1. **Leap Year or Not**

#include <stdio.h>

int main()

{

int year;

printf("Enter a year: ");

scanf("%d",&year);

if(year%4 == 0)

{

if( year%100 == 0)

{

// year is divisible by 400, hence the year is a leap year

if ( year%400 == 0)

printf("%d is a leap year.", year);

else

printf("%d is not a leap year.", year);

}

else

printf("%d is a leap year.", year );

}

else

printf("%d is not a leap year.", year);

return 0;

}

#include <stdio.h>

int main()

{

int year;

scanf("%d",&year);

if( year%100 == 0)

{

// year is divisible by 400, hence the year is a leap year

if ( year%400 == 0)

printf("%d is a leap year.", year);

else

printf("%d is not a leap year.", year);

}

else if(year%4==0)

printf("%d is a leap year.", year );

else

printf("%d is not a leap year.", year);

return 0;

}

Example of leap year : 1996, 2004, 2008 and 2000,1600 etc 2100 is not considered as leap year here

#include<stdio.h>

void main()

{

intyear;

printf(“Enter a year”);

scanf(“%d”,&year);

(year%400 ==0 ||( year%4==0 && year%100!=0))

?printf(“Leap Year”)

:printf(“Non Leap Year”);

}

Output:

Enter a year

2016

Leap Year

Enter a year

2015

Non Leap Year

Enter a Year

2000

Leap Year

Enter a Year

1900

Non Leap Year

1. **/ Print second largest number of three number given as input**

**Identify the correction required**

#include <stdio.h>

int main()

{

double n1, n2, n3;

printf("Enter three different numbers: ");

scanf("%lf %lf %lf", &n1, &n2, &n3);

if( n1<n2 && n1>n3 )

printf("%.2f is the largest number.", n1);

if( n2<n1 && n2>n3 )

printf("%.2f is the largest number.", n2);

if( n3<n1 && n3>n2 )

printf("%.2f is the largest number.", n3);

return 0;

}

**Corrected Code**

#include <stdio.h>

int main()

{

double n1, n2, n3;

printf("Enter three different numbers: ");

scanf("%lf %lf %lf", &n1, &n2, &n3);

if( (n1<n2 && n1>n3) || (n1>n2 && n1<n3) )

printf("%.2f is the largest number.", n1);

if( (n2<n1 && n2>n3 ) || (n2>n1 && n2<n3 ))

printf("%.2f is the largest number.", n2);

if( (n3<n1 && n3>n2 ) || (n3>n1 && n3<n2 )))

printf("%.2f is the largest number.", n3);

return 0;

}

**Logic 2**

#include <stdio.h>

#include <stdlib.h>

int main(int argc,char \*argv[])

{

int a,b,c,lar,seclar;

// scanf("%d%d%d",&a,&b,&c);

a=atoi(argv[1]);

if(a>b)

{

if(a>c)

lar=a;

else

seclar=a;

}

if(b>c)

{

if(b>a)

lar=a;

else

seclar=b;

}

if(c>a)

{

if(c>b)

lar=c;

else

seclar=c;

}

printf("second largest number :%d",seclar);

}

1. **Write a C program to swap two numbers without using a temporary variable.**

#include<stdio.h>

#include<conio.h>

void main()

{

int a,b;

printf("Enter two numbers to be swapped\n");

scanf("%d%d",&a,&b);

printf("\nBefore swapping a: %d and b:%d", a, b);

a=a+b;

b=a-b;

a=a-b;

printf("\nAfter swapping a: %d and b:%d", a, b);

getch();

}

OUTPUT:

Enter twonumbers to be swapped

1. 5

Before swapping a: 3 and b:5

After swapping a: 5 and b:3

1. **Area of circle**

#include<stdio.h>

#define PI 3.14

void main()

{

float A,r;

scanf(“%f”, &r);

A=PI\*r\*r;

printf("%f”, A);

}

1. **Circumference of circle**

#include<stdio.h>

#define PI 3.14

void main()

{

float C,r;

scanf(“%f”, &r);

C=2\*PI\*r;

printf("%f”, C);

}

1. **Arc Length of a Circle**

#include<stdio.h>

#define PI 3.14

void main()

{

float angle,arclen,r;

scanf(“%f”, &r);

scanf(“%f”, &angle);

arclen =2\*PI\*r\*(angle/360);

printf("%f”, arclen);

}

1. **Hypotenuse of right angled triangle**

#include<stdio.h>

#include<Math.h>

void main()

{

float hypo,s1,s2;

scanf(“%f”, &s1);

scanf(“%f”, &s2);

hypo =sqrt( (s1\*s1)+(s2\*s2) );

printf("%f”, hypo);

}

**8.**

**Write a program to check equivalence of two numbers**

void main()

{

int m,n;

printf(“enter two numbers”);

scanf(“%d%d”,&m,&n)

if(m-n==0)

{

printf(“\n two numbers are equal”);

}

getch();

}

Output:

enter two numbers 5 5

two numbers are equal

**Write a program for comparison of two numbers**

void main()

{

int x,y;

printf(“Enter two numbers”);

scanf(“%d%d”,&x,&y);

if(x>=y)

{

if(x>y)

printf(“\n%d is greater than %d”,x,y);

else

printf(“\n%d is equal to %d”,x,y);

}

else

printf(“\n%d is less than %d”,x,y);

}

Output:

Enter two numbers 10 20

10 is less than 20

Enter two numbers 50 40

50 is greater than 40

if (a>b&&a>c&&a>b)

printf(“a is big”);

else if (b>c&&b>d)

printf(“b is big”);

else if (c>d)

printf (“c is big”);

else

printf(“d is big”);

1. **Write a program to check whether the candidate age is greater than 17 or not .if yes display the message “eligible for voting”**

void main()

{

int age;

printf(“\n enter age”);

scanf(“%d”,age);

if(age>17)

prinft(“eligible for voting”);

getch();

}

Output:

enter age 18

eligible for voting

1. **Write a program to find whether the given number is positive or negative**

void main()

{

int n;

printf(“enter a number\n”);

scanf(“%d”,&n);

if(n>0)

printf(“the number is positive \n”);

getch();

}

**Output:**

enter a number 10

the number is positive

enter a number -8

1. **Write a program for comparison of two numbers**

void main()

{

int n1,n2;

printf(“enter two numbers”);

scanf(“%d%d”,&n1,&n2);

if(n1>n2)

printf(“%d is greater than %d”,n1,n2);

if(n1<n2)

printf(“%d is less than %d”,n1,n2);

if(n1==n2)

printf(“%d is equal to %d”,n1,n2);

getch();

}

**Output:**

Enter two numbers 10 12

10 is less than 12

Enter two numbers 15 11

15 is greater than 11

1. **Write a program to check whether the candidate age is greater than 17 or not .if yes display the message “eligible for voting”**

void main()

{

int age;

printf(“\n enter age”);

scanf(“%d”,age);

if(age>17)

prinft(“\neligible for voting”);

else

prinft(“\nNot eligible for voting”);

getch();

}

Output:

enter age 18

eligible for voting

enter age 18

Not eligible for voting

1. **Write a program to find whether the given number is positive or negative**

void main()

{

int n;

printf(“enter a number\n”);

scanf(“%d”,&n);

if(n>0)

printf(“the number is positive \n”);

else

printf(“the number is not positive \n”);

getch();

}

**Output:**

enter a number 10

the number is positive

enter a number -8

the number is not positive

1. **Write a program to find the roots of a quadratic equation by using if else condition**

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

int a,b,c;

float r1,r2,desc;

printf(“Enter values for a,b,c”);

scanf(“%d %d%d”,&a,&b,&c);

desc=b\*b-4\*a\*c;

if(desc>0)

{

r1=(-b+sqrt(desc))/(2\*a);

r2=(-b-sqrt(desc))/(2\*a);

printf(“\n r1=%.2f,r2=%.2f”,r1,r2);

}

else if(desc==0)

{

r1=-b/(2.0\*a);

printf(“\n r1=%.2f”,r1);

}

else

printf(“\nRoots are imaginary”);

getch()

}

Output:

Enter values for a b c 5 1 5

Roots are imaginary

Enter values for a b c 1 -2 1

r1=1.00000,r2=-1.00000

1. **Write a program to find whether the given number is even or odd**

void main()

{

int n;

printf(“Enter a number to check”);

scanf(“%d”,&n);

if(n%2==0)

printf(“\n%d is an even number “,n);

else

printf(“\n%d is an odd number”,n);

getch();

}

Output:

Enter a number to check 7

7 is an odd number

Enter a number to check 4

4 is an even number

1. **Write a program to find the biggest among two numbers**

Void main()

{

*i*nt x,y,big;

printf(“Enter two numbers”);

scanf(“%d%d”,&x,&y);

if(x>y)

big=x;

else

big=y;

printf(“\nBiggest number %d”,big);

getch();

}

Output:

Enter two numbers 5 6

Biggest number 6

1. **Write a program to check whether a given character is vowel or not.**

void main()

{

char ch;

printf(“enter any character”);

scanf(%c”,&ch);

if(ch==’a’||ch==’e’||ch==’I’||ch==’o’||ch==’u’||ch==’A’||ch==’E’||ch==’I’||ch==’O’||ch==’U’)

printf(“ you entered vowel”);

else

printf(“you didn’t entered vowel”);

getch();

}

Output:

Enter any character e

You entered vowel

if(a>b)

{

if(a>c)

printf(“a is big”);

else

printf(“c is big”);

}

else if (b>c)

{

printf(“b is big”);

}

else

{

printf(“c is big”);

}

**Write a program to calculate area of square/rectangle/circle/triangle depending upon user choice.**

void main()

{

int choice;

float area,a,b,c,s;

printf(“main menu \n”);

printf(“1. area of square\n”);

printf(“2.area of rectangle\n”);

printf(“3.area of circle\n”);

printf(“4.area of triangle\n”);

printf(“enter your choice\n”);

scanf(“%d”,&choice);

if(choice==1)

{

printf(“enter the side of square”);

scanf(“%f”,&a);

area=a\*a;

printf(“area of square is %f”,area);

}

else if(choice==2)

{

printf(“enter the length and breadth of rectangle”);

scanf(“%f%f”,&a,&b);

area=a\*b;

printf(“area of rectangle is %f”,area);

}

else if(choice==3)

{

printf(“enter the radius of circle”);

scanf(“%f”,&a);

area=3.14\*a\*a;

printf(“area of circle is %f”,area);

}

else if(choice==4)

{

printf(‘enter three sides of triangle”);

scanf(“%f%f%f”,&a,&b,&c);

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

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

printf(“area of triangle is %f”,area);

}

else

printf(“wrong choice\n”);

getch();

}

Output:

main menu

1.area of square

2.area of rectangle

3.area of circle

4.area of triangle

enter your choice 2

enter the length and breadth of rectangle 4 6

area of rectangle is 24

19.

**Write a program that reads marks in three subjects ,calculate average marks and assigns grade as per following specifications**

|  |  |
| --- | --- |
| If marks | Then grade |
| >=90  75-90  60-75  50-60  <50 | Distinction  A  B  C  Fail |

void main()

{

int marks;

printf(“Enter the marks”);

scanf(“%d’,&marks);

if(marks>=90)

printf(“\nGrade is Distinction”);

else if(marks>=75)

printf(“\nGrade is A”);

else if(average>=60)

printf(“grade is B”);

else if(marks>=50)

printf(“\nGrade is C”);

else

printf(“\nFail”);

getch();

}

Output:.

Enter the marks90

Grade is Distinction

1. **Write a program to provide multiple functions such as 1.addition 2.subtraction 3.multiplication 4.division 5.remainder using switch statement. /\* Calculator Program \*/**

void main()

{

char opt;

int a,b,c;

printf(“Enter two numbers”);

scanf(“%d%d”,&a,&b);

printf(“\nEnter the operation to be performed”);

printf(“\n+\n-\n\*\n/\n%\n:”);

scanf(“%d”,&opt);

switch(opt)

{

case 1: c=a+b;

printf(“\nAddition Result:%d”,c);

break;

case 2:c=a-b;

printf(“\nSubtraction Result:%d”,c);

break;

case 3: c=a\*b;

printf(“\nMultiplication Result:%d”,c);

break;

case 4:c=a/b;

printf(“\nDivision Result:%d”,c);

break;

case 5:c=a%b;

printf(“\nRemainder Result:%d”,c);

break;

default:printf(“\nInvalid choice”);

}

}

Output:

Enter two numbers

3 4

Enter the operation to be performed

+

-

\*

/

%

:\*

Multiplication Result:12

1. **Write a program to read a day number and print corresponding day of the week**

void main()

{

int num;

printf(“Enter any number in between 1 to 7”);

scanf(“%d”,&num);

switch(num)

{

case 1:printf(“\nIt is Sunday”);

//break;

case 2:printf(“\nIt is Monday”);

break;

case 3: printf(“\nIt is Tuesday”);

break;

case 4: printf( “\nIt is wedensday”);

break;

case 5: printf(“\nIt is Thursday”);

break;

case 6: printf(“\nIt is Friday”);

break;

case 7:printf(“\nIt is Saturday”);

break;

default: print(“\nWrong number “);

}

getch();

}  
output:

Enter any number in between 1 to 7 3

It is Tuesday.

1. **Write a program to read the month number and print corresponding month’s name.**

void main()

{

int m;

printf(“Enter the month number from 1to 12”);

scanf(“%d”,&m);

switch(m)

{

case 1:printf(“\nIt is January”);

break;

case 2:printf(“\nIt is February”);

break;

case 3: printf(“\nIt is March”);

break;

case 4: printf( “\nIt is April”);

break;

case 5: printf(“\nIt is May”);

break;

case 6: printf(“\nIt is June”);

break;

case 7:printf(“\nIt is July”);

break;

case 8:printf(“\nIt is August”);

break;

case 9:printf(“\nIt is September”);

break;

case 10: printf(“\nIt is October”);

break;

case 11: printf( “\nIt is November”);

break;

case 12: printf(“\nIt is December”);

break;

default: printf(“\nInvalid Month Number “);

}

getch();

}  
output:

Enter the month number from 1to 12 6

It is June.

1. **Switch statement without break statement**

void main()

{

int x;

printf(“enter any value”);

scanf(“%d”,&x);

switch(x)

{

case 1: printf(“tenali”);

case 2:printf(“vizag”);

case 3:printf(“Hyderabad”);

break;

}

getch();

}

Output: enter any value 1

tenali

vizag

Hyderabad

1. **Write a program to read the month number and print corresponding month’s no. of days.**

void main()

{

int m;

printf(“Enter the month number from 1to 12”);

scanf(“%d”,&m);

switch(m)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:printf(“\nNo. Of days are 31 days”);

break;

case 8:

case 9:

case 10:

case 11: printf( “\nNo. Of days are 30 days”);

break;

case 2: printf(“\nNo. Of days are 28 days”);

break;

default: printf(“\nInvalid Month Number “);

}

getch();

}  
output:

Enter the month number from 1to 12 9

No. Of days are 30 days

1. **Program for temperature conversion as per user choice**

void main()

{

int choice;

float tempf,tempc;

printf(“temperature conversion menu\n”);

printf(“1. Faherenheit to Celsius \n”);

printf(“2. Celsius to faherenheit \n”);

printf(“enter your choce”);

scanf(“%d’,&choice);

if(choice==1)

{

printf(“enter the temperature in Fahrenheit”);

scanf(“%f”,&tempf);

tempc=(tempf-32)/1.8;

printf(“temperature in Celsius is %f”,tempc);

}

else if(choice==2)

{

printf(“enter the temperature in Celsius”);

scanf(“%f”,&tempc);

tempf=(tempc\*1.8)+32;

printf(“temperature in Fahrenheit %f”,tempf);

}

else

printf(“wrong choce”);

getch();

}

Output:

Temperature conversion menu

1.fahrenheit to Celsius

2.celsius to Fahrenheit

Enter your choice 2

Enter the temperature in Celsius 37

Temperature in Fahrenheit 98.59999

1. **Reverse of three digit number without using loop**

int main()

{

int num,n1n2,n3,rev;

scanf(“%d’,&num);

n1=num/100;

n2=(num%100)/10;

n3=num%10;

rev=n3\*100+n2\*10+n1;

printf(“%d”,rev);

return 0;

}

1. **Program to check whether a character is an uppercase or lowercase alphabet or a digit or a special symbol**

#include<stdio.h>

#include<conio.h>

void main()

{

char ch;

printf(“enter any character”);

scanf(“%c”,&ch);

if(ch>=’A’&&ch<=’Z’)

printf(“you entered an uppercase alphabet”);

else if(ch>=’a’&&ch<=’z’)

printf(“you entered a lowercase character”);

else if(ch>=’0’ &&ch<=’9’)

printf(“you entered a digit”);

else

printf(“you entered a special character”);

getch();

}

Output:

enter any character y

you entered a lowercase character

1. **/\* C Program to print numbers from 1 to n using while loop\*/**

#include<stdio.h>  
#include<conio.h>  
 void main()   
 {   
 int n,i;   
 printf("Enter the number:\n");   
 scanf("%d",&n);

i=1;

printf(”\n The numbers from 1 to %d are \n”,n);  
 while(i<=n)   
 {   
 printf("%d\t",i);   
 i++;   
 }   
 getch();   
 }

**Output:**

Enter the number:

5

The numbers from 1 to 5 are

1 2 3 4 5

1. **/\* C Program to print numbers from n to 1 using while loop\*/**

#include<stdio.h>  
#include<conio.h>  
 void main()   
 {   
 int n,i;   
 printf("Enter the number:\n");   
 scanf("%d",&n);

i=n;

printf(”\n The numbers from %d to 1 are \n”,n);  
 while(i>=1)   
 {   
 printf("%d\t",i);   
 i--;   
 }   
 getch();   
 }

**Output:**

Enter the number:

5

The numbers from 1 to 5 are

5 4 3 2 1

1. **/\* C Program to print numbers from even numbers from 1 to n using while loop\*/**

#include<stdio.h>  
#include<conio.h>  
 void main()   
 {   
 int n,i;   
 printf("Enter the number:\n");   
 scanf("%d",&n);

i=1;

printf(”\n The even numbers from 1 to %d are \n”,n);  
 while(i<=n)   
 {

if(i%2==1)

{  
 printf("%d\t",i);

}  
 i++;   
 }   
 getch();   
 }

**Output:**

Enter the number:

5

The even numbers from 1 to 5 are

2 4

Example:

1. **/\* C Program to find factorial of a number using while loop\*/**

#include<stdio.h>  
#include<conio.h>  
 void main()   
 {   
 int n,f=1;   
 printf("Enter the number:\n");   
 scanf("%d",&n);   
 while(n>0)   
 {   
 f=f\*n;   
 n--;   
 }   
 printf("The factorial of the integer is:%d",f);   
 getch();   
 }

**Output:**

Enter the number:

5

The factorial of the integer is:

120

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])   
 {   
 int n,f=1;

n=atoi(argv[1]);  
 while(n>0)   
 {   
 f=f\*n;   
 n--;   
 }   
 printf("The factorial of the integer is:%d",f);   
 return 0;

}

1. **/\*Write a C program to find the product of 4 integers entered by a user. If user enters 0 skip it.\*/**

# include <stdio.h>

int main()

{

int i,num,product;

for(i=1,product=1;i<=4;++i)

{

printf("Enter num%d:",i);

scanf("%d",&num);

if(num==0)

continue;

product\*=num;

}

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

return0;

}

/ \*In this program, when num equals to zero, it skips the statement product\*=num and continue the loop. \*/

**Output**

Enter num1: 3

Enter num2: 0

Enter num3: -5

Enter num4: 2

product= - 30

APPLICATIONS OF LOOPS:

1. **/\*Write a C program to print the Fibonacci series up to n terms\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

*i*nt f0,f1,f2,i,n;

printf(“Enter the required no. of terms in the Fibonacci series”);

scanf(“%d”,&n);

f0=0;

f1=1;

printf(“\n The Fibonacci series up to %d terms is…\n”,n);

printf(“%d\t%d”,f0,f1);

i=2;

while(i<=n)

{

f2=f0+f1;

printf(“%d\t”,f2);

f0=f1;

f1=f2;

i++;

}

getch();

}

**OUTPUT**

Enter the required no. of terms in the Fibonacci series8

The Fibonacci series up to 8 terms is…

0 1 1 2 3 5 8 13

1. **/\*Write a C program to print sum of individual digits of a number, reverse of the number and check if the given number is palindrome or not\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

*i*nt n,k,rev=0,sum=0;

printf(“Enter a number”);

scanf(“%d”,&n);

k=n;

while(n>0)

{

rev=rev\*10+n%10;

sum=sum+n%10;

n=n/10;

}

printf(“\nThe Sum of individual digits of the given number is %d”, sum);

printf(“\nThe reverse of the given number is %d”, rev);

if(k==rev)

{

printf(“\n The given number %d is Palindrome”, k);

}

else

{

printf(“\n The given number %d is not Palindrome”, k);

}

getch();

}

Output:

Enter a number

7892

The Sum of individual digits of the given number is 26

The reverse of the given number is 2987

The given number 7892 is not Palindrome

Enter a number

4994

The Sum of individual digits of the given number is 26

The reverse of the given number is 4994

The given number 4994 is not Palindrome

1. **/\*Write a C program to check if the given number is amstrong or not\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

*i*nt n,num,r,sum=0;

printf(“Enter a number”);

scanf(“%d”,&num);

n=num;

nd=0;

while(num!=0)

{

num=num/10;

nd++;

}

num=n;

sum=0;

while(num!=0)

{

r=num%10;

sum=sum+pow(r,nd);

num=num/10;

}

if(n==sum)

{

printf(“\n The given number %d is Amstrong”, k);

}

else

{

printf(“\n The given number %d is not Amstrong”, k);

}

getch();

}

**Output**

Enter a number

7892

The given number 7892 is not Amstrong

Enter a number

153

The given number 153 is Amstrong

1. **/\*Write a C program to print the multiplication table of a given number up to a another specific number. \*/**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,n,m;

printf(“Enter a number for multiplication table”);

scanf(“%d”,&n);

printf(“\nEnter the number up to which you want the multiplication table”);

scanf(“%d”,&m);

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

{

printf(“\n%d\*%d=%d”,n,i,i\*n);

}

getch();

}

Output:

Enter a number for multiplication table

3

Enter the number up to which you want the multiplication table

5

3\*1=3

3\*2=6

3\*3=9

3\*4=12

3\*5=15

1. **// c program to find m power n**

#include<stdio.h>

#include<conio.h>

void main()

{

*i*nt I,n,m,product=1;

printf(“enter the value of M”);

scanf(“%d”,&m);

printf(“enter the value of N”);

scanf(“%d”,&n);

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

{

product =product\*m;

}

printf(“the result is %d”,product);

getch();

}

**OUTPUT**

enter the value of M

5

enter the value of N

3

The result is 125

1. **//Sum of the elements in an array**

#include<stdio.h>

#include<conio.h>

void main(int argc, char \*agv[])

{

int A[20],i,sum=0,n; /\*array declaration\*/

n=atoi(argv[1]);

for(i=0;i<n;i++) /\*reading values for array\*/

{

A[i]=atoi(argv[2+i];

}

for(i=0;i<n;i++) /\*accessing the values of an array\*/

{

sum=sum+A[i];

}

printf("Sum of elemets in array is %d",sum);

getch();

}

**Ouput:**

Enter the no. of elements in the array

5

Enter A[0]

10

Enter A[1]

20

Enter A[2]

30

Enter A[3]

40

Enter A[4]

50

Sum of elements is 150

Review Question:

Find out Average of n numbers.

1. **//minimum and maximum elements in Array**

#include<stdio.h>

#include<conio.h>

void main()

{

int min,max,i,a[20],n;

printf(“Enter the no. of elements in the array”);

scanf(“%d”,&n);

for(i=0;i<n;i++) //reading array elements

{

printf("Enter the A[%d] elements ",i);

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

}

Max=min=a[0];

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

{

if(a[i]<min) //checking for minimum element

min=a[i];

if(a[i]>max) //checking for maximum element

max=a[i];

}

printf("\n Minimum element is %d",min);

printf("\n Maximum element is %d",max);

getch();

}

**Ouput:**

Enter the no. of elements in the array

5

Enter A[0]

10

Enter A[1]

20

Enter A[2]

30

Enter A[3]

40

Enter A[4]

50

Maximum element is 50

Minimum element is 10

1. **Write a C Program to find minimum and maximum elements in array and exchange their positions**

#include<stdio.h>

#include<conio.h>

void main()

{

int min,max,i,a[20],n,max\_pos,min\_pos,temp;

printf(“Enter the no. of elements in the array”);

scanf(“%d”,&n);

for(i=0;i<n;i++) //reading array elements

{

printf("Enter the A[%d] elements ",i);

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

}

Max=min=a[0];

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

{

if(a[i]<min) //checking for minimum element

{

min=a[i];

min\_pos=i;

}

if(a[i]>max) //checking for maximum element

{

max=a[i];

max\_pos=i;

}

}

temp=a[min\_pos];

a[min\_pos]=a[max\_pos];

a[max\_pos]=temp;

printf("\n Minimum element is %d",min);

printf("\n Maximum element is %d",max);

printf(“\nThe list of elements after exchanging the minimum element and maximum element position are….\n”);

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

{

printf(“%d\t”,a[i]);

}

getch();

}

**Ouput:**

Enter the no. of elements in the array

5

Enter A[0]

10

Enter A[1]

20

Enter A[2]

30

Enter A[3]

40

Enter A[4]

50

Maximum element is 50

Minimum element is 10

The list of elements after exchanging the minimum element and maximum element position are….

50 20 30 40 10

1. **//Write a C Program to perform linear search on a list of numbers**

#include<stdio.h>

#include<conio.h>

void main()

{

int a[20],i,n,key,flag=0;

printf(“Enter the no. of elements in the array”);

scanf(“%d”,&n);

for(i=0;i<n;i++) //reading array elements

{

printf("Enter the A[%d] element: ",i);

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

}

printf("Enter the key element to be found in the array: ");

scanf("%d",&key);

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

{

if(a[i]==key) //checking if array element is equal to key element

{

flag==1;

break;

}

}

if(flag==1)

printf("\n Key element %d is found at location %d",key,i);

else

printf("\n Key element %d is not found in the array",key);

getch();

}

**Ouput:**

Enter the no. of elements in the array

5

Enter A[0]

10

Enter A[1]

20

Enter A[2]

30

Enter A[3]

40

Enter A[4]

50

Enter the key element to be found in the array:40

Key element 40 is found at location 3

Enter the no. of elements in the array

5

Enter A[0]

10

Enter A[1]

20

Enter A[2]

30

Enter A[3]

40

Enter A[4]

50

Enter the key element to be found in the array:90

Key element 90 not found in the array

1. **//Write a C Program to perform binary search on a list of numbers**

#include<stdio.h>

#include<conio.h>

void main()

{

int a[20],low,mid,high,n,key,flag=0;

printf(“Enter the no. of elements in the array”);

scanf(“%d”,&n);

printf("Enter %d elements in ascending order: ",n);

for(i=0;i<n;i++) //reading array elements

{

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

}

printf("Enter the key element to be found in the array: ");

scanf("%d",&key);

low=0;

high=n-1;

while(low<=high)

{

mid=(low+high)/2; //checking if array element is equal to key element

if(key==a[mid])

{

flag==1;

break;

}

else if(key<a[mid])

{

high=mid-1;

}

else

low=mid+1;

}

if(flag==1)

printf("\n Key element %d is found at location %d",key,mid);

else

printf("\n Key element %d is not found in the array",key);

getch();

}

**Ouput:**

Enter the no. of elements in the array

5

Enter 5 elements in ascending order:

10

20

30

40

50

Enter the key element to be found in the array:40

Key element 40 is found at location 3

Enter the no. of elements in the array

5

Enter 5 elements in ascending order:

10

20

30

40

50

Enter the key element to be found in the array:90

Key element 90 is not found

1. **Write a C program to enter a decimal number and calculate and display the binary equivalent of that number.**

#include<stdio.h>

#include<conio.h>

void main()

{

int bi\_num[20],dec\_num,i,n,flag=0;

printf(“Enter the decimal number”);

scanf(“%d”,&dec\_num);

for(i=0,n=0;dec\_num>0;i++,n++) //reading array elements

{

bi\_num[i]=dec\_num%2;

dec\_num=dec\_num/2;

}

printf("The Binary equivalent of the decimal number %d is : ",n);

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

printf(“%d”,bi\_num[i]);

getch();

}

1. **Write a C program to enter a decimal number and calculate and display the 2’s complement of its binary equivalent.**

#include<stdio.h>

#include<conio.h>

void main()

{

int bi\_num[20],dec\_num,i,n,flag=0;

printf(“Enter the decimal number”);

scanf(“%d”,&dec\_num);

for(i=0,n=0;dec\_num>0;i++,n++) //reading array elements

{

bi\_num[i]=dec\_num%2;

dec\_num=dec\_num/2;

}

printf("The Binary equivalent of the decimal number %d is : ",n);

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

printf(“%d”,bi\_num[i]);

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

{

if((bi\_num[i]==0 && flag==0) || (bi\_num[i]==1 && falg==1))

{

bi\_num[i]=0;

}

else if((bi\_num[i]==1 && flag==0) || (bi\_num[i]==0 && flag==1))

{

bi\_num[i]=1;

flag=1;  
}

}

printf("The 2’s complement of the binary equivalent of the decimal number %d is : ",n);

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

printf(“%d”,bi\_num[i]);

getch();

}

1. **/\* Program to demonstrate two dimensional array.**#include <stdio.h>

#include <conio.h>

void main()

{

int a[3][3], i, j;

printf("\n\t Enter matrix of 3\*3 : ");

for(i=0; i<3; i++)

{

for(j=0; j<3; j++)

{

scanf("%d",&a[i][j]); //read 3\*3 array

}

}

printf("\n\t Matrix is : \n");

for(i=0; i<3; i++)

{

for(j=0; j<3; j++)

{

printf("\t %d",a[i][j]); //print 3\*3 array

}

printf("\n");

}

getch();

}

#### Output :

Enter matrix of 3\*3 : 3 4 5 6 7 2 1 2 3

Matrix is :

3 4 5

6 7 2

1 2 3

**45. Matrix Addition:**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,j,n,a[10][10],b[10][10],c[10][10];

printf("Enter size of the matrix ");

scanf("%d",&n); //reading size of the matrix

printf("Enter Matrix 1 elements ");

for(i=0;i<n;i++) //for reading each row

{

for(j=0;j<n;j++) //for reading columns

{

scanf("%d",&a[i][j]); //reading the elements of matrix 1

}

}

printf("Enter Matrix 2 elements ");

for(i=0;i<n;i++) //for reading each row

{

for(j=0;j<n;j++) //for reading columns

{

scanf("%d",&b[i][j]); //reading the elements of matrix 2

}

}

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

{

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

{

c[i][j]=a[i][j]+b[i][j]; //adding 1,2 matrices

}

}

printf("Sum of two matrices is \n");

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

{

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

{

printf("%2d ",c[i][j]); //displaying of matrices

}

printf("\n"); //It is for going to next line after each row printing

}

getch();

}

**46. Matrix Multiplication:**

void main()

{

int i,j,k,a[10][10],b[10][10],c[10][10],p,q,m,n;

printf("Enter size of the first matrix\n");

scanf("%d%d",&p,&q); //reading size of the matrix

printf("\nEnter size of the second matrix\n");

scanf("%d%d",&m,&n); //reading size of the matrix

if(q!=m)

{

printf(“\n The matrix multiplication is not possible\n”);

exit(0);

}

printf("\nEnter first matrix elements\n");

for(i=0;i<p;i++) // for going to each row

{

for(j=0;j<q;j++) //for going to each column

{

scanf("%d",&a[i][j]); //reading the elements of matrix 1

}

}

printf("\nEnter second matrix elements\n");

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

{

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

{

scanf("%d",&b[i][j]); //reading the elements of matrix 2

}

}

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

{

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

{

c[i][j]=0;

for(k=0;k<m;k++)

{

c[i][j]=c[i][j]+a[i][k]\*b[k][j]; //multiplying 1,2 matrices

}

}

}

printf("Product of two matrices is \n");

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

{

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

{

printf("%2d",c[i][j]); //displaying of matrices

}

printf("\n"); //It is for going to next line after each row printing

}

getch();

}

**47. //transpose the matrix and checking for symmetry.**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,j,n,a[10][10],b[10][10],flag=0;

printf("Enter size of the matrix ");

scanf("%d",&n); //reading size of the matrix

printf("Enter Matrix elements ");

for(i=0;i<n;i++) //for reading each row

{

for(j=0;j<n;j++) //for reading columns

{

scanf("%d",&a[i][j]); //reading the elements of matrix 1

}

}

printf("\nThe transpose of the given matrix...\n");

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

{

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

{

b[j][i]=a[i][j]; //creating transpose matrix

printf("%3d",b[i][j]);

}

printf("\n");

}

getch();

}

**48. checking for symmetry of a Matrix.**

#include<stdio.h>

#include<conio.h>

void main()

{

int i,j,n,a[10][10],b[10][10],flag=0;

printf("Enter size of the matrix ");

scanf("%d",&n); //reading size of the matrix

printf("Enter Matrix elements ");

for(i=0;i<n;i++) //for reading each row

{

for(j=0;j<n;j++) //for reading columns

{

scanf("%d",&a[i][j]); //reading the elements of matrix 1

}

}

printf("\nThe transpose of the given matrix...\n");

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

{

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

{

b[j][i]=a[i][j]; //creating transpose matrix

printf("%3d",b[i][j]);

}

printf("\n");

}

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

{

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

{

if(a[i][j]!=b[i][j]) //It transpose is not equal to the original matrix

{

flag=1; //set the flag to 0 to print it is not symmentric

break; //break the loop..

}

}

}

if(flag==1)

printf(" Matrices are not Symmetric ");

else

printf(" Matrices are Symmetric ");

getch();

}

**49. Predict the output:**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[20]; /\* declaration \*/

printf("enter any string\n");

scanf("%s",str1); /\* reading the string \*/

printf("U entered\n");

printf("%s",str1); /\*printing the string \*/

getch();

}

**OUTPUT:**

Enter any string

Hi

U entered

Hi

Enter any string

Hello World

U entered

Hello

**Formatted Output function:**

The printf function in C program is used to display information required by the user and also prints the values of the variables.

Its syntax can be given as:

**printf ("control string", arg1,arg2,arg3,...,argn);**

After the control string, the function can have as many arguments as specified in the control string. The control string may contain text to be printed or format specifiers or a mix of both.

**Printing only text:**

printf can be used to print text on the screen. Any number of characters can be used to print. All the text to be printed is placed in the control string which is enclosed in “”. When only text is being printed that means no values of variables, we need only the text in “” and don’t need any other arguments. The function returns the number of output fields successfully printed.

Eg:

***printf ("Welcome to the world of C language");***

output:

***Welcome to the world of C language***

* printf(“Hello”);

printf(“Engineer”);

output:

HelloEngineer

printf will print whatever is in the “”. Since there is no gap after Hello, the output also don’t have gap.

Can be rewritten as

* printf(“Hello”);

printf(“Engineer”);

output:

Hello Engineer

or

* printf(“Hello”);

printf(“ Engineer”);

output:

Hello Engineer

or

* printf(“Hello Engineer”);

output:

Hello Engineer

Escape sequences can also be used as part of text

* printf(“Hello\tEngineer”); \\ \t is horizontal tab

output:

Hello Engineer

* printf(“Hello\nEngineer”); \\ \n is new line

output:

Hello

Engineer

**Note:** Only one control string in one printf.

printf(“H\nE\nL\nL\nO”);

output:

H

E

L

L

O

Printing values of variables through printf:

int x=9;

to print the value of x,

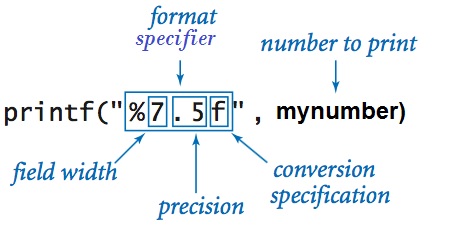
printf(“x”);

output: x

Here the value of x is not printed when given directly in the printf, ‘x’ itself is printed. To print the value of variable, we need to provide format specifiers.

The format specifiers canbe given as below:

**%[flags][width][.precision][specifier]**

****

Each format specifier must begin with a % sign.

flags is an optional argument. The following table shows different types offlags with their descriptions.

**Flags** **Description**

– Left–justify within the given field width

+ Displays the data with its numeric sign (either + or –)

**Width:**width is an optional argument which specifies the minimum number of positions that the outputcharacters will occupy. If the number of output characters is smaller than the specified width,then the output would be right justified with blank spaces to the left. However, if the number ofcharacters is greater than the specified width, then all the characters would be printed.

**precision** is an optional argument which specifies the number of digits to print after the decimalpoint or the number of characters to print from a string.

**Spcifier** indicates the type is used and the interpretation of the value of the corresponding argument.The list of format specifes for different data type variables is following.

|  |  |
| --- | --- |
| **Datatype** | **Format specifier** |
| char | %c |
| int or(signed int ) | %d or %i, %x - (hexadecimal) %o -(octal) |
| unsigned int | %u |
| short or short int or signed short int | %d |
| unsigned short int | %u |
| long int or signed long int | %ld |
| unsigned long int | %lu |
| float | %f or %g or %e(exponential ) |
| double | %lf |
| long double | %Lf |

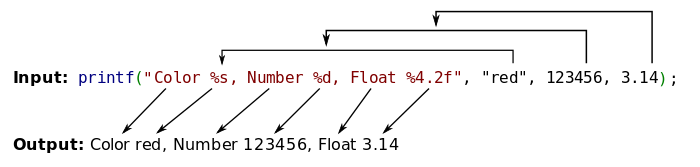
The function when executed prompts the message enclosed in the quotation to be displayed on the screen.

For float x = 8900.768, the following examples show output under different format specifications:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| printf ("%f", x) | 8900.768 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 8 | 9 | 0 | 0 | . | 7 | 6 | 8 | |
| printf("%10f", x); | 8900.768 | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | 8 | 9 | 0 | 0 | . | 7 | 6 | 8 | |
| printf("%9.2f", x); | 8900.77 | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | 8 | 9 | 0 | 0 | . | 7 | 7 | |
| printf("%6f", x); | 8900.768 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 8 | 9 | 0 | 0 | . | 7 | 6 | 8 | |

Printing text along with variables’ values:

The format specifiers must be arranged in the order so that theycorrespond with the arguments in the variable list.



## A summary of printf format specifiers

Here’s a quick summary of the available printf format specifiers:

|  |  |
| --- | --- |
| %c | character |
| %d | decimal (integer) number (base 10) |
| %e | exponential floating-point number |
| %f | floating-point number |
| %i | integer (base 10) |
| %o | octal number (base 8) |
| %s | a string of characters |
| %u | unsigned decimal (integer) number |
| %x | number in hexadecimal (base 16) |
| %% | print a percent sign |
| \% | print a percent sign |

## Controlling integer width with printf

The %3d specifier is used with integers, and means a minimum width of three spaces, which, by default, will be right-justified:

|  |  |
| --- | --- |
| printf("%3d", 0); | 0 |
| printf("%3d", 123456789); | 123456789 |
| printf("%3d", -10); | -10 |
| printf("%3d", -123456789); | -123456789 |

## Left-justifying printf integer output

To left-justify integer output with printf, just add a minus sign (-) after the % symbol, like this:

|  |  |
| --- | --- |
| printf("%-3d", 0); | 0 |
| printf("%-3d", 123456789); | 123456789 |
| printf("%-3d", -10); | -10 |
| printf("%-3d", -123456789); | -123456789 |

## The printf integer symbol-fill option

To symbol-fill your printf integer output, just add a symbol after the %symbol, like this:

|  |  |
| --- | --- |
| printf("%#3d", 0); | ##0 |
| printf("%$3d", 1); | $$1 |
| printf("%#3d", 123456789); | 123456789 |
| printf("%#3d", -10); | -10 |
| printf("%#3d", -123456789); | -123456789 |

## printf integer formatting

As a summary of printf integer formatting, here’s a little collection of integer formatting examples. Several different options are shown, including a minimum width specification, left-justified, zero-filled, and also a plus sign for positive numbers.

|  |  |  |
| --- | --- | --- |
| **Description** | **Code** | **Result** |
| At least five wide | printf("'%5d'", 10); | '   10' |
| At least five-wide, left-justified | printf("'%-5d'", 10); | '10   ' |
| At least five-wide, zero-filled | printf("'%05d'", 10); | '00010' |
| At least five-wide, with a plus sign | printf("'%+5d'", 10); | '  +10' |
| Five-wide, plus sign, left-justified | printf("'%-+5d'", 10); | '+10  ' |

## formatting floating point numbers with printf

|  |  |  |
| --- | --- | --- |
| **Description** | **Code** | **Result** |
| Print one position after the decimal | printf("'%.1f'", 10.3456); | '10.3' |
| Two positions after the decimal | printf("'%.2f'", 10.3456); | '10.35' |
| Eight-wide, two positions after the decimal | printf("'%8.2f'", 10.3456); | '   10.35' |
| Eight-wide, four positions after the decimal | printf("'%8.4f'", 10.3456); | ' 10.3456' |
| Eight-wide, two positions after the decimal, zero-filled | printf("'%08.2f'", 10.3456); | '00010.35' |
| Eight-wide, two positions after the decimal, left-justified | printf("'%-8.2f'", 10.3456); | '10.35   ' |
| Printing a much larger number with that same format | printf("'%-8.2f'", 101234567.3456); | '101234567.35' |

## printf string formatting

|  |  |  |
| --- | --- | --- |
| **Description** | **Code** | **Result** |
| A simple string | printf("'%s'", "Hello"); | 'Hello' |
| A string with a minimum length | printf("'%10s'", "Hello"); | '     Hello' |
| Minimum length, left-justified | printf("'%-10s'", "Hello"); | 'Hello     ' |

## printf special characters

|  |  |
| --- | --- |
| \a | audible alert |
| \b | backspace |
| \f | form feed |
| \n | newline, or linefeed |
| \r | carriage return |
| \t | tab |
| \v | vertical tab |
| \\ | backslash |

**50. Write a program to print length of a string without using string function.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str[30]=”helloworld”;

int len;

len=0;

while(str[len]!=’\0’)

{

len++;

}

printf(“\nThe length of the given string \”%s\” is %d”,str,len);

getch();

}

**OUTPUT:**

The length of given string “hello world” is 11.

**51. Write a program to print length of a string with using string function.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str[30]=”helloworld”;

int len;

len=strlen(str);

printf(“\nThe length of the given string \”%s\” is %d”,str,len);

getch();

}

**OUTPUT:**

The length of given string “hello world” is 11.

**52. Write a program to copy one string to another without using string function.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[20]="hello",str2[20];

int i=0;

for(i=0;str1[i]!=’\0’;i++)

{

str2[i]=str1[i];

}

str2[i]=’\0’;

printf("the string in str2 is %s\n",str2);

getch();

}

**OUTPUT:**

The string in str2 is hello

**53. Write a program to copy one string to another with using string function.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[20]="hello",str2[30],str3[20];

strcpy(str2,str1); /\* copying from str1 to str2 \*/

printf("the string in str2 is %s\n",str2);

strcpy(str3,"hi"); /\* copying a constant “hi” to str3\*/

printf("the string in str3 is %s",str3);

getch();

}

**OUTPUT:**

The string in str2 is hello

The string in str3 is hi

**54. /\*Implement comparision of two strings without using string handling functions\*/**

#include<conio.h>

#include<string.h>

void main()

{

char str1[20],str2[20];

int i;

printf(“Enter the first string\n”);

gets(str1);

printf(“Enter the second string\n”);

gets(str2);

for(i=0;str1[i]!=’\0’ && str2[i]!=’\0’;i++)

{

if(str1[i]<str2[i])

{

printf("\nThe first string is less than second string\n");

}

else if(str1[i]>str2[i])

{

printf("\nThe first string is less than second string\n");

}

}

if(str1[i]==’\0’ && str2[i]==’\0’)

{

printf("\n The strings are equal\n");

}

getch();

}

**OUTPUT:**

The strings are equal

**55. /\*Implement comparision of two strings with using string handling functions\*/**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[20],str2[30];

int i;

printf(“Enter the first string\n”);

gets(str1);

printf(“Enter the second string\n”);

gets(str2);

i=strcmp(str2,str1); /\* comparing str1 and str2 \*/

printf("comparision of str1 and str2 is %d",i);

getch();

}

**OUTPUT:**

comparision of str1 and str2 is 0

**Example:**

char str1[20]="xyz",str2[30]="xxz";

int i;

i=strcmp(str2,str1);

**OUTPUT:**

comparision of str1 and str2 is 1

**Example:**

char str1[20]="xxz",str2[30]="xyz";

int i;

i=strcmp(str2,str1);

**OUTPUT:**

comparision of str1 and str2 is -1.

**56. /\*Implement concatenation of two strings without using string handling functions\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[20],str2[20];

int i,j;

printf(“Enter the first string\n”);

gets(str1);

printf(“Enter the second string\n”);

gets(str2);

for(i=strlen(str1),j=0;str2[j]!=’\0’;i++,j++)

{

str1[i]=str2[j];

}

str1[i]=’\0’;

printf("After concatenation of the two given strings is\n");

puts(str1);

getch();

}

**57. /\*Implement concatenation of two strings with using string handling functions\*/**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[20]="hello",str2[10]="hi";

strcat(str1,str2);

printf(“after concatenating str1 contains %s”,str1);

strcat(str1,”welcome”);

printf(“after concatenating str1 contains %s”,str1);

getch();

}

**Output:**

After concatenating str1 contains **hellohi**

After concatenating str1 contains **hellohiwelcome**

**58. /\*Implement reversing characters in a string without using string handling functions\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[20],str2[20];

int i;

printf(“Enter a string\n”);

gets(str1);

for(i=strlen(str1),j=0;i>=0;i--,j++)

{

str2[j]=str1[i];

}

str2[j]=’\0’;

printf("After reversing the given string\n");

puts(str2);

getch();

}

OUTPUT:

Enter a string

Hello

After reversing the given string

olleH

**59. /\*Implement reversing characters in a string with using string handling functions\*/**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[20];

printf(“Enter a string\n”);

gets(str1);

printf("After reversing the given string\n");

puts(strrev(str1));

getch();

}

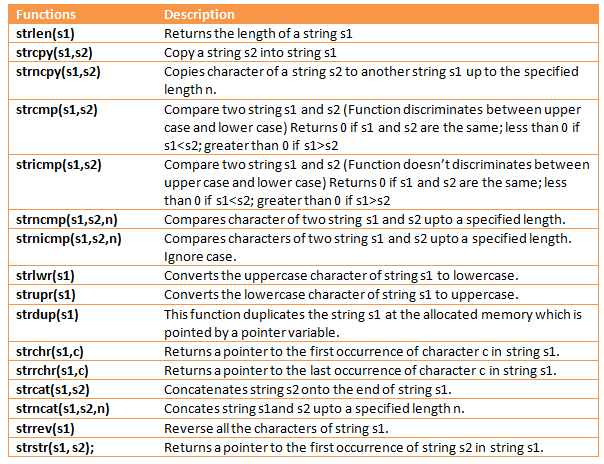
OUTPUT:

Enter a string

Hello

After reversing the given string

olleH



**60. /\* Check Whether a given string is Palindrome or not\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[20],str2[20];

int i,n,flag=0;

printf(“Enter a string\n”);

gets(str1);

i=0;

while(str1[i]!=’\0’)

i++;

n=i;

for(i=n,j=0;i>=0;i--,j++)

{

str2[j]=str1[i];

}

str2[j]=’\0’;

printf("After reversing the given string\n");

puts(str2);

for(i=0;str1[i]!=’\0’;i++)

{

if(str1[i]!=tr2[i])

{

flag=1;

break;

}

}

if(flag==0)

printf(“\n The given string is Palindrome \n”);

else

printf(“\n The given string is NOT Palindrome \n”);

getch();

}

OUTPUT:

Enter a string

MADAM

After reversing the given string

MADAM

The given string is Palindrome

**61. /\*Write a Program to count no. of words, characters in a text\*/**

void main()

{

char text[100];

int i,n\_chars=1,n\_words=1;

printf(“Enter text\n”);

gets(text);

i=0;

n\_chars=0;

n\_words=0;

while(str1[i]!=’\0’)

{

n\_chars++;

if(str1[i]=’ ‘)

n\_words++;

i++;

}

printf("The no. of characters in the given text is %d”, n\_chars);

printf("The no. of words in the given text is %d”, n\_words);

getch();

}

Enter text

Hardwork beats talent when talent doesn’t work hard.

The no. of characters in the given text is 52

The no. of words in the given text is 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data type** | **Description** | **Memory** | **Range** | **Format specifier** | **Example** |
| char | Single character | 1 byte (8 bits) | -128 to +127 | %c | char s=’n’ |
| int  or(signed int ) | Integer quantity | 2 bytes (16 bits) | -32 768 to +32767 (or) -215 to 215-1 | %d or %i, %x(hexadecimal) %o(octal) | int a=39 |
| unsigned int | 2 bytes  (16 bits) | 0 to 216-1 | %u | unsigned int a =70; |
| short or  short int or signed short int | 1 byte  (8 bits) | -128 to + 127 | %d | short int a=100; |
| unsigned short int | 1 byte  (8 bits) | 0 to + 255 | %u | unsigned short int a=200; |
| long int or signed long int | 4bytes  (32 bits) | -231 to 231-1 | %ld | long int a=100; |
| unsigned long int | 4 bytes (32 bits) | 0 to + 232-1 | %lu | unsigned long int a=200; |
| float | Single precision numbers | 4 bytes(32 bits) | 3.4e-38 to 3.4e+38 | %f or %g or %e(exponential ) | float f=29.77 |
| double | Double precision numbers | 8 bytes (64 bits) | 1.7e-308 to 1.7+308 | %lf | double d=3.69721 |
| long double | Very long precision numbers | 10 bytes (80 bits) | 3.4e-4932 to 3.4+4932 | %Lf | long double d=3.678458950 |

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Associativity** |
| ( )  [ ]  .  -> | Parentheses (function call)  Brackets (array subscript)  Member selection via object name  Member selection via pointer | left-to-right |
| ++--  +-  !  ~  (type)  \*  &  Sizeof | Prefix increment/decrement Postfix increment/decrement Unary plus/minus  Logical negation  bitwise complement  Cast (convert value to temporary value of type)  Dereference  Address (of operand)  Determine size in bytes on this implementation | right-to-left |
| \*  /  % | Multiplication  division  modulus | left-to-right |
| +  - | Addition  subtraction | left-to-right |
| <<  >> | Bitwise shift left,  Bitwise shift right | left-to-right |
| <  <=  >  >= | Relational less than/  less than or equal to  Relational greater than/  greater than or equal to | left-to-right |
| ==  != | Relational is equal to  /is not equal to | left-to-right |
| & | Bitwise AND | left-to-right |
| ^ | Bitwise exclusive OR | left-to-right |
| | | Bitwise inclusive OR | left-to-right |
| && | Logical AND | left-to-right |
| | | | Logical OR | left-to-right |
| ? : | Ternary conditional | right-to-left |
| = += -= \*= /= %= &= ^= |= <<= >>= | Assignment Addition/subtraction assignment Multiplication/division assignment Modulus/bitwise AND assignment Bitwise exclusive/inclusive OR assignment Bitwise shift left/right assignment | right-to-left |
| , | Comma (separate expressions) | left-to-right |

1. (a) Write a C program to calculate the area of triangle using the formula area = (s(s-a)(s-b)(s-c))1/2  where s=(a+b+c)/2.

#include<stdio.h>

#include<conio.h>

#include<math.h>

void main()

{

inta,b,c;

floats,area;

printf("Enter three sides of triangle - a,b,c values\n");

scanf("%d%d%d",&a,&b,&c);

s=(a+b+c)/2.0;

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

printf("\nArea of the triangle with sides %d,%d and %d is %f",a,b,c,area);

getch();

}

OUTPUT:

Enter three sides of triangle - a,b,c values

3 4 5

Area of the triangle with sides 3, 4 and 5 is 6.000000

**62. Write a C program to find the largest of three numbers using ternary operator.**

#include<stdio.h>

#include<conio.h>

void main()

{

inta,b,c,max;

printf("Enter three numbers\n");

scanf("%d%d%d",&a,&b,&c);

max=((a>b)&&(a>c))?a:( (b>c)?b: c);

printf("\nMaximum of %d,%d and %d is %d",a,b,c,max);

getch();

}

OUTPUT:

Enter three numbers

3 4 5

Maximum of 3, 4 and 5 is 5.

1. **Write a C program to find the Line Equation (y=y1+m(x-x1))**

int main()

{

float x1,x2,y1,y2;

float x3,y3,m;

scanf(“%f%f%f%f”,&x1,&y1,&x2,&y2);

scanf(“%f”,&x3);

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

y3=y1+m\*(x3-x1);

printf(“%0.2f”,y3);

return 0;

}

1. **Write a C program to find the Line Intersection Point**

**(y1-m1\*x1-y3+m2\*x3)/(m2-m1)**

int main()

{

float x1,x2,y1,y2;

float x3,y3,x4,y4,m1,m2;

float x\_intersection, y\_intersection;

scanf(“%f%f%f%f”,&x1,&y1,&x2,&y2);

scanf(“%f%f%f%f”,&x3,&y3,&x4,&y4);

x\_intersection=(y1-m1\*x1-y3+m2\*x3)/(m2-m1);

y\_intersection=y1+m1\*(x\_intersection-x1);

printf(“%0.2f”, y\_intersection);

return 0;

}

1. **Write a C program to find the Difference between Times**

#include <stdio.h>

struct TIME

{

int seconds;

int minutes;

int hours;

};

int main()

{

struct TIME startTime, stopTime, diff;

printf("Enter start time: \n");

printf("Enter hours, minutes and seconds respectively: ");

scanf("%d %d %d", &startTime.hours, &startTime.minutes, &startTime.seconds);

printf("Enter stop time: \n");

printf("Enter hours, minutes and seconds respectively: ");

scanf("%d %d %d", &stopTime.hours, &stopTime.minutes, &stopTime.seconds);

if(stopTime.seconds > start.seconds)

{

--start.minutes;

start.seconds += 60;

}

diff->seconds = start.seconds - stop.seconds;

if(stop.minutes > start.minutes)

{

--start.hours;

start.minutes += 60;

}

diff->minutes = start.minutes - stop.minutes;

diff->hours = start.hours - stop.hours;

printf("\nTIME DIFFERENCE: %d:%d:%d is ", diff.hours, diff.minutes, diff.seconds);

return 0;

}

void differenceBetweenTimePeriod(struct TIME start, struct TIME stop, struct TIME \*diff)

{

}

**using Command line Arguments**

1. **Factorial of a given number using Command line Arguments**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int i,n,f=1;

n=atoi(aargv[1]);

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

f=f\*i; //f\*=i;

printf(“%d”,f);

}

1. **Sum of individual digits**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

*i*nt n,sum=0;

n=atoi(agrv[1]);

while(n>0)

{

sum=sum+n%10;

n=n/10;

}

printf(“\nThe Sum of individual digits of the given number is %d”, sum);

return 0;

}

1. **Reverse of a given Number**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n,sum=0;

n=atoi(agrv[1]);

while(n>0)

{

rev=rev\*10+n%10;

n=n/10;

}

printf %d”, rev);

return 0;

}

1. **Given Number is Paliindrome or not**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n,sum=0,k;

n=atoi(agrv[1]);

k=n;

while(n>0)

{

rev=rev\*10+n%10;

n=n/10;

}

if(k==rev)

printf(“palindrome”);

else

printf(“not palindrome”);

return 0;

}

1. **Perfect Square**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n,x;

n=atoi(agrv[1]);

x=sqrt(n);

(x\*x==n)?printf(“Perfect Square”):printf(“Not Perfect Square”);

return 0;

}

1. **Prime No or not check**

#include<stdlib.h>

#include<Math.h>

int main(int argc,char \*argv[])

{

int n,i,flag=0;

n=atoi(agrv[1]);

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

{

if(n%i==0)

{

flag=1;

break;

}

}

if (flag==0)

printf("%d is a prime number.",n);

else

printf("%d is not a prime number.",n);

return 0;

}

1. **Prime Nos upto ‘n’**

#include<stdlib.h>

#include<Math.h>

int main(int argc,char \*argv[])

{

int n,i,j,flag=0;

n=atoi(agrv[1]);

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

{

flag=0;

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

{

if(i%j==0)

{

flag=1;

break;

}

}

if (flag==0)

printf("%d ",i);

}

return 0;

}

1. **‘n’ Prime Nos**

#include<stdlib.h>

#include<Math.h>

int main(int argc,char \*argv[])

{

int n,i,j,flag=0,count=1;

n=atoi(agrv[1]);

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

{

flag=0;

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

{

if(i%j==0)

{

flag=1;

break;

}

}

if (flag==0)

{

printf("%d ",i);

count++;

}

}

return 0;

}

1. **Power of n**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n,e,val=1;

e=atoi(agrv[1]);

n=atoi(agrv[2]);

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

{

val=val\*e;

}

printf("%d",val);

return 0;

}

1. **GCD of two numbers**

**HCF (Highest common factor) is also called greatest common divisor (gcd)**

#include<stdio.h>

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n1,n2,r;

n1=atoi(agrv[1]);

n2=atoi(agrv[2]);

if(n2>n1)

{

r=n1;

n1=n2;

n2=r;

}

while(1)

{

r=n1%n2;

if(r==0)

{

break;

}

n1=n2;

n2=r;

}

printf("%d",n2);

return 0;

}

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int n1,n2,i,small;

n1=atoi(agrv[1]);

n2=atoi(agrv[2]);

small=n1<n2?n1:n2;

for(i=small;i>=1;i--)

{

if(n1%i==0 && n2%i==0)

{

printf(“%d”,i);

break;

}

}

return 0;

}

1. **Fibonacci Series upto n numbers**

#include<stdio.h>

#include<conio.h>

int main(int argc, char \*argv[])

{

int i, t,f1,f2,n;

n=atoi(agrv[1]);

f1=1;

f2=1;

printf("%d\t%d\t",f1,f2);

i=3;

while(i<=n)

{

t=f2+f1;

f1=f2;

f2=t;

i++;

printf("%d\t",t);

}

return 0;

}

1. **Nth term in Fibonacci series**

#include<stdio.h>

#include<conio.h>

int main(int argc, char \*argv[])

{

int i, t,f1,f2,n;

n=atoi(agrv[1]);

f1=1;

f2=1;

printf("%d\t%d\t",f1,f2);

i=3;

while(i<=n)

{

t=f2+f1;

f1=f2;

f2=t;

i++;

}

printf("%d",t);

return 0;

}

1. **LCM of two numbers**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int n1,n2,i,small,lcm;

n1=atoi(argv[1]);

n2=atoi(argv[2]);

small=n1<n2?n1:n2;

for(i=small;i>=1;i--)

if(n1%i==0 && n2%i==0)

{

break;

}

lcm=n1\*n2/i;

printf(“%d”,i);

return 0;

}

1. **Printing prime numbers within a range**

#include<stdlib.h>

#include<Math.h>

int main(int argc,char \*argv[])

{

int n1,n2,i,j,flag=0;

n1=atoi(agrv[1]);

n2=atoi(agrv[1]);

for(i=n1;i<=n2;i++)

{

flag=0;

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

{

if(i%j==0)

{

flag=1;

break;

}

}

if (flag==0)

printf("%d ",i);

}

return 0;

}

1. **/\*Write a C program to check if the given number is strong or not\*/**

#include<stdio.h>

#include<conio.h>

void main()

{

*i*nt i,f,n,k,r,sum=0;

printf(“Enter a number”);

scanf(“%d”,&n);

k=n;

while(n>0)

{

r=n%10;

f=1;

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

f\*=i;

sum+=f;

n=n/10;

}

if(k==sum)

{

printf(“\n The given number %d is strong”, k);

}

else

{

printf(“\n The given number %d is not strong”, k);

}

getch();

}

1. **/\*Write a C program to check if the given number is perfect or not\*/**

#include<stdlib.h>

#include<Math.h>

int main(int argc,char \*argv[])

{

int n,i,flag=0;

n=atoi(agrv[1]);

for(i=1;i<=n/2;i++)

{

if(n%i==0)

{

sum+=i;

}

}

if (sum==n)

printf("%d perfect ",n);

}

return 0;

}

1. **Nearest multiple of 10**

**Identify the Corrrection**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int n,i,j,flag;

n=atoi(argv[1]);

if(n%10!=0)

{

++n;

}

printf(“%d ”,n);

return 0;

}

**Better Code**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int n,i,j,flag;

n=atoi(argv[1]);

r=n%10;

if(r<5)

n=n-r;

else

n=n+(10-r);

printf(“%d ”,n);

return 0;

}

1. **GCD of n numbers**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

}

1. **Addition of two numbers without using +**

**How to add two numbers without using the plus operator in c**

#include<stdio.h>

int main( )

{

    int a,b;

    int sum;

    printf("Enter any two integers: ");

    scanf("%d%d",&a,&b);

    //sum = a - (-b);

    sum = a - ~b -1;

    printf("Sum of two integers: %d", sum);

    return 0;

}

Enter any two integers: 5 10

Sum of two integers: 15

1. Program for largest of n numbers without using array.

#include<stdio.h>

int main( )

{

  int n,num,i;

  int big;

  printf("Enter the values of n: ");

  scanf("%d",&n);

  printf("Number %d",1);

  scanf("%d",&big);

  for(i=2;i<=n;i++){

    printf("Number %d: ",i);

    scanf("%d",&num);

    if(big<num)

      big=num;

  }

    printf("Largest number is: %d",big);

  return 0;

}

Sample Output:

Enter the values of n:

Number 1: 12

Number 2: 32

Number 3: 35

Largest number is: 35

### FIND OUT GENERIC ROOT OF A NUMBER By C PROGRAM

**C program for generic root**

Meaning of generic root:

It sum of digits of a number unit we don't get a single digit. For example:

Generic root of 456: 4 + 5 + 6 = 15 since 15 is two digit numbers so 1 + 5 = 6

So, generic root of 456 = 6

#include<stdio.h>

int main( )

{

long int num,sum,r;

scanf("%ld",&num);

while(num>10)

{

sum=0;

while(num)

{

r=num%10;

num=num/10;

sum+=r;

}

if(sum>10)

num=sum;

else

break;

}

printf("\nSum of the digits in single digit is: %ld",sum);

return 0;

}

**C code for calculation of generic root in one line**

#include <stdio.h>

int main(){

int num,x;

printf("Enter any number: ");

scanf("%d",&num);

printf("Generic root: %d",(x=num%9)?x:9);

return 0;

}

Sample output:

Enter any number: 731

Generic root: 2

1. **Program in c to print 1 to 100 without using loop**

#include<stdio.h>

int main( )

{

    int num = 1;

    print(num);

    return 0;

}

int print(num)

{

    if(num<=100){

         printf("%d ",num);

         print(num+1);

    }

}



**Write a c program or code to subtract two numbers without using subtraction operator**

#include<stdio.h>

int main( )

{

    int a,b;

    int sum;

    printf("Enter any two integers: ");

    scanf("%d%d",&a,&b);

    sum = a + ~b + 1;

    printf("Difference of two integers: %d",sum);

    return 0;

}

Sample Output:

Enter any two integers: 5 4

Difference of two integers: 1

**Prime factor of a number in c**

#include<stdio.h>

int main( )

{

  int num,i=1,j,k;

  printf("\nEnter a number:");

  scanf("%d",&num);

  while(i<=num)

{

      k=0;

      if(num%i==0)

{

         j=1;

         while(j<=i)

{

            if(i%j==0)

                 k++;

             j++;

         }

         if(k==2)

             printf("\n%d is a prime factor",i);

      }

      i++;

   }

   return 0;

}

**90.**

**Printing ascii value using c program. C code for ASCII table. C program to display ASCII values**

#include<stdio.h>

int main( )

{

    int i;

    for(i=0;i<=255;i++)

         printf("ASCII value of character %c: %d\n",i,i);

    return 0;

}

**91. Conversion from uppercase to lower case using c program**

#include<stdio.h>  
#include<string.h>

int main(){

  char str[20];

  int i;

  printf("Enter any string->");

  scanf("%s",str);

  printf("The string is->%s",str);

  for(i=0;i<=strlen(str);i++){

      if(str[i]>=65&&str[i]<=90)

       str[i]=str[i]+32;

  }

  printf("\nThe string in lower case is->%s",str);

  return 0;

}

**91. Write a c program to convert the string from lower case to upper case**

#include<stdio.h>

int main(){

  char str[20];

  int i;

  printf("Enter any string->");

  scanf("%s",str);

  printf("The string is->%s",str);

  for(i=0;i<=strlen(str);i++){

            if(str[i]>=97&&str[i]<=122)

            str[i]=str[i]-32;

  }

  printf("\nThe string in lowercase is->%s",str);

  return 0;

}

**92. COUNTING DIFFERENT CHARACTERS IN A STRING USING C PROGRAM**

#include <stdio.h>

int isvowel(char chk);

int main(){

  char text[1000], chk;

  int count;

  count = 0;

  while((text[count] = getchar()) != '\n')

            count++;

  text[count] = '\0';

  count = 0;

  while ((chk = text[count]) != '\0'){

      if (isvowel(chk)){

           if((chk = text[++count]) && isvowel(chk)){

               putchar(text[count -1]);

              putchar(text[count]);

              putchar('\n');

           }

      }

      else

           ++count;

  }

  return 0;

}

int isvowel(char chk){

  if(chk == 'a' || chk == 'e' || chk == 'i' || chk == 'o' || chk == 'u')

      return 1;

  return 0;

}

**93. SORTING OF STRING USING C PROGRAM**

**Program for sorting of string in c language**

#include<stdio.h>

int main(){

  int i,j,n;

  char str[20][20],temp[20];

  puts("Enter the no. of string to be sorted");

  scanf("%d",&n);

  for(i=0;i<=n;i++)

      gets(str[i]);

  for(i=0;i<=n;i++)

      for(j=i+1;j<=n;j++){

           if(strcmp(str[i],str[j])>0){

               strcpy(temp,str[i]);

              strcpy(str[i],str[j]);

              strcpy(str[j],temp);

           }

      }

  printf("The sorted string\n");

  for(i=0;i<=n;i++)

      puts(str[i]);

  return 0;

}

**94. C code which prints initial of any name**

#include<stdio.h>

int main(){

   char str[20];

   int i=0;

   printf("Enter a string: ");

   gets(str);

   printf("%c",\*str);

   while(str[i]!='\0'){

       if(str[i]==' '){

            i++;

            printf("%c",\*(str+i));

       }

       i++;

   }

   return 0;

}

Sample output:

Enter a string: Robert De Niro

RDN

**95. Write a c program to print the string from given character**

#include<string.h>

#include<stdio.h>

int main(){

  char \*p;

  char s[20],s1[1];

  printf("\nEnter a string: ");

  scanf("%[^\n]",s);

  fflush(stdin);

  printf("\nEnter character: ");

  gets(s1);

  p=strpbrk(s,s1);

  printf("\nThe string from the given character is: %s",p);

  return 0;

}

**96. C code to reverse a string by recursion:**

#include<stdio.h>

#define MAX 100

char\* getReverse(char[]);

int main( )

{

    char str[MAX],\*rev;

    printf("Enter  any string: ");

    scanf("%s",str);

    rev = getReverse(str);

    printf("Reversed string is: %s",rev);

    return 0;

}

char\* getReverse(char str[]){

    static int i=0;

    static char rev[MAX];

    if(\*str)

{

         getReverse(str+1);

         rev[i++] = \*str;

    }

    return rev;

}

Sample output:

Enter any string: mona

Reversed string is: anom

**97. Seggregate 0’s& 1’s**

#include<stdio.h>

int main( )

{

    int arr[10],n,i, count0s=0;

scanf(“%d”, &n);

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

{

scanf(“%d”, &a[i]);

if(a[i]==0)

count0s++;

}

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

{

if(i<count0s)

a[i]=0;

else

a[i]=1;

}

    return 0;

}

**98. Find the second last index occurrence of a number in list of elements and if the given number does not occur twice, display -1.**

#include<stdio.h>

int main( )

{

    int arr[10],n,i, key, prev\_occ=-1,curr\_occ=-1;

scanf(“%d”, &n);

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

{

scanf(“%d”, &a[i]);

}

scanf(“%d”, &key);

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

{

if(a[i]==key)

{

prev\_occ=curr\_occ;

curr\_occ=i;

}

}

printf(“%d”, prev\_occ);

    return 0;

}

**99.** [**Write a c program to print Fibonacci series of given range.**](http://cquestionbank.blogspot.com/2008/01/write-c-program-to-find-fibonacci.html)

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int f0,f1,f2,i,n,startno,endno;

startno=atoi(argv[1]);

endno=atoi(argv[2]);

f0=1;

f1=1;

f2=f0+f1;

while(f2<startno)

{

f0=f1;

f1=f2;

f2=f0+f1;

}

while(f2>=startno && f2<=endno)

{

printf(“%d ”,f2);

f0=f1;

f1=f2;

f2=f0+f1;

}

return 0;

}

**100. Write a C program to print hello world without using semicolon**

int main(int argc, char \*argv[])

{

if(printf(“%s”, argv[1])

{

}

}

**101.** [**Write a c program to find largest among three numbers using binary minus operator.**](http://cquestionbank.blogspot.com/2010/06/write-c-program-to-find-largest-among.html)

#include<stdio.h>

int main()

{

    int a,b,c;

    printf("\nEnter 3 numbers: ");

    scanf("%d%d%d",&a,&b,&c);

    if(a-b>0 && a-c>0)

         printf("\nGreatest is a :%d",a);

    else

         if(b-c>0)

             printf("\nGreatest is b :%d",b);

         else

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

    return 0;

}

**102.** [**Write a c program to find out NCR factor of given number.**](http://cquestionbank.blogspot.com/2010/06/write-c-program-to-find-out-ncr-factor.html)

#include<stdio.h>

int main( )

{

  int n,r,ncr;

  printf("Enter any two numbers->");

  scanf("%d %d",&n,&r);

  ncr=fact(n)/(fact(r)\*fact(n-r));

  printf("The NCR factor of %d and %d is %d",n,r,ncr);

  return 0;

}

 int fact(int n)

{

  int i=1;

  while(n!=0)

{

      i=i\*n;

      n--;

  }

  return i;

 }

[**103. How to convert string to int without using library functions in c**](http://cquestionbank.blogspot.com/2011/09/how-to-convert-string-to-int-without.html)

#include<stdio.h>

#include<string.h>

int main(int argc,char \*argv[])

{

int num=0,i,t[10],j;

clrscr();

for(i=strlen(argv[1])-1,j=0;i>=0;i--,j++)

{

t[j]=argv[1][i]-'0';

}

for(i=strlen(argv[1])-1;i>=0;i--)

{

num=num\*10+t[i];

}

printf("%d",num);

getch();

return 0;

}

**104. Consider the below series:**

**1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, …**

**This series is a mixture of 2 series – all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order.**

**Write a program to find the Nth term in this series.**

#include<stdio.h>  
#define MAX 1000  
void fibonacci(int n)  
{  
int i, t1 = 0, t2 = 1, nextTerm;  
for (i = 1; i<=n; i++)  
{  
nextTerm = t1 + t2;  
t1 = t2;  
t2 = nextTerm;  
}  
printf(“%d”, t1);  
}  
   
void prime(int n)  
{  
int i, j, flag, count =0;  
for (i=2; i<=MAX; i++)  
{  
flag = 0;  
for (j=2; j<i; j++)  
{  
if(i%j == 0)  
{  
flag = 1;  
break;  
}  
}  
if (flag == 0)  
count++;  
if(count == n)  
{  
printf(“%d”, i);  
break;  
}  
}  
}  
int main( )  
{  
int n;  
scanf(“%d”, &n);  
if(n%2 == 1)  
fibonacci (n/2 + 1);  
else  
prime(n/2);  
return 0;  
}

char c1;

char \*c2;

char \*c3[];

**105. //Reading array in command line arguments**

#include<stdio.h>

#include<stdlib.h>

int main(int argc, char \*argv[])

{

int n,a[100],i,index\_1,index\_2,ele1,ele2;

n=atoi(agrv[1]);

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

{

a[i]=argv[2+i];

}

ele2=argv[argc-1];

ele1=argv[argc-2];

index\_1=index\_2=-1;

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

{

if(a[i]==ele1)

{

index\_1=i;

}

if(a[i]==ele2)

{

index\_2=i;

}

}

printf(“%d”, index\_1);

printf(“%d”, index\_2);

return 0;

}

**106. Display the factors of a given number**

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int n,i;

n=atoi(agrv[1]);

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

{

if(n%i==0)

{

printf("%d",i);

}

}

return 0;

}

**Pattern printing**

Programs to print square using \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

107.

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

for(j=1; j<=rows; ++j)

{

printf("\* ");

}

printf("\n");

}

return 0;

}

108.

Print M stars in N lines

M=3;

N=4;

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

{

for(j=1; j<=M; ++j)

{

printf("\* ");

}

printf("\n");

}

Programs to print triangles using \*

109.

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

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

{

printf("\* ");

}

printf("\n");

}

return 0;

}

110. Programs to print triangles using numbers

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

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

{

printf("%d ",j);

}

printf("\n");

}

return 0;

}

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

111.

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

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

{

printf("%d ",i);

}

printf("\n");

}

return 0;

}

5

5 4

5 4 3

5 4 3 2

5 4 3 2 1

112.

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

for(i=rows; i>=1; i--)

{

for(j=rows; j>=i; j--)

{

printf("%d ",i);

}

printf("\n");

}

return 0;

}

5

4 4

3 3 3

2 2 2 2

1 1 1 1 1

113. #include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

for(i=rows; i>=1; i--)

{

for(j=rows; j>=i; j--)

{

printf("%d ",i);

}

printf("\n");

}

return 0;

}

114. Programs to print triangles using characters

A

B B

C C C

D D D D

E E E E E

#include <stdio.h>

int main()

{

int i, j;

char input, alphabet = 'A';

printf("Enter the uppercase character you want to print in last row: ");

scanf("%c",&input);

for(i=1; i <= (input-'A'+1); ++i)

{

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

{

printf("%c", alphabet);

}

++alphabet;

printf("\n");

}

return 0;

}

115. Print Floyd's Triangle.

1

2 3

4 5 6

7 8 9 10

Source Code

#include <stdio.h>

int main()

{

int rows, i, j, number= 1;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

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

{

printf("%d ", number);

++number;

}

printf("\n");

}

return 0;

}

116. Programs to print inverted half pyramid using \*

Example 4: Inverted half pyramid using \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

Source Code

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

for(i=rows; i>=1; --i)

{

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

{

printf("\* ");

}

printf("\n");

}

return 0;

}

117. Inverted half pyramid using numbers

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

Source Code

#include <stdio.h>

int main()

{

int i, j, rows;

printf("Enter number of rows: ");

scanf("%d",&rows);

for(i=rows; i>=1; --i)

{

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

{

printf("%d ",j);

}

printf("\n");

}

return 0;

}

118. Programs to display pyramid and inverted pyramid using \* and digits

Program to print full pyramid using \*

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

Source Code

#include <stdio.h>

int main()

{

int i, space, rows, k=0;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

for(space=1; space<=rows-i; space++)

{

printf(" ");

}

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

{

printf("\* ");

}

printf("\n");

}

return 0;

}

119. Program to print full pyramid using \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \*

\* \* \*

\*

Source Code

#include <stdio.h>

int main()

{

int i, space, rows, k=0;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

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

{

printf(" ");

}

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

{

printf("\* ");

}

printf("\n");

}

return 0;

}

120. Program to print Rhombus/Diamond using \*

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \*

\* \* \*

\*

#include <stdio.h>

int main()

{

int i, space, rows, k=0;

printf("Enter number of rows: ");

scanf("%d",&rows);

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

{

for(space=1; space<=rows-i; space++)

{

printf(" ");

}

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

{

printf("\* ");

}

printf("\n");

}

for(i=1; i<=rows-1; i++)

{

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

{

printf(" ");

}

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

{

printf("\* ");

}

printf("\n");

}

return 0;

}

int func(a,4,10)

int \*( \*a)(int \*x)[5]

int a(int x)

int a(int \*x)

int \* a()

int \*a()

int\*( a)(int \*x)