**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **DATE** | **Name of the Experiment** | **Page. No** |
| 01 | 29-09-2020 | Usage of Input/Output functions | 1 |
| 02 | 6-10-2020 | Usage of Operators | 16 |
| 03 | 13-10-2020 | Usage of Conditional Statements | 34 |
| 04 | 20-10-2020 | Usage of Loop Control Statements | 56 |
| 05 | 10-11-2020 | Usage of Arrays | 82 |
| 06 | 17-11-2020 | String Operations | 97 |
| 07 | 24-11-2020 | Usage of User Defined Functions | 107 |
| 08 | 1-12-2020 | Usage of Pointers | 117 |
| 09 | 8-12-2020 | Structures in C Programming | 128 |
| 10 | 12-12-2020 | File Handling | 138 |

|  |  |
| --- | --- |
| **Exp.No:1** | **Usage of Input/Output Functions** |
| **Date: 29-09-2020** |

**1)Aim:**

To write a C program to calculate distance travelled by a vehicle.

**Algorithm:**

**Step1**: Start.

**Step2**: Include required header files.

**Step3**: Declare the variable speed, time and distance.

**Step4**: Get the values for speed and time from the user.

**Step5**: Calculate Distance = speed \* time

**Step6**: print the result.

**Step7**: End.

**Flowchart:**

**START**

**READ SPEED AN**

**D**

**TIME**

|  |
| --- |
| **DISTANCE = SPEED \* TIME** |

**DISPLAY**

**STATEMENT**

**END**

**Program:**

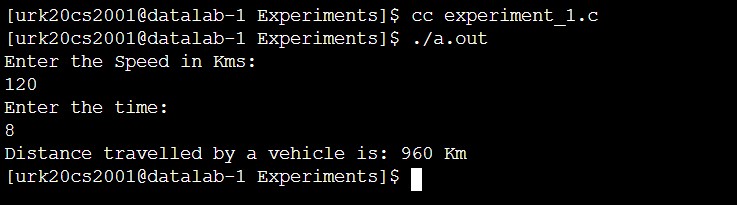
#include<stdio.h> int main()

{ int speed, time, distance; printf(“Enter the Speed in Kms: \n”); scanf(“%d”, &speed);

printf(“Enter the time: \n “); scanf(“%d”, &time);

distance = speed \* time; printf(“Distance travelled by a vehicle is: %d Km \n” , distance ); }

**Output:**



**2)Aim:**

To write a C program to print the size of the built-in datatypes using the sizeof() operator.

**Algorithm:**

**Step1**: Start.

**Step2**: Include the required header files.

**Step3**: Declare the required variables.

**Step4**: Read the datatypes of the sizeof() operators.

**Step5**: Calculate the size of the given data type using sizeof().

**Step6**: Print sizeof() all the required data types.

**Step7**: End.

**Flowchart:**

**START**

**READ THE DATATYPES**

**INSIDE THE sizeof()**

**OPERATOR**

|  |
| --- |
| **CALCULATE THE SIZE OF THE GIVEN DATA TYPE USING sizeof()** |

**PRINT THE RESULTS**

**END**

**Program:**

#include<stdio.h> int main()

{

int int\_var; float float\_var; char char\_var;

int int\_array[5];

int size;

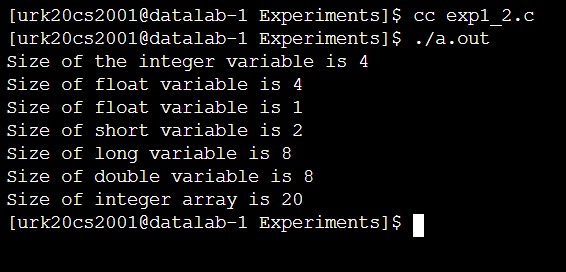
printf(“Size of the integer variable is %d \n”, sizeof(int\_var)); printf(“Size of float variable is %d \n”, sizeof(float\_var));

size = sizeof(char\_var);

printf(“Size of float variable is %d \n”, sizeof(char\_var)); printf(“Size of short variable is %d \n”, sizeof(short)); printf(“Size of long variable is %d \n”, sizeof(long)); printf(“Size of double variable is %d \n”, sizeof(double)); printf(“Size of integer array is %d \n”, sizeof(int\_array));

}

**Output:**



**3)Aim:**

To write a C program to calculate the area of a triangle.

**Algorithm:**

**Step1**: Start.

**Step2**: Include the required header files.

**Step3**: Declare the variables.

**Step4**: Get the values from the user.

**Step5**: Calculate the area of the triangle.

**Step6**: Print the results.

**Step7**: End.

**Flowchart:**

**START**

**READ THE VALUES**

|  |
| --- |
| **CALCULATE AREA** |

**PRINT THE RESULTS**

**END**

**Program:**

#include<stdio.h> #include<math.h> int main(void)

{

double a,b,c,s,area;

printf(“Enter the value of a: \n”); scanf(“%lf”, &a);

printf(“Enter the value of b: \n”); scanf(“%lf”, &b);

printf(“Enter the value of c: \n”); scanf(“%lf”, &c);

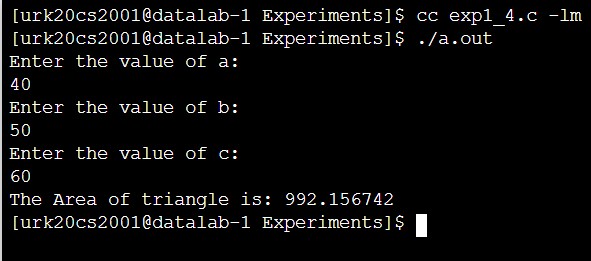
s = (a+b+c)/2;

area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

printf(“The Area of triangle is: %lf \n”, area);

}

**Output:**



**4)Aim:**

To write a C program to print your student profile.

**Algorithm:**

**Step1**: Start.

**Step2**: Include the required header files.

**Step3**: Declare the variable regno, name, age and Dob.

**Step4**: Get the Name, Reg No, age and Dob.

**Step5**: Read the Name, RegNO, age and Dob from the user.

**Step6**: Print the result.

**Step7**: End

**Flowchart:**

**GET THE NAME,**

**REGNO, AGE AND DOB**

**FROM THE USER**

**START**

|  |
| --- |
| **READ THE NAME, REG NO, AGE AND DOB FROM THE USER** |

**PRINT THE RESULTS**

**END**

**Program:**

#include<stdio.h> int main()

{

char regno[10]; char name[10]; int age; int dob;

printf(“Enter your Register Number: \n”); scanf(“%s”, &regno);

printf(“Enter your Name: \n”); scanf(“%s”, &name);

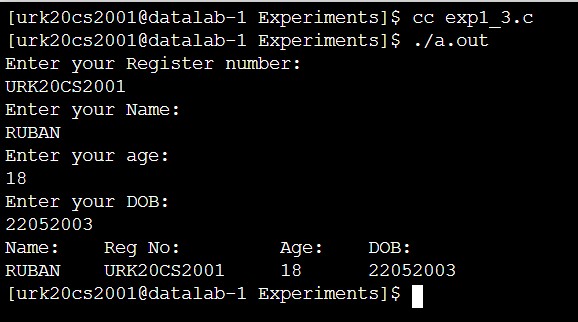
printf(“Enter your age: \n”); scanf(“%d”, &age);

printf(“Enter your DOB: \n”); scanf(“%d”, &dob);

printf(“Name: \t Regno: \t Age: \t DOB: \n); printf(“%s \t %s \t %d \t %d \n”, name, regno, age, dob);

}

**Output:**



**Result:**

Thus the program for experimenting input and output functions in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp.No: 2** | **Usage of Operators** |
| **Date:6-10-2020** |

**1)Aim:**

To write a c program to evaluate the expression (a+b) \* c / d (e-f).

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables a,b,c,d,e and f.

**Step4:** Get the input a,b,c,d,e and f from the user.

**Step5:** Calculate the expression.

**Step6:** Print the result.

**Step7:** End

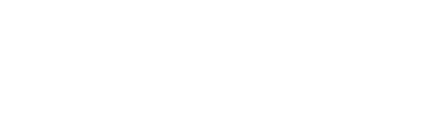
**Flowchart:**

**START**

**GET THE VALUES**

**OF a,b,c,d,e and**

**from the user.**



**CALCULATE TH**

**E**

**EXPRESSION**

**PRINT THE**

**RESULTS**

**END**

**Program:**

#include<stdio.h> int main()

{

int a,b,c,d,e,f,x; printf(“Enter the value of a: \n”); scanf(“%d”, &a);

printf(“Enter the value of b: \n”); scanf(“%d”, &b);

printf(“Enter the value of c: \n”); scanf(“%d”, &c);

printf(“Enter the value of d: \n”); scanf(“%d”, &d);

printf(“Enter the value of e: \n”); scanf(“%d”, &e);

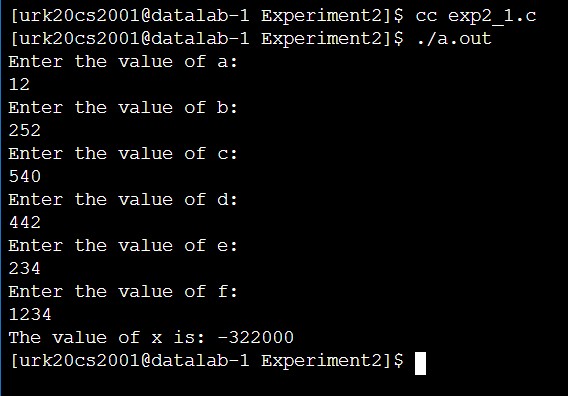
printf(“Enter the value of f: \n”); scanf(“%d”, &f);

x = ( a+b) \* c / d \* (e-f);

printf(“The value of x is: %d \n”, x);

}

**Output:**



**2)Aim:**

To write a C program to calculate the Body Mass Index.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the required variables.

**Step4:** Get the input height and weight from the user.

**Step5:** Calculate the Body Mass Index using the formula

**Step6:** Print the results.

**Step7:** End.

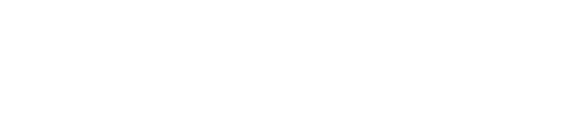
**Flowchart:**

**START**

**READ THE**

**HEIGHT AND**

**WEIGHT**



**CALCULATE**

**BMI=Weight(kg)/Height(m)^2**

**DISPLAY THE**

**STATEMENT**

**END**

**Program:**

#include<stdio.h> int main()

{

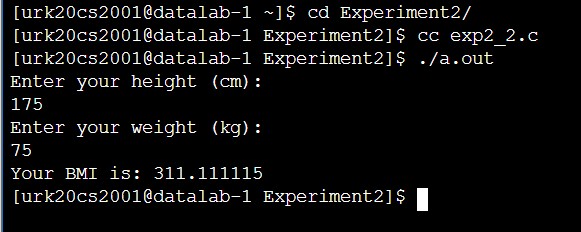
float height, weight, BMI, h; printf(“Enter your height (cm): \n”); scanf(“%f”, &weight);

printf(“Enter your weight (kg): \n”); scanf(“%f”, &height);

h = height / 100 \* height / 100; BMI = weight / h; printf(“Your BMI is: %f \n”, BMI);

}

**Output:**



**3)Aim:**

To write a C program to perform various arithmetic operations.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables

**Step4:** Read the Input from the user.

**Step5:** Calculate the arithmetic operations.

**Step6:** Print the steps.

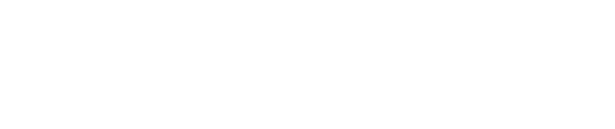
**Step7:** End.

**Flowchart:**

**START**

**GET THE INPUT**

**FROM THE USER**



**CALCULATE THE ARITHMETIC**

**EXPRESSIONS**

**PRINT THE**

**RESULTS**

**END**

**Program:**

#include<stdio.h> int main()

{

int A,S,M,D,Mod,a,b;

printf(“Enter the first number a: \n”); scanf(“%d”, &a);

printf(“Enter the second number b: \n”); scanf(“%d”, &b);

A = a + b;

S = a – b;

M = a \* b;

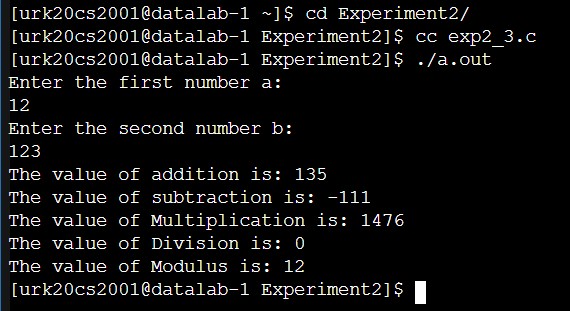
D = a / b;

Mod = a % b;

printf("The value of Addition is: %d \n", A); printf("The value of Subtraction is: %d \n", S); printf("The value of Multiplication is: %d \n", M); printf("The value of Division is: %d \n", D); printf("The value of Modulus is: %d \n", Mod);

}

**Output:**



**4)Aim:**

To write a C program to find the roots of a quadratic equation.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the required variables.

**Step4:** Calculate the equation.

**Step4:** Print the results.

**Step5:**  End.

**Flowchart:**

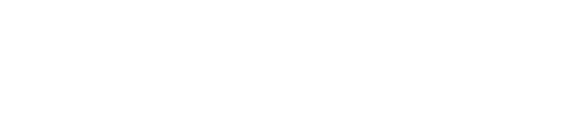
**START**

**GET THE INPUT**

**VALUE OF a,b,c**

**FRO**

**M THE USER**



**CALCULATING root1 and**

**root2.**

**DISPLAY**

**STATEMENT**

**END**

**Program:**

#include<stdio.h> #include<math.h> int main() { int a,b,c,root1,root2;

printf("Enter the value of a: \n"); scanf("%d", &a);

printf("Enter the value of b: \n"); scanf("%d", &b);

printf("Enter the value of c: \n"); scanf("%d", &c);

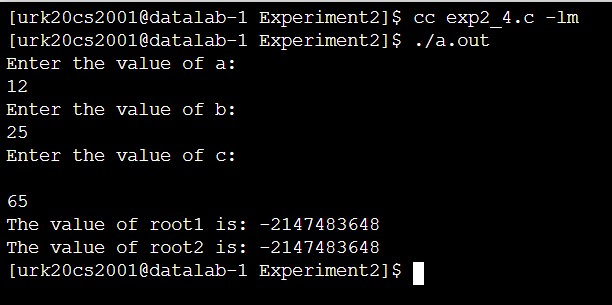
root1 = -b+(sqrt(b^2-4\*a\*c))/2\*a;

root2 = -b-(sqrt(b^2-4\*a\*c))/2\*a;

printf("The value of root1 is: %d \n", root1); printf("The value of root2 is: %d \n", root2);

}

**Output:**



**5)Aim:**

To write a C program to check the number is positive or negative and even or odd

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variable number1 and number2.

**Step4:** Check the numbers

**Step5:** Print the result

**Step6:** End.

**Flow**

**chart:**

Yes

No

Yes

No

Start

R

ead

number1

Display

statement

E

nd

N

umber1<0

Display number 1

is negative

Display number 1

is positive

Number1%=

=0

Display number 1

is negative

Display number 1

is positive

**Program:**

#include<stdio.h> int main() {

int number1, number2;

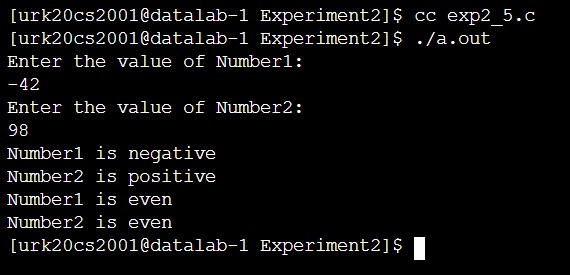
printf("Enter the value of Number1: \n"); scanf("%d", &number1);

printf("Enter the value of Number2: \n"); scanf("%d", &number2);

(number1>0)?printf("Number1 is positive \n") : printf("Number1 is negative \n"); (number2>0)?printf("Number2 is positive \n") : printf("Number2 is negative \n");

(number1%2==0)? printf("Number1 is even \n") : printf("Number1 is odd\n"); (number2%2==0)? printf("Number2 is even \n") : printf("Number2 is odd\n"); }

**Output:**



|  |  |
| --- | --- |
| **Exp.No: 3** | **Usage of Conditional Statements** |
| **Date: 13-10-2020** |

**1) Aim:**

To write a C program to implement a ticket booking application.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

`

**Step3:** Declare the variables age and price.

**Step4:** Get the input from the user.

**Step5:** Checking the condition.

**Step6:** Display the Statements.

**Step7:** End.

**Flowchart:**

TRUE

FALSE

START

age < 6 ||

age >= 60

price=price\*0.3

END

**Program:**

#include<stdio.h> int main() { char name[20]; int age, price = 600;

printf("Enter your name: "); scanf("%s", &name); printf("Enter your age: "); scanf("%d", &age);

if( age < 6 || age >= 60 )

{

price = price \* 0.3;

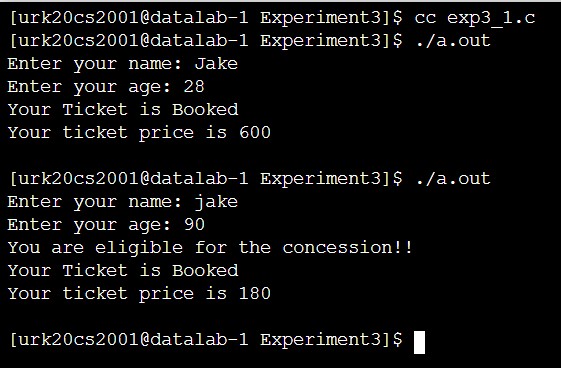
printf("You are eligible for the concession!! \n");

}

printf("Your Ticket is Booked \n"); printf("Your ticket price is %d \n\n", price);

}

**Output:**



**2) Aim:**

To write a C program to check the eligibility of a candidate to poll his vote.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variable age.

**Step4:** check the condition.

**Step5:** Display the Statement.

**Step6:** End.

**Flowchart:**

FALSE(else part)

TRUE (if part)

START

age >18

You are Eligible for voting

END

You are not eligible for

voting

**Program:**

#include<stdio.h> int main() { int age;

printf("Welcome to Indian Election Commision!!\n"); printf("\n->Enter your Age: "); scanf("%d", &age);

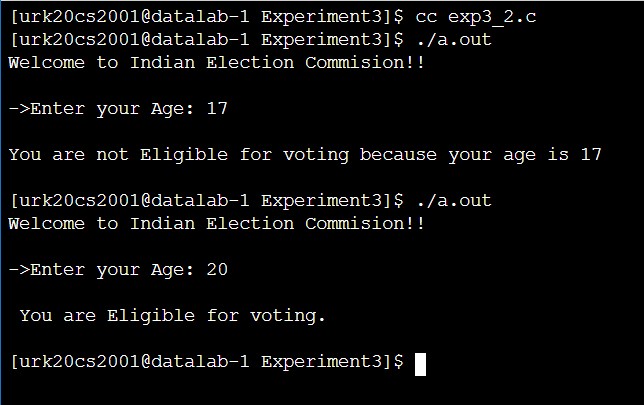
if(age > 18){

printf("\n You are Eligible for voting.\n\n");

} else

printf("\nYou are not Eligible for voting because your age is %d\n\n", age); }

**OUTPUT:**



**3) Aim:**

To write a C program to calculate the Body Mass Index.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables height, weight, BMI, h.

**Step4:**  Get the input from the user.

**Step5:** Calculate the BMI

**Step6:** Print the result.

**Step7:**  End.

**Flowchart:**

Start

Read the input

from the user

If

bmi<15

Print

Starvation

If bmi<18.6

to 24.9

If bmi<17.6

to 18.5

If bmi<15.1

to 17.5

If bmi<25

to 29.9

If bmi<30

to 39.9

Print

Aneorexic

Print Under

weight

Print Ideal

Print Over

weight

Print Obese

**Program**

**:**

#include<stdio.h>

int main()

{

int height, weight, BMI, h;

printf("Enter yo

ur height in (m): ");

scanf("%d", &height);

printf("Enter your weight in kg: ");

scanf("%d", &weight);

h = height/100\*height/100;

BMI = weight/h;

if(BMI < 15){

printf("

\

nStarvation

\

n

\

n");

}

else if(BMI

<= 17.5){

>=

15.1 && BMI

printf("

\

nAnorexic!!

\

n

\

n");

}

Print the result

Else

bmi>=40

End

Print Morbidly

obesese

else if(BMI >= 17.6 && BMI <= 18.5){ printf("\nUnder Weight\n\n");

}

else if(BMI >= 18.6 && BMI <= 24.9){ printf("\nIdeal\n\n");

}

else if(BMI >= 25 && BMI <= 29.9){ printf("\nOver Weight\n\n");

}

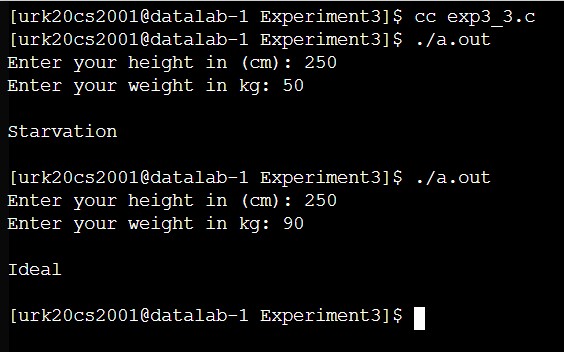
else if(BMI >= 30 && BMI <= 39.9){ printf("\nObese\n\n");

} else

printf("\nMorbidly obese\n\n");

}

**Output:**



**4)Aim:**

To write a C program to calculate the gross salary.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variable bs, hra, da, cv, gs.

**Step4:** Get the input salary from the user.

**Step5:** Calculating the basic salary.

**Step6:** Print the results.

**Step7:** End.

**Flowchart:**

Start

Read Salary

If

salary>=5000

If salary >=

&&

3000

salary<5000

0

If salary <

3000

DA=(salary)\*110/100

HRA=(salary)\*20/100

DA=(salary)\*110/100

HRA=(salary)\*15/100

DA=(salary)\*90/100

HRA=(salary)\*10/100

Print DA and HRA

End

**Program:**

#include <stdio.h> int main() {

float bs, hra, da, cv, gs; printf("Enter Basic Salary : "); scanf("%f", &bs);

if(bs >= 5000)

{ da = 110 \* bs / 100; hra = 20 \* bs / 100; cv = 500;

}

else if(bs >= 3000 && bs < 5000)

{ da = 100 \* bs / 100; hra = 15 \* bs / 100; cv = 400;

} else if( bs < 3000)

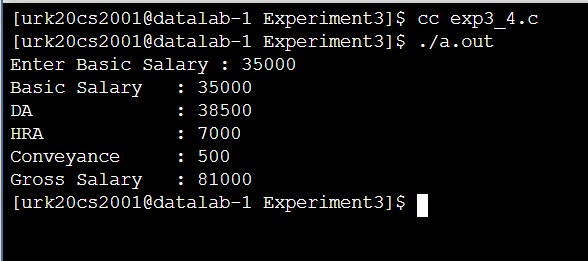
{ da = 90 \* bs / 100; hra = 10 \* bs / 100; cv = 300;

}

gs = bs + da + hra + cv; printf("Basic Salary : %.f \n",bs); printf("DA : %.f \n",da); printf("HRA : %.f \n",hra); printf("Conveyance : %.f \n",cv); printf("Gross Salary : %.f \n",gs); return 0;

}

**Output:**



**5)Aim:**

To write a C program to print a person is Boy/Men/Girl by checking their age using nested if.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required headerfiles.

**Step3:** Declare the variables a and b.

**Step4:** Get the input from the user.

**Step5:** Checking the conditions using nested if.

**Step6:** Print the result.

**Step7:** End.

**Flow chart:**

Start

Read Age and

Gender

If

gender=m&

age>25

Print “You are a Man”

Start

else

Print “You are a Boy”

If

gender=f&

age>20

else

Pr

int “You are a

woman”

Print “You are a

girl”

**Program:**

#include<stdio.h> int main() {

int a; char b;

printf("Enter your gender M for male and F for female: \t"); scanf("%c", &b); printf("\nEnter your age: \t"); scanf("%d", &a);

if (b=='M' || b=='m')

{ if (a <= 25)

{

printf("You are a BOY");

} else

{

printf("You are a MAN");

} }

else if (b=='F' || b=='f')

{ if (a<=20)

{

printf("You are a GIRL");

} else

{

printf("You are a WOMAN");

} } else

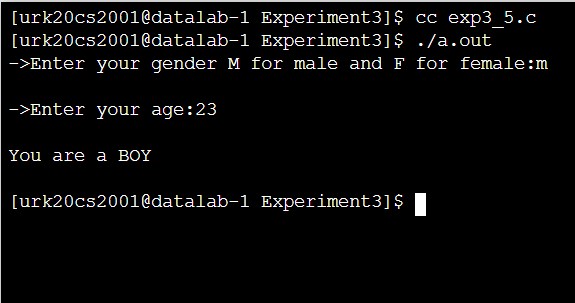
{

printf("OOPS YOUR INPUT IS WRONG!! TRY AGAIN WITH ANOTHER INPUT");

}

}

**Output:**



**6) Aim:**

To write a C program to create a basic arithmetic calculator using switch case statement.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variable a, b and c.

**Step4:** Get the input from the user.

**Step5:** Calculating the switch case statement.

**Step6:** End.

**Flowc**

**hart:**

Start

Read two

numbers

Choice = +

Num1+Num2

Choice

=

-

Choice

=

\*

Choice

=

/

Num1

-

Num2

Num1\*Num2

Num1/Num2

Print the result

End

**Program:**

#include<stdio.h> int main()

{

char a; int num1,num2,result=0;

printf("\nEnter your operation'+' '-' '\*' '/' \n"); scanf("%s",&a); printf("\nEnter num1 and num2\n"); scanf("%d""%d",&num1,&num2);

switch (a)

{ case '+':

printf("\nThe add value is::%d\n",num1+num2);

break;

case '-':

printf("\nThe add value is::%d\n",num1-num2);

break;

case '\*': printf("\nThe add value is::%d\n",num1\*num2);

break;

case '/':

printf("\nThe add value is::%d\n",num1/num2);

break;

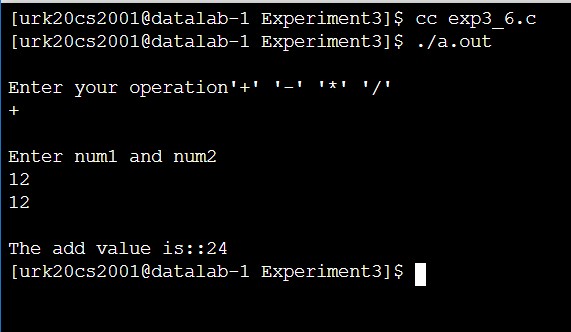
default:

printf("\nThe entered number is invalid\n");

}

}

**Output:**



**Result:**

Thus the program for experimenting Usage of Conditional Statement in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp. No: 4** | **Usage of Control Statements** |
| **Date: 20-10-2020** |

**1)Aim:**

To write a C program to print a pattern.

**Algorithm:**

**Step1:** Start.

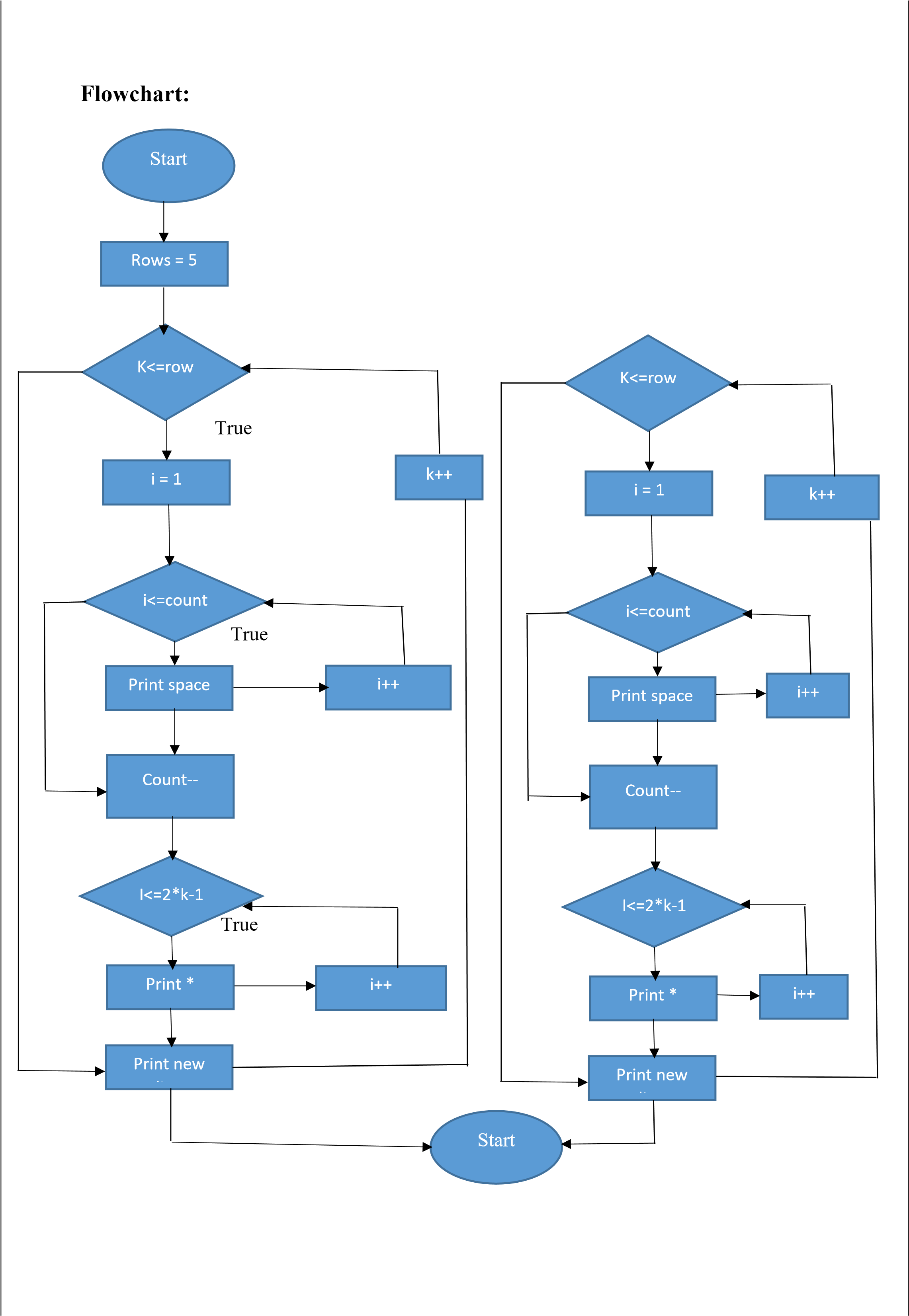
**Step2:** Include the required header files.

**Step3:** Declare the required variables.

**Step4:** Use for loop and while loop to check for the special condition and execute accordingly

**Step5:** print the result.

**Step6:** End.



**Program:**

#include<stdio.h> int main() { int i, k, row=5, count=1; count = row - 1;

for(k=1; k<=row; k++)

{ for(i=1; i<=count; i++)

{ printf(" "); } count--;

for(i=1; i<=2\*k-1; i++)

{ printf("\*"); } printf("\n");

}

count=1;

for(k=1; k<=row-1; k++)

{ for(i=1; i<=count; i++)

{ printf(" ");

}

count++;

for(i=1; i<=2\*(row-k)-1; i++)

{

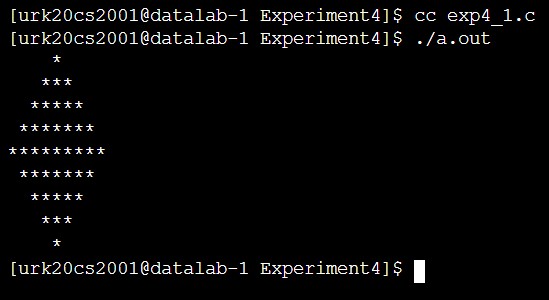
printf("\*");

} printf("\n");

} return 0;

}

**Out put:**



**2)Aim:**

To write a C program to print the multiplication table of a number and Get the number and range from the user.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables number, i, range.

**Step4:** Get the number and range as an input from the user.

**Step5:** calculating and looping the statement.

**Step6:** Print the result.

**Step7:** End.

**Flowchart:**

i++

Yes

No

Start

Read number

Read range

I=0 to

i<=range

Pr

int no\*i

Stop

**Program:**

#include<stdio.h> int main() {

int number, i, range;

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

printf("Enter the range: "); scanf("%d", &range);

for(i=1; i<=range; ++i)

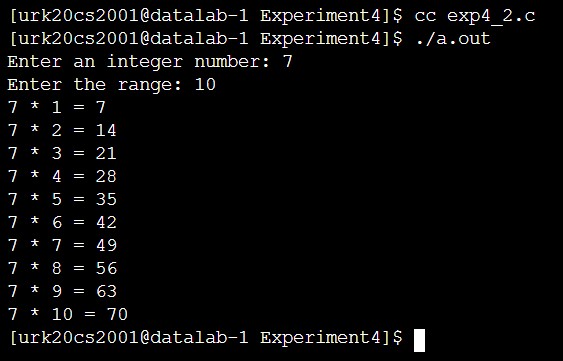
{

printf("%d \* %d = %d \n", number, i, number \* i);

} return 0;

}

**Output:**



**3)Aim:**

To write a C program to check the given number is Armstrong number or not.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables n, r, temp, sum=0;

**Step4:** Get the number as an input from the user.

**Step5:** Calculating the number using while loop and if else statement.

**Step6:** Print the result.

**Step7:** End.

**Flowchart:**

No

Yes

No

Yes

Start

Read n

Temp=n

n>0

Calculate r=n%10

Calculate sum=sum+(r\*r\*r)

Set n = n /10

Temp==n

Not an Armstrong Number

Not an Armstrong Number

End

**Program:**

#include<stdio.h> int main() { int n, r, temp, sum=0;

printf("Enter a Number: "); scanf("%d", &n);

temp = n;

while(n>0){ r = n % 10; sum = sum + (r \* r \* r); n = n / 10;

}

if(temp == sum)

{

printf("This is an Armstrong number\n");

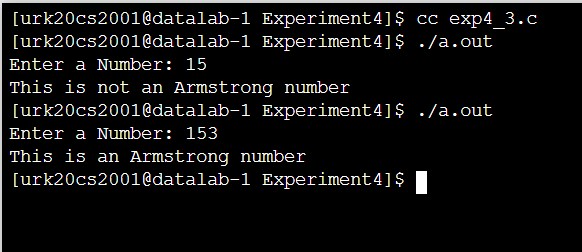
} else{

printf("This is not an Armstrong number \n");

}

}

**Output:**



**4)Aim:**

To write a C program to print the prime numbers between the given range.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables number1, number2, flag, temp, count=0, i, j;

**Step4:** Get the Input number1 and number2 from the user.

**Step5:** Checking the condition if the number is prime or not.

**Step6:** Print the result.

**Step7:** End.

**Flowchart:**

No

i++

Yes

No

Yes

No

No

Yes

Start

Read the numbers

i=0 to

i<=range

Flag = 0

i=2 to

j<=i/2

I%j==0

Flag = 1

Flag == 0

Print=i

Stop

j++

**Program:**

#include<stdio.h> #include<stdlib.h> void main() {

int number1, number2, flag, temp, count=0, i, j;

printf("Enter the value of Number1 and Number2: "); scanf("%d %d", &number1, &number2);

if(number2<2)

{

printf("There are no primes upto %d \n", number2);

}

printf("Prime Numbers are...\n"); temp = number1;

if(number1 % 2 == 0)

{

number1++;

}

for(i=number1; i<=number2; i=i+2)

{ flag = 0; for(j=2; j<=i/2; j++)

{ if((i%j)==0)

{ flag = 1; break;

} } if(flag == 0)

{

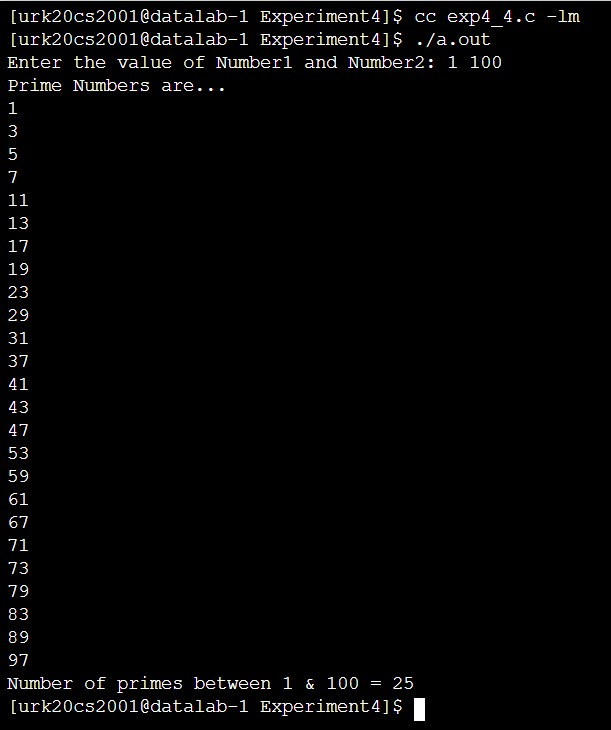
printf("%d \n", i); count++;

}

}

printf("Number of primes between %d & %d = %d \n", temp, number2, count); }

**Output:**



**5)Aim:**

To write a C program to read a password until it is correct. For wrong password print Incorrect, Allow only 3 incorrect password attempts and if it exceeds quit the program.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables enteredPassword, password = 1234.

**Step4:** Get the input as a password from the user.

**Step5:** Checking the condition using if else and for loop.

**Step6:** Print the result as a Display statement.

**Step7:** End.

Yes **Program:**

**Flowchart:**

No

i++

Yes

No

Start

Set i = 0

i <=2

Read Password

Password

= 1234

Print Incorrect

Password

Print correct

Password

End

#include<stdio.h> int main(void) {

int enteredPassword, password=1234;

int i = 0;

printf("Enter the password: "); scanf("%d", &enteredPassword);

if(password == enteredPassword)

{

printf("Login Successful \n ");

} else

{

printf("Login Failed!! Please Try again...\n");

}

for(i = 0; i <= 2; i++)

{

printf("Enter the password: "); scanf("%d", &enteredPassword);

if(password == enteredPassword)

{

printf("Login Successful \n");

}

else

{

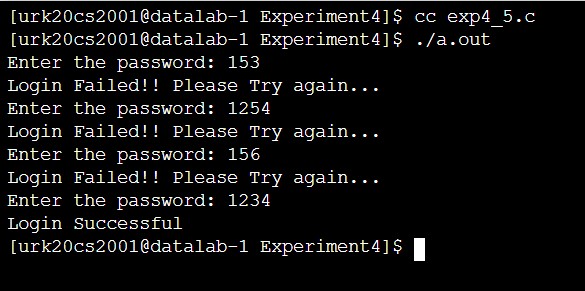
printf("Login Failed!! Please Try again...\n");

}

}

}

**Output:**



**6)Aim:**

To write a C Program to develop a basic arithmetic calculator with options.

**Algorithm:**

**Step1:** Start

**Step2:** Include the required header files.

**Step3:** Declare the variables a, b, c, d, e, z=y.

**Step4:** Get the values from the user.

**Step5:** calculating using switch case statement and while loop.

**Step6:** Print the result.

**Step7:** End.

**Flowchart:**

Yes

No

Yes

No

Start

Read a

Read b

Display Menu

Addition

1)

Subtraction

2)

Multiplication

3)

4)

Division

5)

Modulus

6)

Power

Read

option

case == 1

case == 2

Calculate a+b

Calculate a

-

b

Print a+b

Print a

-

b

Yes

No

Yes

No

Yes

N

o

Yes

No

case == 3

case == 4

case == 5

case == 6

Print Wrong Input

Calculate a\*b

Print a\*b

Calculate a/b

Print a/b

Calculate a%b

Print a%b

Calculate a^b

Print a+b

End

**Program:**

#include<stdio.h> void main() { int a,b,c,d; float e; char z='y'; do

{

printf("Enter the Values:\n"); scanf(" %d%d",&a,&b);

printf("MENU\n"); printf("1)Addition\n"); printf("2)Subtration\n"); printf("3)Multiplication\n"); printf("4)Division\n"); printf("5)Modulus\n"); printf(“6)Power\n”);

printf("Enter the choice:\n"); scanf(" %d",&d);

switch(d)

{ case 1:

printf("You have chosen Addition\n"); c=a+b;

printf("The result is : %d \n",c);

break;

case 2:

printf("You have chosen Subtraction \n"); c=a-b;

printf("The result is : %d \n",c); break;

case 3:

printf("You have chosen Multiplication\n"); c=a\*b;

printf("The result is : %d \n",c); break;

case 4:

printf("You have chosen Division\n"); e=(float) a/b; printf("The result is : %f \n",e); break; case 5:

printf("You have chosen Modulus\n"); c=a%b;

printf("The result is : %d \n",c); break;

case 6:

printf("You have chosen Power\n"); c=a^b;

printf("The result is : %d \n",c); break;

default:

printf("WRONG CHOICE....!! \n");

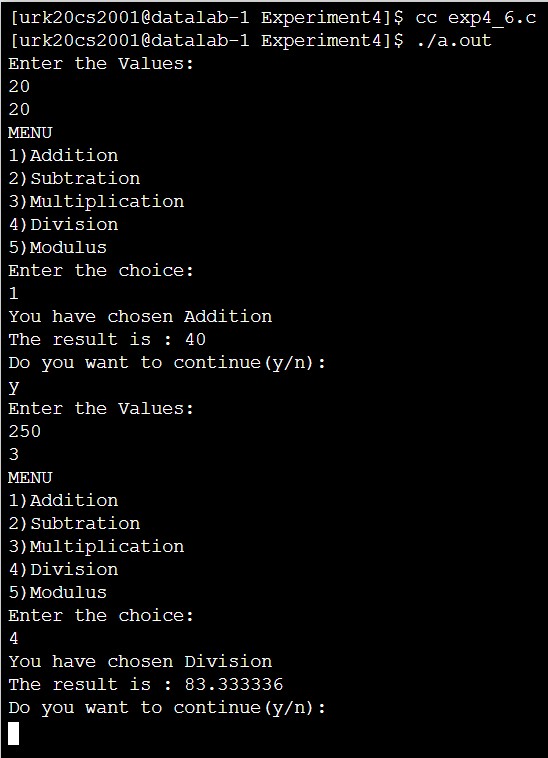
}

printf("Do you want to continue(y/n):\n"); scanf(" %c",&z);

}while( z=='y' || z=='Y'); switch(d);

}

**Output:**



**Result:**

Thus the program for experimenting Usage of conditional statement in C Programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp no: 5** | **Usage of Arrays** |
| **Date: 10-11-2020** |

**1)Aim:**

To write a C program to get 10 numbers from the user and find the largest and smallest number of the array and print it’s index value.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables a[10], i, minimum, index1, maximum, index2.

**Step4:** Calculating the values of array.

**Step5:** Print the results.

**Step6:** End.

**Program:**

#include<stdio.h> int main() { int a[10], i, minimum, index1, maximum, index2; printf("\n Enter 10 numbers: "); for(i=0; i<10; i++)

{ scanf("%d", &a[i]);

}

minimum = a[0]; maximum = a[0];

for(i=0; i<10; i++)

{

if(minimum > a[i])

{

minimum = a[i]; index1 = i;

}

if(maximum < a[i])

{

maximum = a[i]; index2 = i;

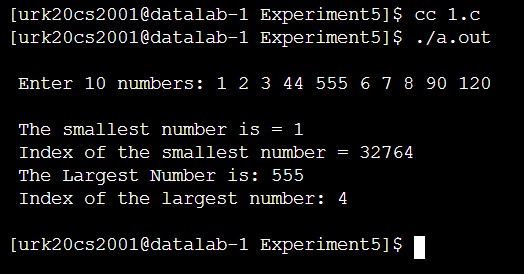
}

}

printf("\n The smallest number is = %d", minimum); printf("\n Index of the smallest number = %d", index1); printf("\n The Largest Number is: %d", maximum); printf("\n Index of the largest number: %d\n\n", index2); return 0;

}

**Output:**



**2)Aim:**

To write a C program to find their union and intersection for the given arrays.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables a[100], b[100], c[100], an, bn, s, i, j.

**Step4:** Calculating the Union and Intersection for the two given arrays.

**Step5:** Print the results.

**Step6:** End.

**Program:**

#include<stdio.h> void main()

{ int a[100], b[100], c[100], an, bn, s, i, j; printf("\n Enter the number of elements to be stored in A: "); scanf("%d", &an); printf("\n Enter the number of elements to be stored in B: "); scanf("%d", &bn);

for(i=0;i<an;i++)

{

printf("\n Enter the Elements of A at position %d:", i); scanf("%d", &a[i]);

}

for(j=0;j<bn;j++)

{

printf("\n Enter the Elements of B at position %d:", j); scanf("%d", &b[j]);

}

printf("\n Intersection between A and B: "); for(i=0;i<an;i++)

{ for(j=0;j<bn;j++)

{ if(a[i]==b[j]) printf("%d", a[i]);

}

}

printf("\n Union of A and B is: \n"); i=an; s=an+bn;

for(i=an,j=0;i<s;i++,j++)

{ a[i]=b[j]; }

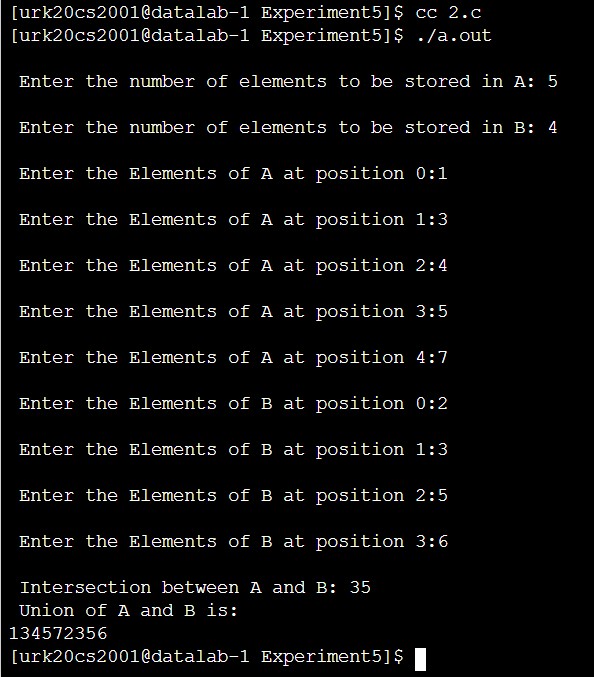
for(i=0;i<s;i++)

{ printf("%d", a[i]);

}

printf("\n"); }

**Output:**



**3)Aim:**

To write a C program to perform two dimensional array operations and perform the matrices to be added, subtraction and multiplication and perform the operations for two matrices.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables a[10][10], b[10][10], i, j, m, n, p, q;

**Step4:** Calculating the arithmetic operations of addition, subtraction, multiplication with the two matrices.

**Step5:** End.

**Program:**

#include<stdio.h> int main() { int a[10][10], b[10][10], i, j, m, n, p, q; printf("Enter the number of rows for the first matrix: "); scanf("%d", &m); printf("Enter the number of columns for first matrix: "); scanf("%d", &n);

printf("Enter the number of rows for second matrix: "); scanf("%d", &p); printf("Enter the number of columns for the second matrix: "); scanf("%d", &q);

if((m==p)&&(n==p))

{ printf("Enter the values for first matrix: "); for(i=0;i<m;i++)

{ for(j=0;j<n;j++)

{ scanf("%d", &a[i][j]);

}

}

printf("Enter the values for second matrix: "); for(i=0;i<p;i++)

{ for(j=0;j<q;j++)

{

scanf("%d", &b[i][j]);

}

}

printf("Addition of two matrices is: \n"); for(i=0;i<m;i++)

{ for(j=0;j<n;j++)

{ printf("%d\t", a[i][j]+b[i][j]);

} printf("\n");

}

printf("Subtraction of two matrices is: \n"); for(i=0;i<m;i++)

{ for(j=0;j<n;j++)

{ printf("%d\t", a[i][j]-b[i][j]);

} printf("\n");

}

}

else

{

printf("Matrix addition and subtraction are impossibe.\n"); } if(n==p)

{ printf("Multiplication of two matrices is: \n"); for(i=0;i<m;i++)

{ for(j=0;j<n;j++)

{ printf("%d\t", a[i][j]\*b[i][j]);

} printf("\n");

} } else

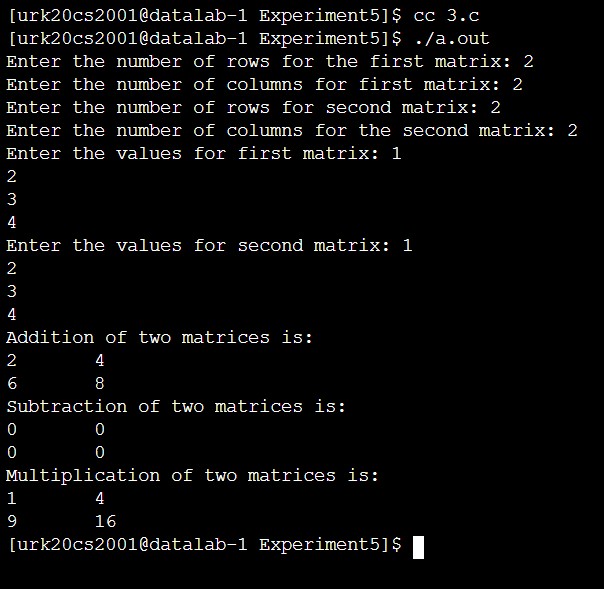
{

printf("Matrix multiplication is impossible!!\n");

}

}

**Output:**



**4)Aim:**

To write a C program to implement a Bubble sort.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables array[100], swap, n, i, j.

**Step4:** Calculating the array using bubble sort.

**Step5:** Print the results.

**Step6:** End.

**Program:**

#include<stdio.h> int main() { int array[100], swap, n, i, j;

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

printf("Enter %d integers: ", n);

for(i=0;i<n;i++) scanf("%d", &array[i]);

for(i=0;i<n-1;i++)

{ for(j=0;j<n-i-1;j++)

{ if(array[j]>array[j+1])

{ swap = array[j]; array[j] = array[j+1]; array[j+1] = swap;

}

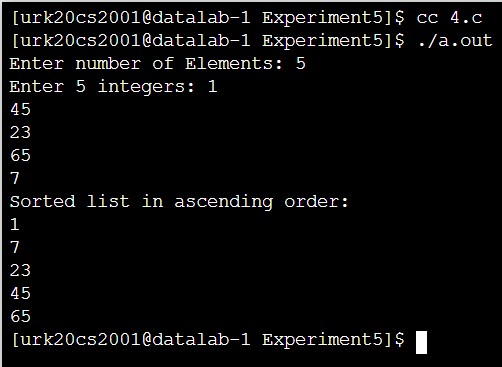
}

}

printf("Sorted list in ascending order: \n"); for(i=0;i<n;i++) printf("%d\n", array[i]); return 0;

}

**Output:**



**Result:**

Thus the program for experimenting Usage of arrays in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp no: 6** | **String Operations** |
| **Date: 17-11-2020** |

**1)Aim:**

To write a C program to show the usage of string library functions.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables len, comp, char.

**Step4:** calculating the output by using the string functions.

**Step5:** Print the result.

**Step6:** End.

**Program:**

#include<stdio.h> #include<string.h> int main() {

int len, comp; char name1[50] = "Ruban"; char name2[50] = "Gino"; char name3[50] = "Singh"; char name4[50];

len = strlen(name1);

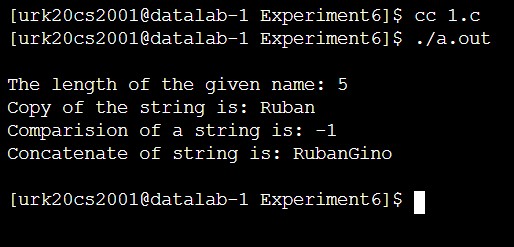
printf("\nThe length of the given name: %d \n", len); printf("Copy of the string is: "); strcpy(name4, name1); printf("%s\n", &name4);

printf("Comparision of a string is: "); comp = strcmp(name1, name3);

printf("%d \n", comp); printf("Concatenate of string is: "); strcat(name1, name2); printf("%s\n\n", name1);

}

**Output:**



**2)Aim:**

To write a C program to get a string input from the user and check if its palindrome.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files

**Step3:** Declare the variables char str1[20], int i, length, flag=0.

**Step4:** Get the string input from the user.

**Step5:** Calculating the result.

**Step6:** Print the output.

**Step7:** End.

**Program:**

#include <stdio.h>

#include <string.h>

int main(){ char str1[20]; int i, length; int flag = 0;

printf("Enter a string:"); scanf("%s", str1);

length = strlen(str1);

for(i=0;i < length ;i++)

{

if(str1[i] != str1[length-i-1])

{ flag = 1; break;

}

}

if (flag)

{

printf("%s is not a palindrome\n", str1);

} else

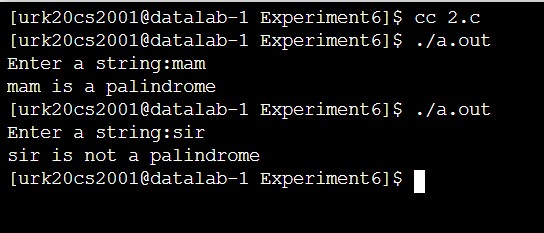
{

printf("%s is a palindrome\n", str1);

} return 0;

}

**Output:**



**3)Aim:**

To write a C program to create a login application.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables char username[20], password[20].

**Step4:** Get the username and password from the user.

**Step5:** Calculating the program with If else condition.

**Step6:** Print the result.

**Step7:** End.

**Program:**

#include<stdio.h> #include<string.h> int main() {

char username[20]; char password[20];

printf("Enter the Username: "); scanf("%s", &username); printf("Enter the password: "); scanf("%s", &password);

if(strcmp(username, "admin")==0)

{

if(strcmp(password, "karunya")==0)

{

printf("Login Successful \n");

} else

{

printf("Wrong password\n");

} } else

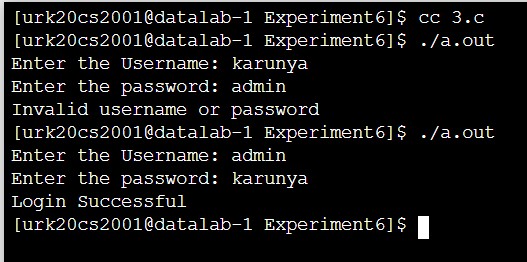
{

printf("Invalid username or password\n");

} return 0;

}

**Output:**



**4)Aim:**

To write a C program to get full name as input from the user and Identify the vowels in the full name and change it to uppercase.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables char str1[100], int i.

**Step4:** Get the Name as an input.

**Step5:** Checking the while condition.

**Step6:** Print the result.

**Step7:** End.

**Program:**

#include<stdio.h> int main() {

char str1[100]; int i;

printf("Enter your Name: "); gets(str1);

printf("The original string is: ");

puts(str1);

i=0;

while(str1[i]!='

\

0

'

)

{

if(st

r1[i]=='a' || str1[i]=='e' || str1[i]=='i' || str1[i]=='o' || str1[i]=='u')

str1[i]=str1[i]

-

32

;

i++;

}

printf("After Converting vowels into uppercase:

\

n");

puts(str1);

}

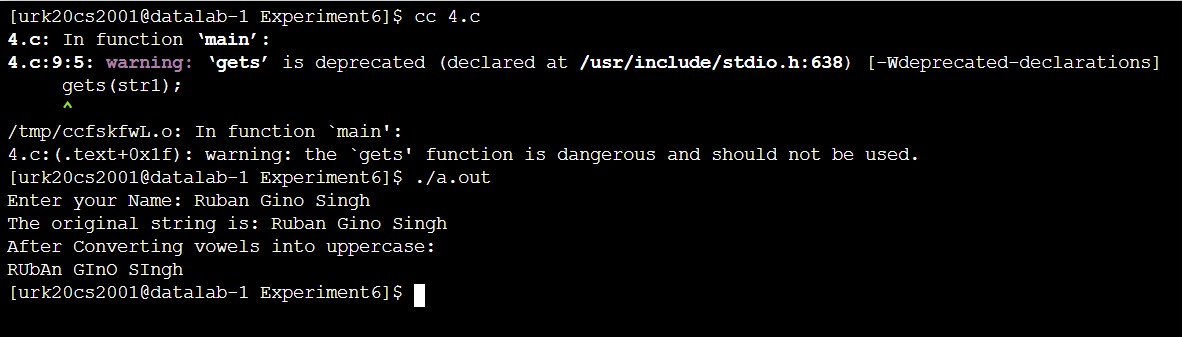
**Output:**

**Result:**

Thus the program for exper

imenting String operators in C programming are coded,

compiled and executed successfully.



|  |  |
| --- | --- |
| **Exp No: 7** | **User Defined Functions** |
| **Date: 24-11-2020** |

**Programs:**

**1)Aim:**

To Write a C program to create a user defined function that accepts seconds as input. And display the time in hh:mm:ss format.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables sec.

**Step4:** Get the input from the user.

**Step5:** Convert the seconds to hh:mm:ss format.

**Step6:** Print the results.

**Step7:** End.

**Programs:**

#include<stdio.h> #include<math.h> int time(int); int main()

{ int sec;

printf("Enter the time in seconds: "); scanf("%d", &sec);

time(sec); return(0);

}

int time(int sec)

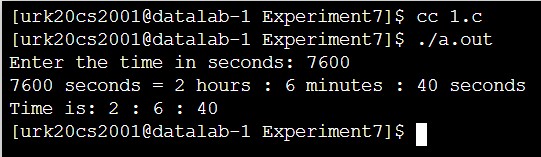
{

int hh, mm, ss; hh = sec / 3600; mm = (sec - hh \* 3600) / 60; ss = sec - hh \* 3600 - mm \* 60;

printf("%d seconds = %d hours : %d minutes : %d seconds\n", sec, hh, mm, ss); printf("Time is: %d : %d : %d\n", hh, mm, ss);

}

**Output:**



**2)Aim:**

To write a C program to Create a user defined function that accepts a decimal number as an input and prints the binary equivalent of the number.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables n, i, a[10].

**Step4**: Get the integer input from the user.

**Step5:** Convert the decimal number to a Binary number.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h> #include<stdlib.h> int binary(int); int main() {

int n,i,a[10]; printf("Enter the number that you want to Convert: "); scanf("%d", &n);

binary(n); return(0);

}

int binary(int n)

{

int a[10],i; for(i=0;n>0;i++)

{ a[i]=n%2; n=n/2;

}

printf("Binary form of the given number is= "); for(i=i-1;i>=0;i--)

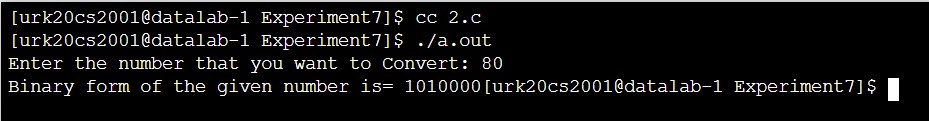
{

printf("%d", a[i]);

}

}

**Output:**



**3)Aim:**

To Write a C program to create a user defined function that accepts a string as input and returns the number of vowels and consonants in the string.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables vowels=0, consonants=0, i.

**Step4:** Get the string input from the user.

**Step5:** Returning the no of vowels and consonants in the giver string.

**Step6:** Print the Results.

**Step7:** End.

**Program:**

#include<string.h> void stringcount(char\*s)

{

int vowels=0, consonants=0, i;

for(i=0;s[i];i++)

{

if((s[i]>=65 && s[i]<=90)|| (s[i]>=97 && s[i]<=122))

{ if(s[i]=='a'|| s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u') vowels++;

else consonants++;

}

}

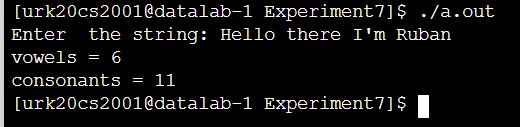
printf("vowels = %d\n",vowels); printf("consonants = %d\n",consonants);

} int main() {

char s[1000]; printf("Enter the string: "); gets(s); stringcount(s);

}

**Output:**



**4)Aim:**

To write a C program to create a functions that accepts a float array as input and returns the maximum value of the array.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables int i, n and float arr[100].

**Step4:** Get the number of elements from the user.

**Step5:** Return the maximum of the array.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h> int maximum(int); int main() {

int n, i; float arr[100];

printf("Enter the number of elements form 1 to 100: ");

scanf("%d", &n); maximum(n);

return 0;

}

int maximum(int n)

{

int i; float arr[100]; for(i=0;i<n;++i)

{

printf("Number%d: ", i+1); scanf("%f", &arr[i]);

}

for(i=1;i<n;++i)

{

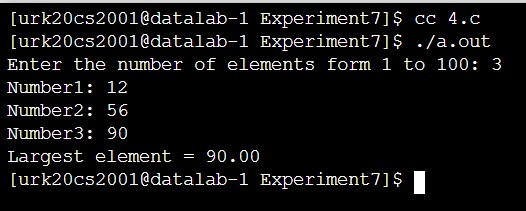
if(arr[0]<arr[i]) arr[0] = arr[i];

}

printf("Largest element = %.2f\n", arr[0]);

}

**Output:**



**5)Aim:**

To write a C program to print the Fibonacci series up to a given range by using recursive functions.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables int n, m=0, i.

**Step4:** Get the total terms from the user.

**Step5:** Calculating the Fibonacci series.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h> int fibonacci(int); int main() {

int n, m=0, i;

printf("\n Enter total terms: \n"); scanf("%d", &n); printf("\nFibonacci series terms are:\n"); for(i = 1; i <= n; i++)

{ printf("%d\t", fibonacci(m)); m++;

} return 0;

}

int fibonacci(int n)

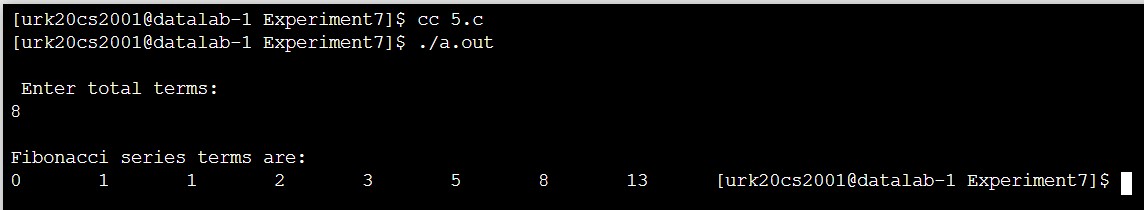
{

if(n==0 || n==1) return n; else

return(fibonacci(n-1) + fibonacci(n-2));

}

**Output:**



**Result:**

Thus the program for experimenting the user defined functions in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp no: 8** | **Usage of Pointers** |
| **Date: 1-12-2020** |

**1)Aim:**

To write a C program to swap two numbers using pass by value and pass by reference method.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables int temp inside the function void swap.

**Step4:** Get the input as a numbers from the user.

**Step5:** Pass the value inside the functions void swap to calculate the program.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h>

void swap(int number1, int number2)

{ int temp; printf("The values before swapping: %d %d \n", number1, number2); temp = number1; number1 = number2;

number2 = temp;

printf("The values after swapping is: %d %d \n", number1, number2);

}

int main() {

int num1, num2; printf("Enter the 1st Number: "); scanf("%d", &num1);

printf("Enter the 2nd Number: "); scanf("%d", &num2);

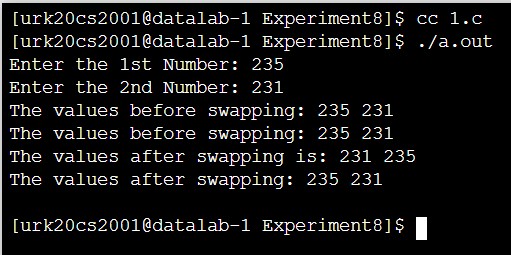
printf("The values before swapping: %d %d \n", num1, num2); swap(num1, num2);

printf("The values after swapping: %d %d \n \n", num1, num2);

return 0;

}

**Output:**



**2)Aim:**

To write a C program to pass an array in a function. Search and replace the element with 0 inside the function. Print the array before function call and after function call.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables arr[5], I, number. int \*arr, int number inside the function.

**Step4:** Get the input from the user.

**Step5:** Passing the array inside of the function to get the required result.

**Step6:** Print the result.

**Step7:** End.

**Program:**

#include<stdio.h> int main() {

int arr[5], i, number;

printf("Enter the values for the array: \n"); for(i=0; i<5; i++)

{

scanf("%d", &arr[i]);

}

printf("\n Enter the number that you wants to be searched and replaced with 0: \n"); scanf("%d", &number);

printf("\nThe Main Array Before function call: \n"); for(i=0; i<5; i++)

{

printf("%d\t", arr[i]);

}

search(arr, number);

printf("\nThe Main Array After function call: \n"); for(i=0; i<5; i++)

{

printf("%d\t", arr[i]);

}

}

void search(int \*arr, int number)

{

int i;

for(i=0; i<5; i++)

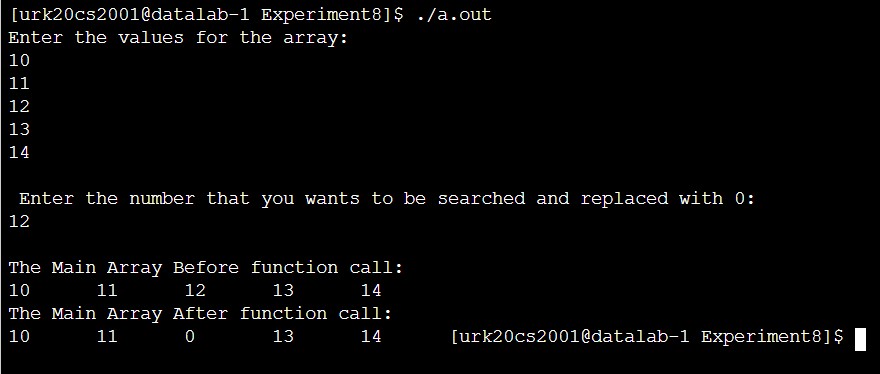
{

if(\*arr == number) \*arr = 0; arr++;

}

}

**Output:**



**3)Aim:**

To write a C program to pass a character array in a function. Convert the character array to uppercase inside the function. And print the array before function call and after function call.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variales int i and char str[10].

**Step4:** Get the input string from the user.

**Step5:** Passing the character array in a function converting the character array to uppercase inside the function.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h> #include<string.h> void convert(char \*str)

{

int i;

while(\*str != '\0')

{

\*str = toupper(\*str); str++;

}

}

int main()

{

int i; char str[10];

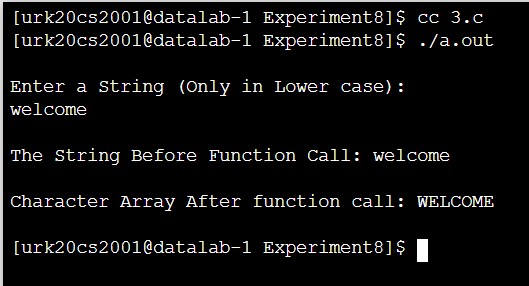
printf("\nEnter a String (Only in Lower case): \n"); scanf("%s", &str);

printf("\nThe String Before Function Call: %s\n", str);

convert(str);

printf("\nCharacter Array After function call: %s\n\n", str); }

**Output:**



**4)Aim:**

To write a C program to store an array using pointer as an argument and dynamic memory allocation.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables int \*ptr, n, i.

**Step4:** Get the array size as an input from the user.

**Step5:** sorting the array using pointer.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h>

#include<stdlib.h>

void sorting(int n, int \*ptr)

{

int i, j, t;

for(i=0; i<n; i++)

{

for(j=i+1; j<n; j++)

{

if(\*(ptr+j) < \*(ptr + i))

{

t = \*(ptr + i);

\*(ptr + i) = \*(ptr + j);

\*(ptr + j) = t;

}

}

}

}

int main() { int \*ptr; int n, i;

printf("\nEnter the size of an Array that you want to be Created: "); scanf("%d", &n);

ptr = (int\*)malloc(n\*sizeof(int)); if(ptr == NULL)

{

printf("Memory Not Allocated. \n"); exit(0); } else

{

printf("\nEnter the Elements in the array: ");

for(i=0; i<n; i++)

{

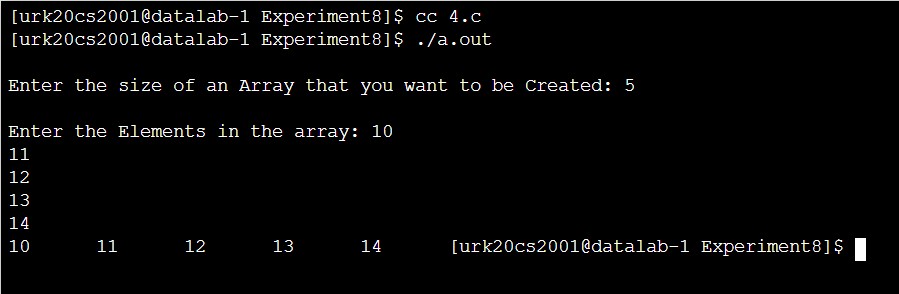
scanf("%d", &ptr[i]);

} }

sorting(n, ptr); for(i=0; i<n; i++) printf("%d\t", \*(ptr + i)); return 0;

}

**Output:**



**Result:**

Thus the program for experimenting Usage of Pointer in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp no: 9** | **Structures In C Programming** |
| **Date: 8-12-2020** |

**1)Aim:**

To write a C program to create a structure date to Instantiate DOB and Current date to calculate the age of the person

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables currentDate, birthdate, calculateDate and other required varables etc..,

**Step4:** Create a Structure of functions.

**Step5:** Get the Input from the user to ask current date details and Birth date details.

**Step6:** Calculate the structure of dates using If else conditions inside of functions.

**Step7:** Print the results.

**Step8:** End.

**Program:**

#include<stdio.h>

#include<stdlib.h>

int currentDate, birthDate, calculatedDate, currentMonth, birthMonth, calculatedMonth, currentYear, birthYear, calculatedYear;

void year(int currentDate, int currentMonth, int currentYear, int birthDate, int birthMonth, int birthYear)

{

if(birthDate > currentDate)

{

currentMonth = currentMonth - 1; currentDate = currentDate + 30;

}

if(birthMonth > currentMonth)

{

currentYear = currentYear - 1; currentMonth = currentMonth + 12;

}

if(birthYear > currentYear)

{ exit(0);

}

calculatedDate = currentDate - birthDate; calculatedMonth = currentMonth - birthMonth; calculatedYear = currentYear - birthYear;

printf("\nYour Present Age \nYears: %d\t Months: %d\t Days: %d\n", calculatedYear, calculatedMonth, calculatedDate);

}

int main() {

printf("\nEnter current date details below!! ");

printf("\nEnter Today's Date: \t"); scanf("%d", &currentDate);

printf("Enter Current Month: \t"); scanf("%d", &currentMonth);

printf("Enter Current Year: \t"); scanf("%d", &currentYear);

printf("\nEnter your birth details: \n");

printf("Enter Day: \t"); scanf("%d", &birthDate);

printf("Enter Month: \t"); scanf("%d", &birthMonth);

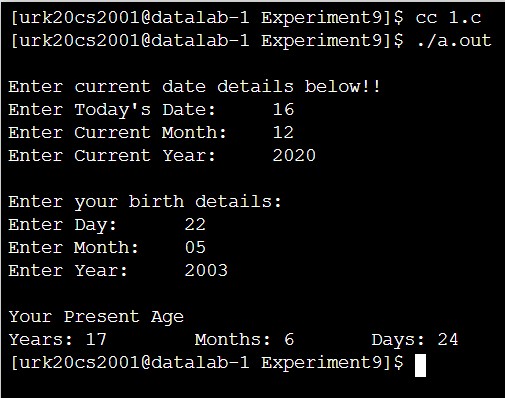
printf("Enter Year: \t"); scanf("%d", &birthYear);

year(currentDate, currentMonth, currentYear, birthDate, birthMonth, birthYear);

return 0;

}

**Output:**



**2)Aim:**

To write a C program using structure to create a payroll report of employees in an organization.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables int empno, char name[10], allowances, basicPay, deductions, nPay.

**Step4:** Get the employee name, number and basic pay etc…,

**Step5:** Calculating the values.

**Step6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h>

struct emp { int empno; char name[10];

int allowances, basicPay, deductions, nPay;

}e[10];

void main() {

int i, n;

printf("Enter the number of employees: "); scanf("%d", &n);

for(i=0; i<n; i++)

{

printf("\nEnter the employee number: "); scanf("%d", &e[i].empno);

printf("\nEnter the Name: "); scanf("%s", e[i].name);

printf("\nEnter the Basic pay, Allowances & Deductions: "); scanf("%d %d %d", &e[i].basicPay, &e[i].allowances, &e[i].deductions);

e[i].nPay = e[i].basicPay + e[i].allowances - e[i].deductions;

}

printf("Employee.No.\tName\tBasic Pay\tAllow\tDeductions\tNpay\n\n");

for(i=0; i<n; i++)

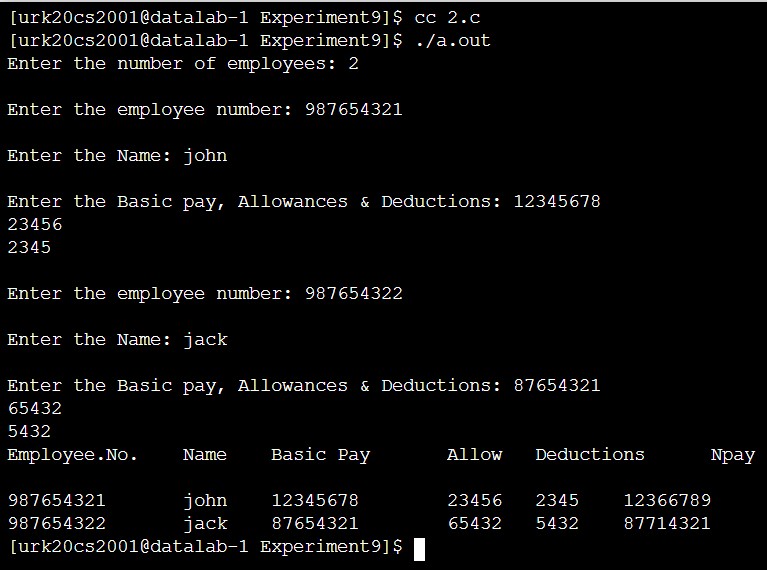
{

printf("%d\t%s\t%d\t%d\t%d\t%d\n", e[i].empno, e[i].name, e[i].basicPay, e[i].allowances, e[i].deductions, e[i].nPay);

} getchar();

}

**Output:**



**3)Aim:**

To write a C program accept records of different states using arrays of structures.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables engineering, medical, management, university, total.

**Step4:** Calculating and arranging the details in a table format.

**Step5:** Print the results.

**Step6:** End.

**Program:**

#include<stdio.h>

struct names{ char state[10];

int engineering, medical, management, university, total;

};

int main() {

int n, i, max=0;

printf("Enter the number of states to accept: ");

scanf("%d", &n);

struct names arr[n];

for(i=0; i<n; i++)

{

printf("Enter the name of the state number %d: ", i+1); scanf("%s", &arr[i].state);

printf("Enter the number of Engineering Colleges: "); scanf("%d", &arr[i].engineering);

printf("Enter the number of Medical Colleges: "); scanf("%d", &arr[i].medical);

printf("Enter the number of Management Colleges: "); scanf("%d", &arr[i].management);

printf("Enter the number of Universities: "); scanf("%d", &arr[i].university);

arr[i].total = arr[i].engineering + arr[i].medical + arr[i].management + arr[i].university;

if(arr[i].total > arr[max].total) max = i;

}

printf("SI.No.\tState\tEngineering\tMedical\tManagement\tUniversities\n");

for(i=0; i<n; i++)

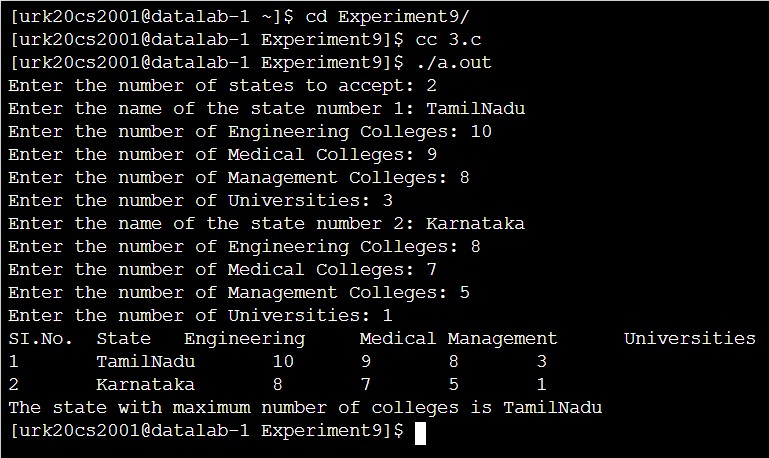
{

printf("%d\t%s\t%d\t%d\t%d\t%d\n", i+1, arr[i].state, arr[i].engineering, arr[i].medical, arr[i].management, arr[i].university);

}

printf("The state with maximum number of colleges is %s \n", arr[max].state); }

**Output:**



**Result:**

Thus the program for experimenting the Structures in C programming are coded, compiled and executed successfully.

|  |  |
| --- | --- |
| **Exp no: 10** | **File Handling** |
| **Date: 12-12-2020** |

**1)Aim:**

To write a C program for student information storage and retrieval using file handling.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables char text, ch \*fp and other required variables.

**Step4:** Get the Student Information from an user.

**Step5:** Creating a file to saving that informations.

**Step 6:** Print the results.

**Step7:** End.

**Program:**

#include<stdio.h>

#define SIZE 1024

int main() {

char text[SIZE] = {0},record[100];

char ch; int cnt=0,len;

FILE \*fp;

fp = fopen("file1.txt", "w");

if(fp==NULL)

{

printf("Error in creating file"); return -1;

}

printf("\nEnter student details line wise (Press # to save and close): \n");

printf("Enter in format NAME - AGE - TOTAL MARKS \n"); while(1)

{ ch=getchar(); if(ch=='#')

{ break; } fputc(ch,fp);

} fclose(fp);

printf("File saved successfully\n");

fp=fopen("file1.txt", "r"); if(fp==NULL)

{

printf("Error in opening file.\n"); return -1;

}

printf("Entered records are: \n");

cnt=0;

while((ch=fgetc(fp))!=EOF)

{

if(ch==0x0A)

{ record[cnt++]='\0'; cnt=0; printf("%s\n", record); continue;

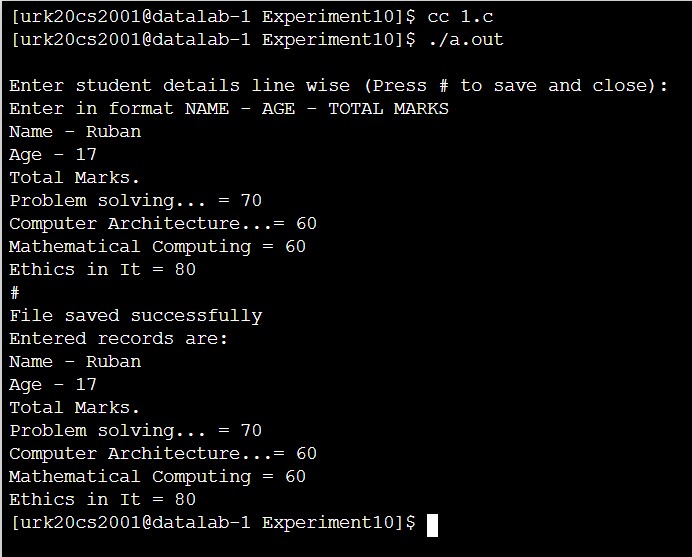
}

record[cnt++]=ch;

} fclose(fp); return 0;

}

**Output:**



**2)Aim:**

To write a C program to copy a file from one location to another.

**Algorithm:**

**Step1:** Start.

**Step2:** Include the required header files.

**Step3:** Declare the variables \*fp, \*fp1, char name, file, x and z.

**Step4:** Get the input from the user.

**Step5:** Type the input in a new created file.

**Step6:** Copy the text from the File.txt to File1.txt.

**Step7:** Print the results.

**Step8:** End.

**Program:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

FILE \*fp, \*fp1;

char name[100], file[100], x[100], z[100];

printf("Enter what you wants to store in the File.txt in order to be copied in File1.txt:\n"); scanf("%[^\n]%\*c", name);

printf("Enter the file name you want to store data in: \n"); scanf("%s", file);

strcpy(x, file); fp=fopen(file, "w+"); fprintf(fp, "%s\n", name); fclose(fp);

printf("Enter the file name you want to copy data in:\n"); scanf("%s", file);

strcpy(z, file);

fp1=fopen(file, "w+"); fclose(fp1);

FILE \*fptr1, \*fptr2; char filename[100], c;

printf("Enter the filename to open for reading: \n"); scanf("%s", filename);

fptr1=fopen(filename, "r");

if(fptr1==NULL)

{ printf("Cannot open file %s \n"< filename); exit(0);

}

printf("Enter the filename to open for writing \n"); scanf("%s", filename);

fptr2=fopen(filename, "w");

if(fptr2==NULL)

{

printf("Cannot open file %s \n", filename); exit(0);

}

c=fgetc(fptr1); while(c!=EOF)

{ fputc(c, fptr2); c=fgetc(fptr1);

}

printf("\n Contents copied to %s\n", filename); fclose(fptr1); fclose(fptr2);

FILE \*fps, \*fps1;

printf("\nThe contents of %s \n", x);

fps=fopen(x, "r");

if(fps==NULL)

{

printf("Cannot open file \n"); exit(0);

}

c=fgetc(fps); while(c!=EOF)

{ printf("%c", c); c=fgetc(fps); } fclose(fps);

printf("\nThe contents of %s \n", z); fps1=fopen(z, "r");

if(fps1==NULL)

{ printf("Cannot open file\n"); exit(0);

}

c=fgetc(fps1); while(c!=EOF)

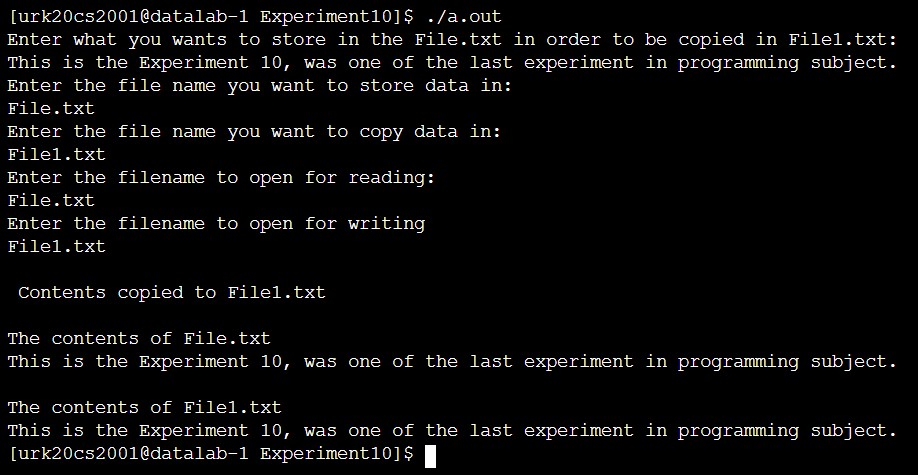
{ printf("%c", c); c=fgetc(fps1);

}

fclose(fps1); return 0;

}

**Output:**



**Result:**

Thus the program for experimenting File Handling in C programming are coded, compiled and executed successfully.