**SET:-1**

**1.1 Simple programs-1: Print “Hello world!”; Print “My name is ”; Print “.**

#include<stdio.h>

void main()

{

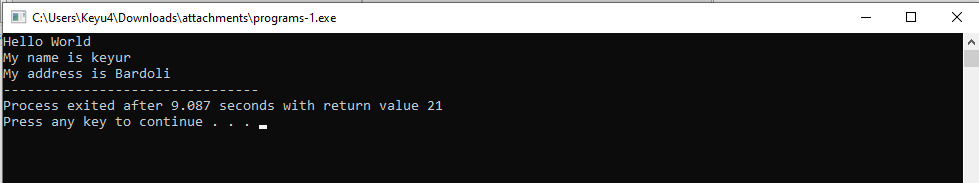
printf("Hello World\n");

printf("My name is keyur\n");

printf("My address is Bardoli");

}

**Output:-**



**1.2 Simple programs-2a: x1 = 2; x2 = 7; Print “x1 = value, y1 = value”.**

#include<stdio.h>

void main()

{

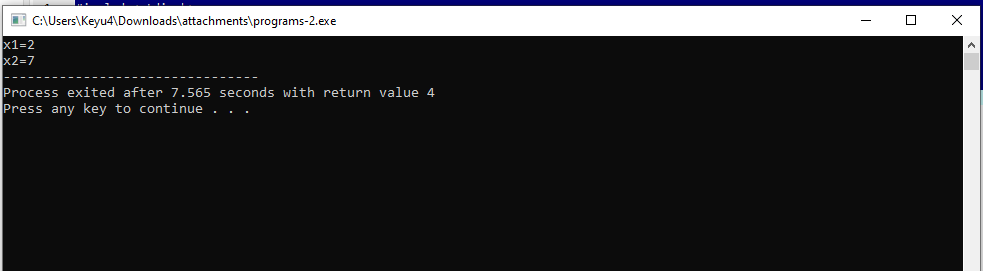
int x1=2,x2=7;

printf("x1=%d\n",x1);

printf("x2=%d",x2);

}

**Output:-**



**1.3 Write a program to take 5 integers and find and print the total and average of the 5 numbers. Repeat the same for floating point numbers instead of integers.**

#include<stdio.h>

void main()

{

float a=40,b=40,c=40,d=40,e=40,tot=0,avg=1;

tot=a+b+c+d+e;

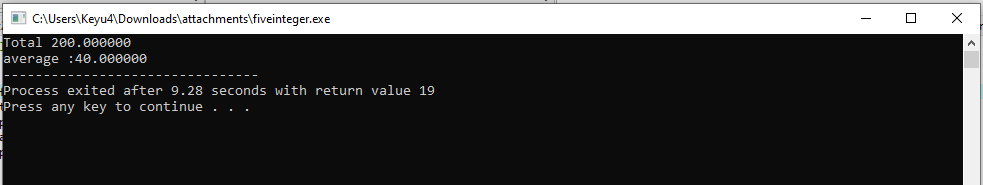
printf("Total %f",tot);

avg=tot\*100/500;

printf("\naverage :%f",avg);

}

**Output:-**

****

**1.4 Write a program to find the sum of 1, 2, 3, … , n. Print average (avg) also.**

#include<stdio.h>

void main()

{

int i,val=5,sum=0,avg=1;

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

{

sum=sum+i;

}

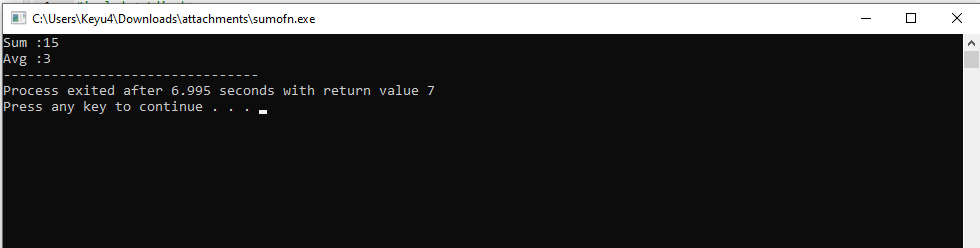
avg=sum/val;

printf("Sum :%d",sum);

printf("\nAvg :%d",avg);

}

**Output:-**



**1.5 Write a program to accept n. Find sum of n values accepted 1-by-1. Also find average (avg). Print sum, avg. Additionally, print the input values also.**

#include<stdio.h>

void main()

{

int i,val,sum=0,avg=1;

printf("Enter values:");

scanf("%d",&val);

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

{

sum=sum+i;

}

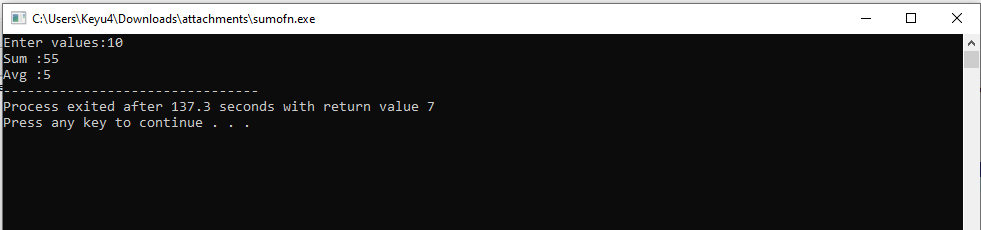
avg=sum/val;

printf("Sum :%d",sum);

printf("\nAvg :%d",avg);

}

**Output:-**

****

**1.6 Write a program to accept n and n input values to be stored in an array. Find sum and average (avg) of n values. Print input values followed by sum, avg.**

#include<stdio.h>

void main()

{

int i,sum=0,avg=1;

int val[5];

printf("Enter values:");

for(i=1;i<=5;i++)

{

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

sum=sum+val[i];

}

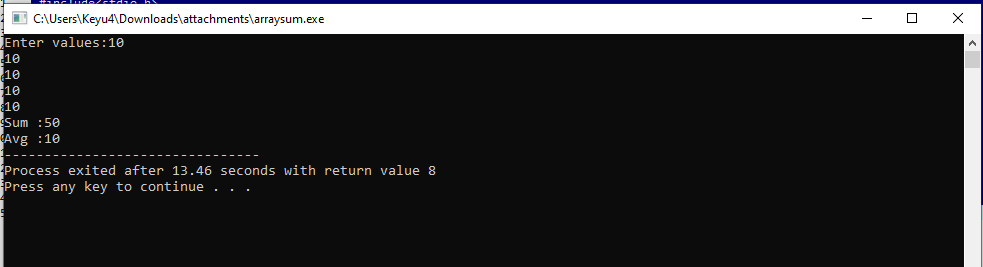
avg=sum/5;

printf("Sum :%d",sum);

printf("\nAvg :%d",avg);

}

**Output:-**



**SET:-2**

**2.1 Write a program to accept as input: first name, middle name, surname; then print name, first as (a) first mid surname; and thereafter (b) surname first mid,**

#include<stdio.h>

void main()

{

char nm[5];

char fnm[5];

char mnm[5];

printf("Enter FirstName: ");

scanf("%s",&nm);

printf("Enter MidName: ");

scanf("%s",&fnm);

printf("Enter Surname: ");

scanf("%s",&mnm);

printf("\nEnter FirstName: %s",nm);

printf("\nEnter MidName: %s",fnm);

printf("\nEnter Surname: %s",mnm);

printf("\n");

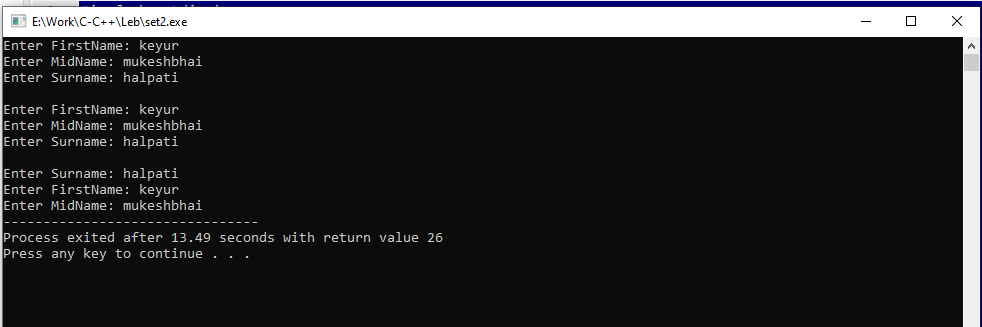
printf("\nEnter Surname: %s",mnm);

printf("\nEnter FirstName: %s",nm);

printf("\nEnter MidName: %s",fnm);

}

**Output:-**



**2.2** **Write a program to find string length. Is the string length same as number of characters in the string?**

#include<stdio.h>

#include<string.h>

void main()

{

int i,c=0;

char s1[30];

printf("Enter string: ");

gets(s1);

for(i=0;s1[i]!='\0';i++)

{

c++;

if(s1[i]==' ')

{

c--;

}

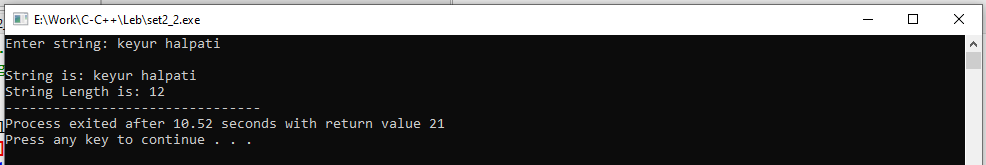
}

printf("\nString is: %s",s1);

printf("\nString Length is: %d",c);

}

**Output:-**



**2.3 write the program to find string length by using function for finding string length. Test this program to find lengths of first, mid, and surname.**

#include<stdio.h>

#include<string.h>

void main()

{

char a[30];

int i,l=0;

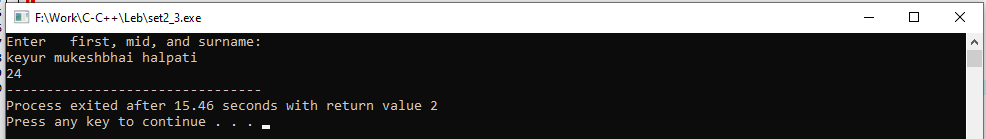
printf("Enter first, mid, and surname: \n");

gets(a);

printf("%s",strlen(a));

}

**Output:-**

****

**2.4 Write a program to print a given string in reverse order.**

#include<stdio.h>

#include<string.h>

void main()

{

char a[30];

int i,l=0;

printf("Enter string: \n");

gets(a);

l=strlen(a);

for(i=l;i>=0;i--)

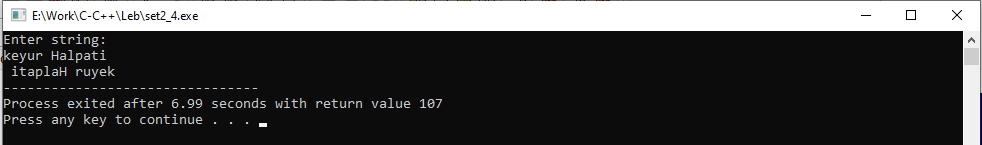
{

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

}

}

**Output:-**



**SET:-3**

**3.1 Write a program to take Input: n, and n values of temperature in oF, convert these into oC and print the values in a table with 1st column containing oF and 2nd column oC.**

#include<stdio.h>

void main()

{

int n;

printf("1.Fahrenheit to degrees Celsius:\n");

printf("2.Celsius to degrees Fahrenheit:\n");

printf("Enter Choice:");

scanf("%d",&n);

if(n==1)

{

float f,c;

printf("Enter Fahrenheit:");

scanf("%f",&f);

c=(f-32)/1.8;

printf("Celsius :%f",c);

}

else if(n==2)

{

float f,c;

printf("Enter Celsius:");

scanf("%f",&c);

f=(c\*1.8)+32;

printf("Fahrenheit :%f",f);

}

else

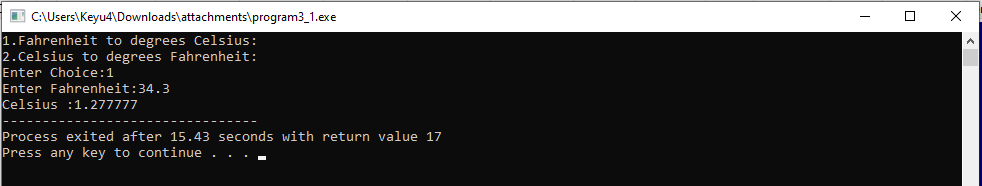
{

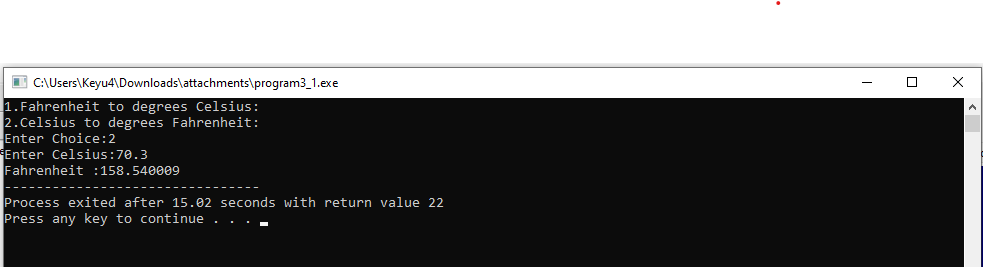
printf("Invalid choice");

}

}

**Output:-**





**3.2 write program to convert temperature in oC into oF. Write a function c2f() for this operation.**

#include<stdio.h>

float c2f(float c)

{

float f1=0;

f1=(1.8\*c)+32;

return f1;

}

void main()

{

float c,f;

printf("Enter celsius: ");

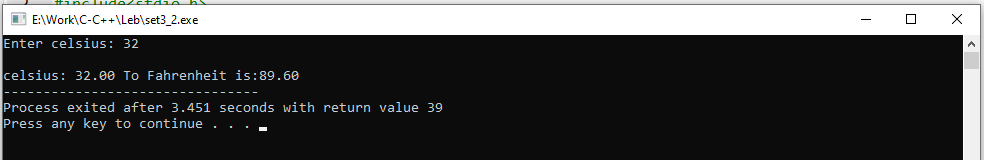
scanf("%f",&c);

f=c2f(c);

printf("\ncelsius: %.2f To Fahrenheit is:%.2f ",c,f);

}

**Output:-**



**SET:- 4**

**4.1 Write a program in C: Inputs: 2 integer values: Numerator (num1) and Denominator (num2). Output: Quotient (q) and Remainder (r).**

#include<stdio.h>

void main()

{

int n1,n2;

float q,rem;

printf("\nEnter the value of n1: ");

scanf("%d",&n1);

printf("\nEnter the value of n2: ");

scanf("%d",&n2);

rem = n1%n2;

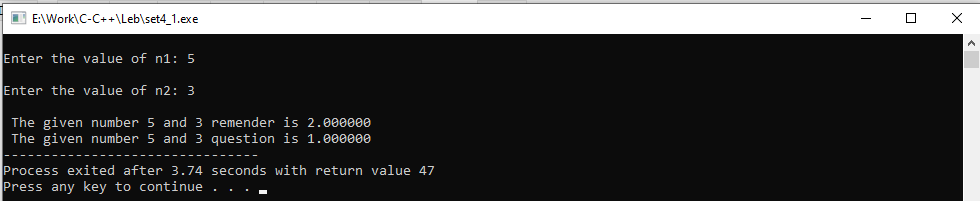
q = n1/n2;

printf("\n The given number %d and %d remender is %f ",n1,n2,rem);

printf("\n The given number %d and %d question is %f",n1,n2,q);

}

**Output:-**



**4.2 Develop an algorithm write a program in C to print all primes in the first n (n > 1) integers. Develop the most efficient algorithm.**

#include<stdio.h>

void main()

{

int i,j,n;

printf("Enter Number: ");

scanf("%d",&n);

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

{

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

{

if(i%j==0)

{

break;

}

}

if(j==i)

{

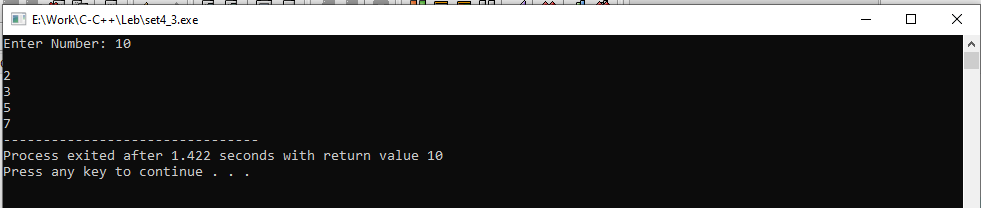
printf("\n%d ",j);

}

}

}

**Output:-**

****

**SET:-5**

**5.1 Given two integers m and n (m, n >= 0), develop an algorithm and write a program in C to find their greatest common divisor (gcd).**

#include<stdio.h>

int gcd\_algo(int a,int b)

{

if(b==0)

{

return a;

}

else if (b!=0)

{

return gcd\_algo(b,(a%b));

}

}

void main()

{

int x,y;

printf("\n Enter number1 : ");

scanf("%d",&x);

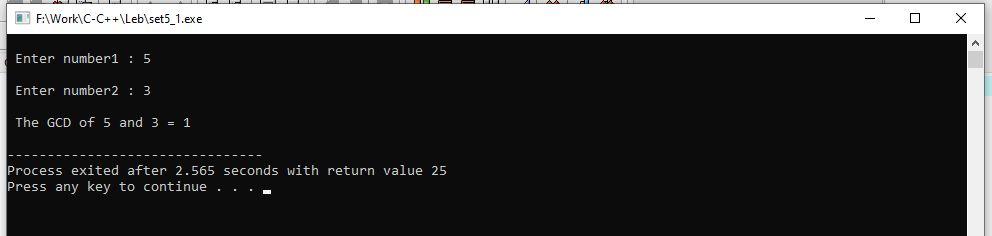
printf("\n Enter number2 : ");

scanf("%d",&y);

printf("\n The GCD of %d and %d = %d\n",x,y,gcd\_algo(x,y));

}

**Output:-**

****

**5.2 Write a program in C to rearrange the elements in an array so that they appear in reverse order.**

#include<stdio.h>

void main()

{

int a[5],i;

printf("Enter Array Elements\n");

for(i=1;i<=5;i++)

{

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

}

printf("Elements are:\n");

for(i=5;i>0;i--)

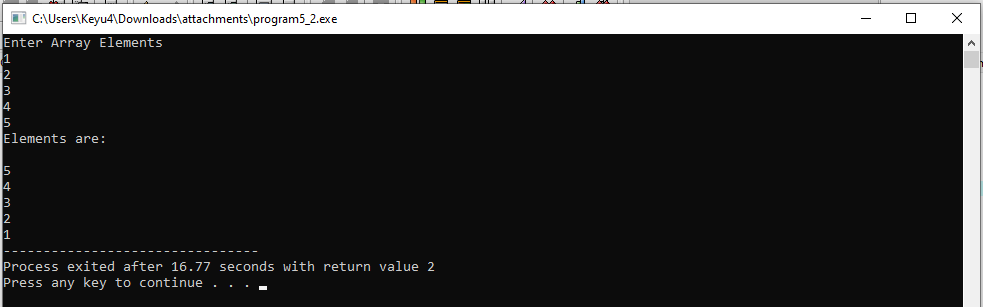
{

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

}

}

**Ouput:-**



**SET:-6**

**6.1 Write a program to calculate and display the value of the slope of the line connecting the two points whose coordinates are (3, 7) and (8, 12). Slope of a line between two points (x1 , y1 ) and (x2 , y2 ) is (y2 – y1 ) / (x2 – x1 ). Run the same program for the line connecting the points (2, 10) and (12, 6), and other pairs of points.**

#include<stdio.h>

void main()

{

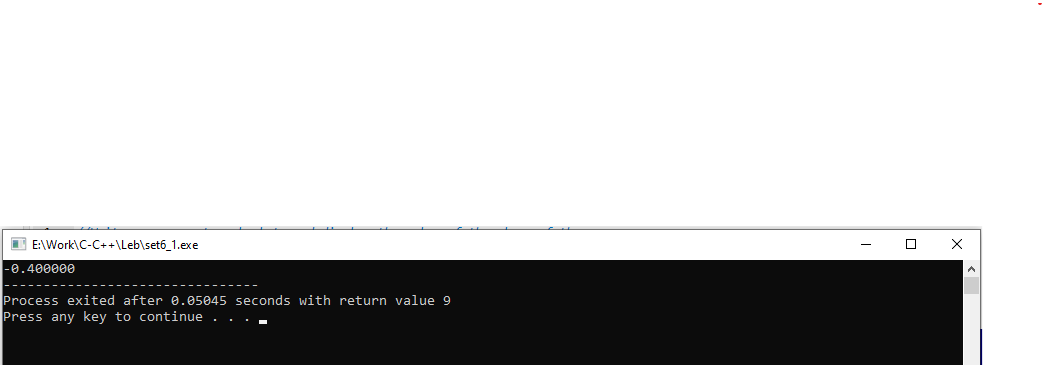
float x1=2,x2=12,y1=10,y2=6,c=0;

c=(y2 - y1)/(x2 - x1);

printf("%f",c);

}

**Output:-**

****

**6.2 Write a program to calculate and display the coordinates of the midpoint of the line connecting the two points given in the previous Exercise. The coordinates of the midpoint between two points having coordinates (x1 , y1 ) and (x2 , y2 ) are ((xl + x2 ) / 2, (y1 + y2 ) / 2).**

#include<stdio.h>

void main()

{

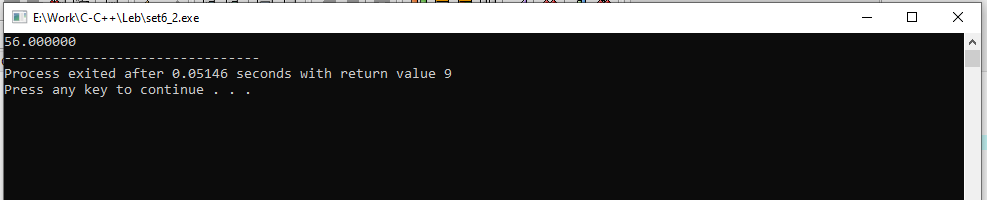
float x1=2,x2=12,y1=10,y2=6,c=0;

c=((x1 + x2 ) / 2 \* (y1 + y2 ) / 2);

printf("%f",c);

}

**Output:-**

****

**6.3 Write a program that calculates the distance between two points whose coordinates are (7, 12) and (3, 9). Distance between two points having coordinates (xl , y1 ) and (x2 , y2 ) = sqrt([xl – x2 ] 2 + [yl – y2 ] 2 ). Also, run the program for the points (-12, -15) and (22, 5) and a few other points.**

#include<stdio.h>

#include<math.h>

void main()

{

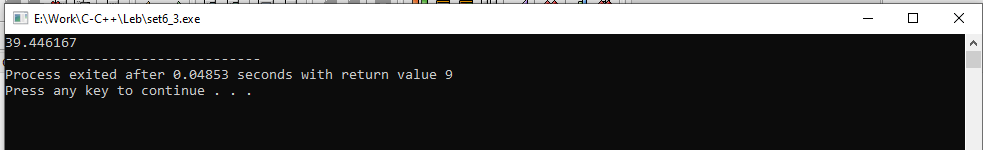
float x1=-12,x2=22,y1=-15,y2=5,c=0;

c= sqrt((x1 - x2)\*(x1 - x2) + (y1 - y2)\*(y1 - y2));

printf("%f",c);

}

**Output:-**



**SET:- 7**

**7.1 Given some integer x, develop an algorithm and write a program to compute the value of x^n where n is considerably larger than 1. This algorithm has time complexity O(n).**

#include<stdio.h>

#include<math.h>

void main()

{

int i,x,n,ans=1;

printf("Enter X:");

scanf("%d",&x);

printf("Enter N:");

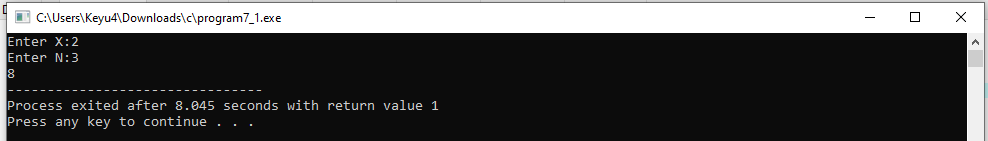
scanf("%d",&n);

ans=pow(x,n);

printf("%d",ans);

}

**Output:-**



**7.2 Develop an improved algorithm having time complexity O(log2n).**

#include<stdio.h>

#include<math.h>

void main()

{

float n,ans;

printf("Enter N:");

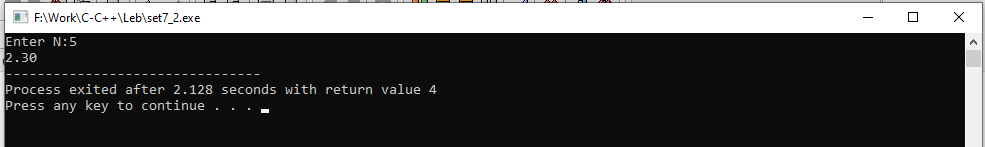
scanf("%f",&n);

ans=2\*n;

printf("%.2f",log(ans));

}

**Output:-**

****

**SET:-8**

**8.1 Write a program to evaluate sin(x).**

#include<stdio.h>

#include<math.h>

void main()

{

float x;

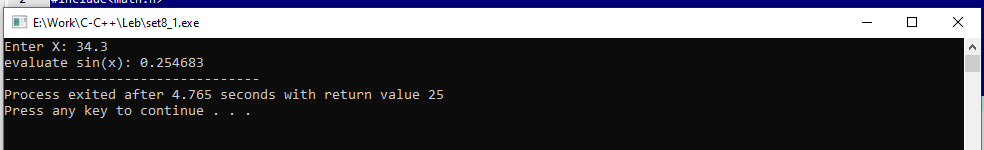
printf("Enter X: ");

scanf("%f",&x);

printf("evaluate sin(x): %f",sin(x));

}

**Output:-**

****

**8.2 Write a program to determine and display the maximum height reached when the ball is thrown at 5 miles / hour at an angle of 60 degrees. (Hint: Make sure to convert the initial velocity into the correct units.) The maximum height reached by a ball thrown with an initial velocity v in feet/sec at an angle of θ is given by the formula height = (0.5 \* v2 \* sin2 θ ) / 32.2. Run the program for v = 7 miles / hour and angle = 45 degrees.**

#include<stdio.h>

#include<math.h>

void main()

{

float r=45,h,a;

printf("hour and angle r=%.2f\n",r);

h = (0.5 \* (7\*7) \* sin(45\*45)) / 32.2;

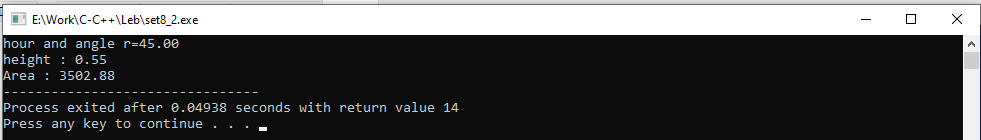
a=3.14\*(r\*r)\*h;

printf("height : %.2f\n",h);

printf("Area : %.2f",a);

}

**Output:-**

****

**8.3 Write a program to calculate and print the height (h) = L \* sin θ , where L is the Length of the Ladder, and θ is the angle the ladder makes with the horizontal base. Data: (a) L = 20 feet, θ = 85o , (b) L = 25 feet, θ = 75o .**

#include<stdio.h>

#include<math.h>

void main()

{

int ch;

float l,h;

printf("(1) L = 20 feet, thita = 85\n");

printf("(2) L = 25 feet, thita = 75\n");

printf("Enter choice:\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

l=20;

h= l \* sin(85);

printf("height :%.2f",h);

break;

case 2:

l=25;

h= l \* sin(75);

printf("height :%.2f",h);

break;

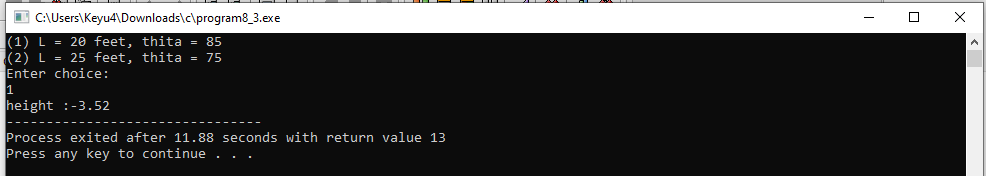
default:

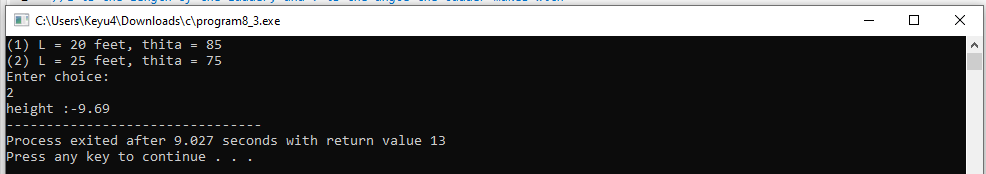
printf("Give valid choice..!");

}

}

**Output:-**





**8.4 Write a program that calculates the x and y coordinates of the point whose polar coordinates are r = 10 and θ = 30o , using the following formulas: x = r cos θ and y = r sin θ . Run the program again to convert polar coordinates: r = 12.5 and θ = 67.8° into rectangular coordinates.**

#include<stdio.h>

#include<math.h>

void main()

{

float r=12.5,x,y;

x=r\*cos(67.8);

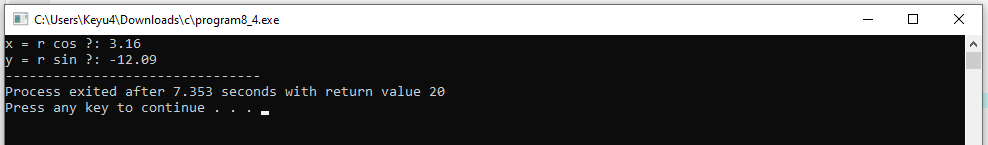
y=r\*sin(67.8);

printf("%.2f",x);

printf("\n%.2f",y);

}

**Output:-**



**SET:-9**

**9.1 Given an integer n >= 1, develop an algorithm and write a program to find the smallest exact divisor of n other than one.**

#include<stdio.h>

#include<math.h>

void main()

{

int i,n;

printf("Enter number:");

scanf("%d",&n);

for(i=2;i<=sqrt(n);i++)

{

if(n%i==0)

{

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

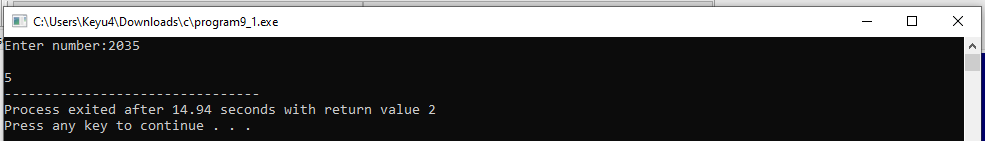
break;

}

}

}

**Output:-**



**9.2 Every integer can be expressed as a product prime numbers. Develop an algorithm and write a program to compute all the prime factors of a given integer n > 0.**

#include<stdio.h>

void main()

{

int i,j,n,f=1;

printf("Enter number:");

scanf("%d",&n);

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

{

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

{

if(i%j==0)

{

break;

}

}

if(j==i)

{

printf("\n%d",j);

f=f\*j;

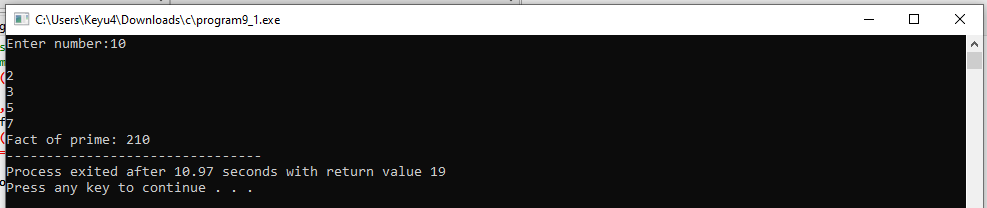
}

}

printf("\nFact of prime: %d",f);

}

**Output:-**

****

**SET:-10**

**10.1 Write a program in C (a) to find factorial of n (n!), and (b) first n terms of the Fibonacci sequence using an iterative algorithm.**

#include<stdio.h>

int factn(int n)

{

int i=1,f=1;

fact:

if(i<=n)

{

f=f\*i;

i++;

goto fact;

}

return f;

}

int fibo(int n)

{

int i=0,a=0,b=1,c;

printf("\n");

printf("Fibonacci sequence :\n");

printf("%d ",a);

printf("%d ",b);

fibo:

if(i<n-2)

{

c=a+b;

printf("%d ",c);

a=b;

b=c;

i++;

goto fibo;

}

return 0;

}

void main()

{

int n;

char ch;

printf(" Menu:\n");

printf ("-------------------------------------------\n");

printf(" (a) to find factorial of n (n!) \n");

printf(" (b) first n terms of the Fibonacci sequence \n");

printf ("-------------------------------------------\n");

printf(" Enter your choice:");

scanf("%c",&ch);

printf ("-------------------------------------------\n");

switch(ch)

{

case 'a':

printf(" Enter N:");

scanf("%d",&n);

printf("factorial of %d : %d",n,factn(n));

break;

case 'b':

printf(" Enter N:");

scanf("%d",&n);

fibo(n);

break;

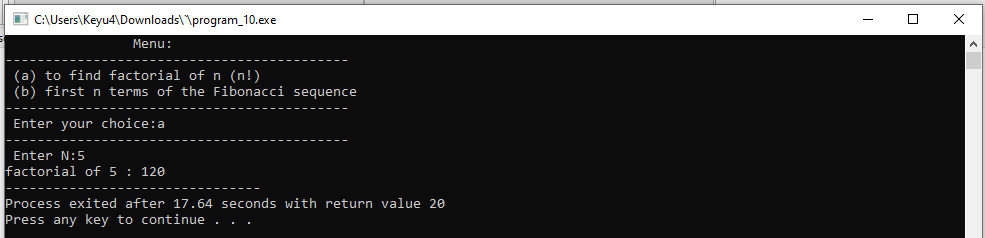
default:

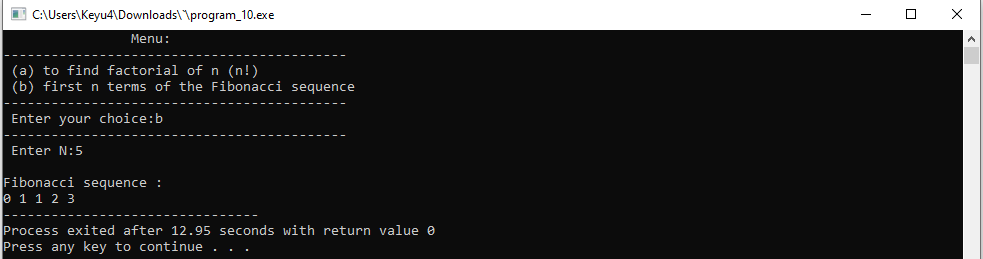
printf("Invalid choice...!");

}

}

**Output:-**



****

**10.2 Explain Recursion formulation of a problem and its working by taking 3 examples: (a) n! ; (b) fibo(n); and (c) sum of a0 + a1 + … + an-1 + an .**

#include<stdio.h>

int factn(int n)

{

if(n==0)

{

return 1;

}

else

{

return n \* factn(n-1);

}

}

int fibon(int n)

{

if(n==0)

{

return 0;

}

else if(n==1)

{

return 1;

}

else

{

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

}

}

int sumn(int n)

{

if(n==1)

{

return 1;

}

else

{

return n + sumn(n-1);

}

}

void main()

{

int i,n;

printf(" Enter N:");

scanf("%d",&n);

factn(n);

printf(" Fibonacci Series: ");

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

{

printf("%d ",fibon(i));

}

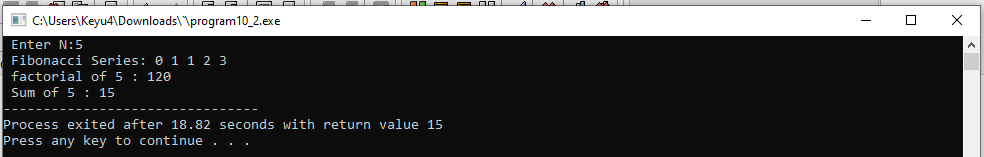
fibon(n);

printf("\n factorial of %d : %d",n,factn(n));

printf("\n Sum of %d : %d",n,sumn(n));

}

**Output:-**

****

**10.3 Write a program in C (a) to find factorial of n (n!); (b) first n terms of the Fibonacci sequence using an iterative algorithm; and (c) sum of a0 + a1 + … + an-1 + an**

**Output:-**

#include<stdio.h>

int factn(int n)

{

if(n==0)

{

return 1;

}

else

{

return n \* factn(n-1);

}

}

int fibon(int n)

{

if(n==0)

{

return 0;

}

else if(n==1)

{

return 1;

}

else

{

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

}

}

int sumn(int n)

{

if(n==1)

{

return 1;

}

else

{

return n + sumn(n-1);

}

}

void main()

{

int i,n;

printf(" Enter N:");

scanf("%d",&n);

factn(n);

printf("Fibonacci Series: ");

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

{

printf("%d ",fibon(i));

}

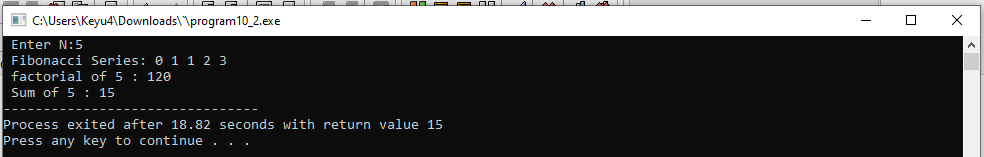
fibon(n);

printf("\nfactorial of %d : %d",n,factn(n));

printf("\nSum of %d : %d",n,sumn(n));

}

**Output:-**



**SET:-11**

**11.1 Program : Input: a, b Output: a (prints the value of b), b (prints the value of a).**

#include<stdio.h>

void main()

{

int a,b;

printf("Enter a:");

scanf("%d",&a);

printf("Enter b:");

scanf("%d",&b);

a=a+b;

b=a-b;

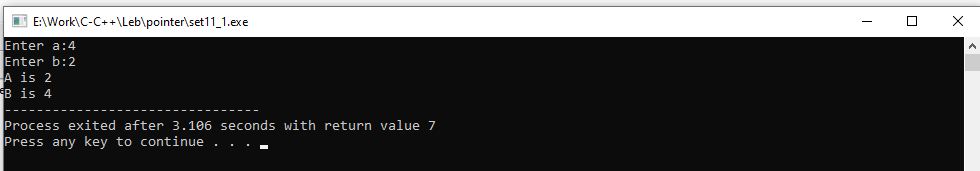
a=a-b;

printf("A is %d",a);

printf("\nB is %d",b);

}

**Output:-**



**11.2 Implement the program in C for “exchanging the values of two variables” using function (which will require use of pointers for function arguments in C).**

#include<stdio.h>

int swap(int \*a,int \*b)

{

\*a=\*a+\*b;

\*b=\*a-\*b;

\*a=\*a-\*b;

}

void main()

{

int a,b;

printf("Enter a:");

scanf("%d",&a);

printf("Enter b:");

scanf("%d",&b);

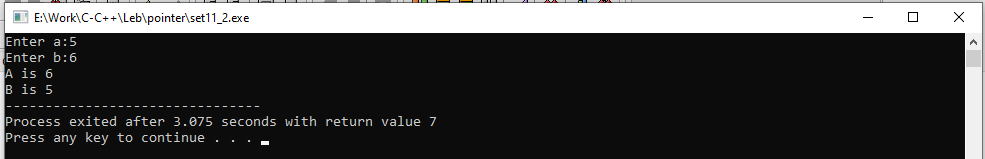
swap(&a,&b);

printf("A is %d",a);

printf("\nB is %d",b);

}

**Output:-**



**11.3 Write a C program to find sum of n values ai , i = 1 to n, using pointers instead of arrays.**

#include<stdio.h>

int sumn(int \*n)

{

int i;

int sum=0;

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

{

sum = sum + i;

}

return sum;

}

void main()

{

int a[10];

int n;

int sum;

printf("Enter N:");

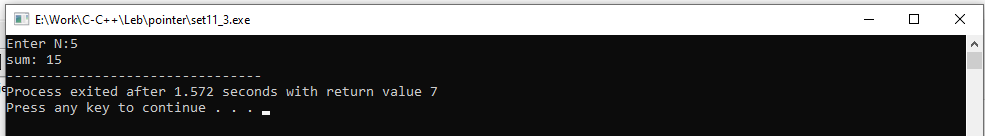
scanf("%d",&n);

sum=sumn(&n);

printf("sum: %d",sum);

}

**Output:-**



**11.4 Write a C program to count number of words in a given text by representing text string as pointer instead of array.**

#include<stdio.h>

int wordcount(char \*s)

{

int i,c=1;

for(i=0;s[i]!='\0';i++)

{

if(s[i]==' ')

{

c++;

}

}

return c;

}

void main()

{

char s[20];

int c;

printf("Enter string:");

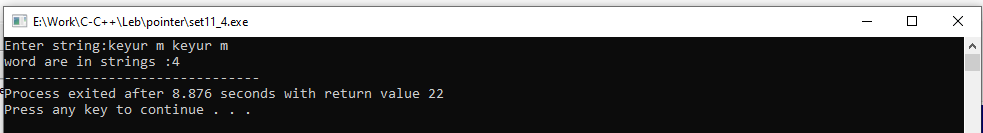
gets(s);

c=wordcount(s);

printf("word are in strings :%d",c);

}

**Output:-**



**SET:-12**

**12.1 Remove all duplicates from an ordered array and contract the array accordingly.**

#include<stdio.h>

void main()

{

int a[10],i,j,k,n,temp=0;

printf("Enter Array size:\n");

scanf("%d",&n);

printf("Enter Array:\n");

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

{

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

}

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

{

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

{

if(a[i]==a[j])

{

for(k = j; k < n; k++)

{

a[k] = a[k + 1];

}

j--;

n--;

}

}

}

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

{

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

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("Array is:");

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

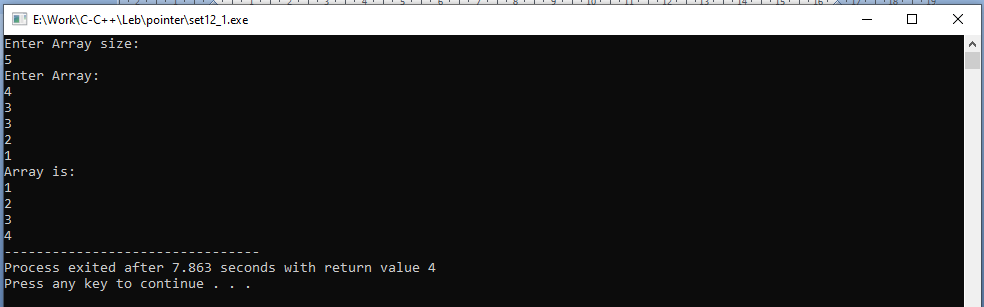
{

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

}

}

**Output:-**



**SET:-13**

**13.1 Given a number n >= 0, develop an algorithm and write a program in C to compute square root of a given non-negative number (n >= 0) by Divide-and-Conquer method**

#include <stdio.h>

void main(){

int x;

printf("Enter number:");

scanf("%d",&x);

printf("Result: %d",sroot(x));

}

int sroot(int x){

int start=1,end=x,ans;

if(x==0||x==1){

return x;

}

while(start<=end){

int mid=(start+end)/2;

if(mid\*mid<x){

start=mid+1;

ans=mid;

}else{

end = mid - 1;

}

if(mid\*mid==x){

return mid;

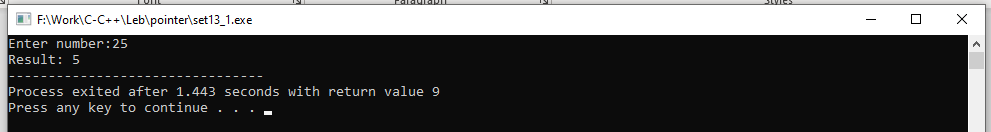
}

}

return ans;

}

**Output:-**



**13.2 Write a program in C using an improved algorithm to compute Square Root using Newton’s method and other methods.**

#include <stdio.h>

// Function to get absolute value of the number given by user.

float absolute(float num){

if(num < 0){

num = -num;

}

return num;

}

// Function to calculate square root of the number using Newton-Raphson method

float square\_root(int x){

const float difference = 0.00001;

float guess = 1.0;

while(absolute(guess \* guess - x) >= difference){

guess = (x/guess + guess)/2.0;

}

return guess;

}

void main(){

int number;

float root;

printf("Enter a number: ");

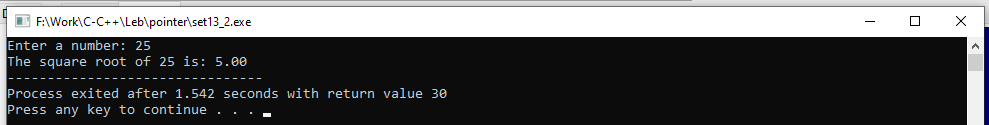
scanf("%i", &number);

root = square\_root(number);

printf("The square root of %i is: %.2f", number, root);

}

**Output:-**



**SET:-14**

**14.1 Write a C program to find Maximum and Minimum values in a given array (or List) of values.**

#include<Stdio.h>

void maxmin(int a[],int n,int min,int max)

{

int i;

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

{

if(max<a[i])

{

max=a[i];

}

if(min>a[i])

{

min=a[i];

}

}

printf("Max no is:%d",max);

printf("\nMin no is:%d",min);

}

void main()

{

int a[20];

int i,max,min,n;

printf("Enter Array size:\n");

scanf("%d",&n);

printf("Enter Array:\n");

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

{

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

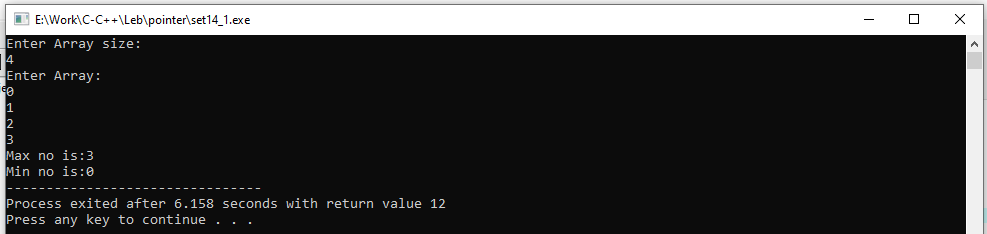
}

min=max=a[0];

maxmin(a,n,min,max);

}

**Output:-**



**14.2** **Write a C program to find Maximum and Minimum values in a given array (or List) of values. Also write the C program using pointers instead of array.**

#include<Stdio.h>

void maxmin(int \*a,int n,int \*min,int \*max)

{

int i;

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

{

if(\*max<a[i])

{

\*max=a[i];

}

if(\*min>a[i])

{

\*min=a[i];

}

}

printf("Max no is:%d",\*max);

printf("\nMin no is:%d",\*min);

}

void main()

{

int a[20];

int i,max,min,n;

printf("Enter Array size:\n");

scanf("%d",&n);

printf("Enter Array:\n");

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

{

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

}

min=max=a[0];

maxmin(a,n,&min,&max);

}

**Output:-**

