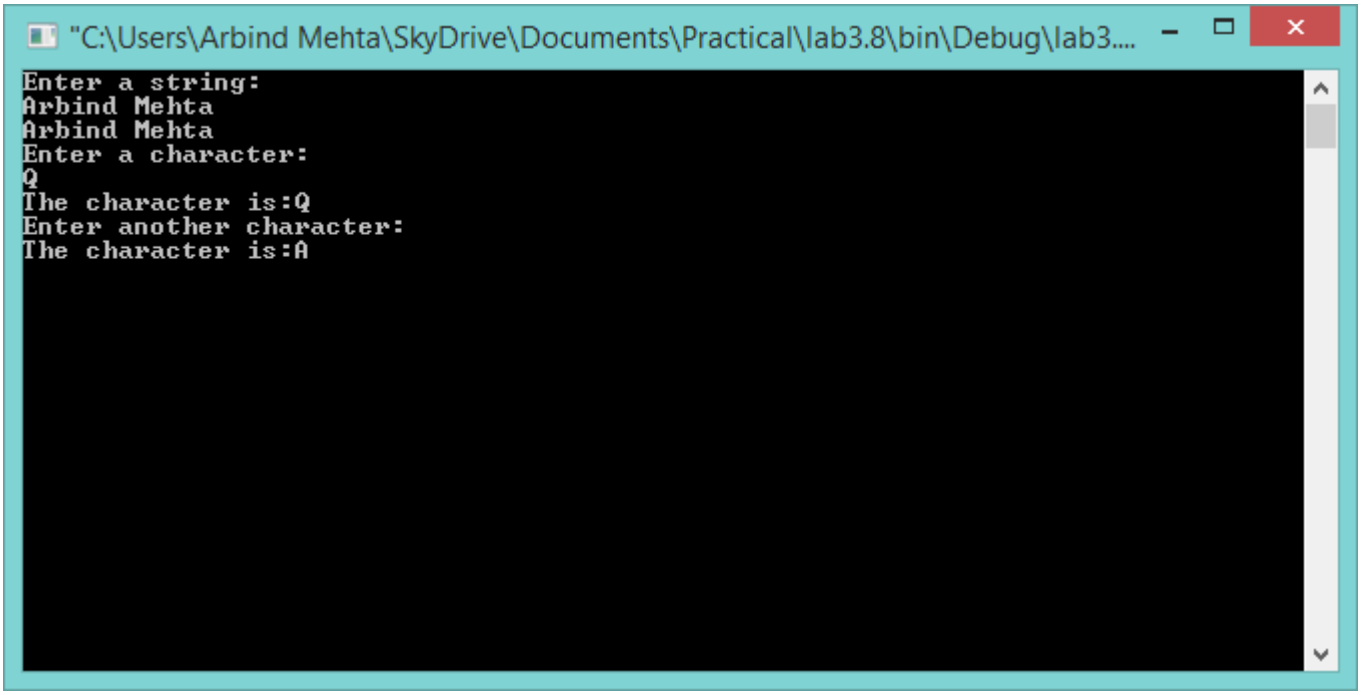
```
#include <stdio.h>
#include<stdlib.h>

int main()
{
   char ch,ch2,str[10],st[10];

   puts("Enter a string:");
   gets(st);
   puts(st);
   printf("Enter a character:\n"); //using printf()
   ch=getchar();  //using getchar()
   printf("The character is:");
   putchar(ch);  //using puts()
   puts("\nEnter another character:");
   ch2=getch();  //using getc()
```

```
    printf("The character is:%c",ch2);
    getch();
  return 0;
}
```

Output (Compilation, Debugging and Testing):







Discussion & Conclusion:

In this lab of C programming, based on the focused objective(s) to understand about C data types, different operation, difference between printf(), puts(), gets(), getchar(), getch(), scanf() functions, with formatted input/output in C.

***