**Practice sessions:**

**Write a simple program that prints the results of all the operators available in C (including pre/ post increment, bitwise and/or/not, etc.). Read required operand values from standard input**.

#include <stdio.h>

int main()

{

int a,b;

printf("enter the values of a and b\n");

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

printf("\na+b=%d",a+b);

printf("\na-b=%d",a-b);

printf("\na\*b=%d",a\*b);

printf("\na/b=%d",a/b);

printf("\na%b=%d",a%b);

printf("\na>b=%d",a>b);

printf("\na>=b=%d",a>=b);

printf("\na<b=%d",a<b);

printf("\na<=b=%d",a<=b);

printf ("\na==b=%d",a==b);

printf("\na&&b=%d",a&&b);

printf("\na||b=%d",a||b);

printf("\na!=b=%d",a!=b);

printf("\na++=%d",a++);

printf("\na--=%d",a--);

printf("\n++a=%d",++a);

printf("\n --a=%d",--a);

printf("\na&b=%d",a&b);

printf("\na|b=%d",a|b);

printf("\n~a=%d",~a);

printf("\na>>2=%d",a>>2);

printf("\n b<<3=%d",b<<3);

return 0;

}

**OUTPUT:**

enter the values of a and b

4 7

a+b=11

a-b=-3

a\*b=28

a/b=0

ab=4

a>b=0

a>=b=0

a<b=1

a<=b=1

a==b=0

a&&b=1

a||b=1

a!=b=1

a++=4

a--=5

++a=5

--a=4

a&b=4

a|b=7

~a=-5

a>>2=1

b<<3=56

**Write a simple program that converts one given data type to another using auto conversion and casting. Take the values form standard input.**

#include <stdio.h>

int main()

{

int i = 17, sum = 17, count = 5;

char c = 'c'; /\* ascii value is 99 \*/

int add;

double mean;

add= i + c;//auto conversion

printf("Value of add : %d\n", add);

mean = (double) sum/count;//casting

printf("Value of mean : %f\n", mean );

return 0;

}

#include <stdio.h>

int main()

{

int i = 17, sum = 17, count = 5;

char c = 'c'; /\* ascii value is 99 \*/

int add;

double mean;

add= i + c;//auto conversion

printf("Value of add : %d\n", add);

mean = (double) sum/count;//casting

printf("Value of mean : %f\n", mean );

return 0;

}

**OUTPUT**

Value of add : 116

Value of mean : 3.400000

**Simple numeric problems:**

**Write a program for find the max and min from the three numbers.**

#include<stdio.h>

int main()

{

int a,b,c;

printf("Enter 3 numbers:\n");

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

if(a>b && a>c)

printf("\nMaximum number is a = %d",a);

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

printf("\nMaximum number is b = %d",b);

else

printf("\nMaximum number is c = %d",c);

if(a<b && a<c)

printf("\nMinimum number is a = %d",a);

else if(b<a && b<c)

printf("\nMinimum number is b = %d",b);

else

printf("\nMinimum number is c = %d",c);

return 0;

}

**OUTPUT:**

Enter 3 numbers:

24 78 9

Maximum number is b = 78

Minimum number is c = 9

**Write the program for the simple, compound interest**

#include<stdio.h>

#include<math.h>

int main()

{

float p,q,r,SI,CI;

int t;

printf("\nEnter the value of Principal p ");

scanf("%f",&p);

printf("\nEnter the value of Rate r ");

scanf("%f",&r);

printf("\nEnter the value of Period in year n = ");

scanf("%d",&t);

SI = ((p\*r\*t)/100);

printf("Simple Interest SI=%f \n",SI);

q = 1+(r/100);

CI=p\*pow(q,t)-p;

printf("Compound Interest CI=%f \n",CI);

return 0;

}

**OUTPUT**

Enter the value of Principal p 5000

Enter the value of Rate r 5

Enter the value of Period in year n = 2

Simple Interest SI=500.000000

Compound Interest CI=512.499499

**Write program that declares Class awarded for a given percentage of marks, where mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input**.

#include <stdio.h>

int main()

{

int per;

printf("Enter your mark ");

scanf("%d",&per);

if(per >= 70)

{

printf(" distinction"); // printing outputs

}

else if ( (per >=60)&&(per<70))

{

printf(" First class");

}

else if ( (per >=40)&&(per<60))

{

printf(" second class");

}

else if ( per< 40)

{

printf(" Failed in the exam");

}

return 0;

}

**OUTPUT:**

Enter your mark 80

Distinction

**Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3**

include<stdio.h>

int main()

{

int num, i = 1,n;

printf("Enter any Number:");

scanf("%d", &num);

printf("\nEnter the number of rows");

scanf("%d",&n);

printf("Multiplication table of %d: ", num);

while (i <= n)

{

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

i++;

}

return 0;

}

**OUTPUT:**

Enter any Number:5

Enter the number of rows6

Multiplication table of 5:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

**Write a program that shows the binary equivalent of a given positive number between 0 to 255**

#include <stdio.h>

void main()

{

int n, i, j, binno=0,dn;

printf("\n\nConvert Decimal to Binary:\n ");

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

printf("Enter a number to convert : ");

scanf("%d",&n);

dn=n;

i=1;

for(j=n;j>0;j=j/2)

{

binno=binno+(n%2)\*i;

i=i\*10;

n=n/2;

}

printf("\nThe Binary of %d is %d.\n\n",dn,binno);

}

**OUTPUT:**

Enter a number to convert : 8

The Binary of 8 is 1000.

**Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,\*, /, % and use Switch Statement**

#include <stdio.h>

int main()

{

int a, b, c;

char ch;

printf("Enter your operator(+, -, /, \*, %)\n");

scanf("%c", &ch);

printf("Enter the values of a and b\n");

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

switch(ch)

{

case '+': c = a + b;

printf("addition of two numbers is %d", c);

break;

case '-': c = a - b;

printf("substraction of two numbers is %d", c);

break;

case '\*': c = a \* b;

printf("multiplication of two numbers is %d", c);

break;

case '/': c = a / b;

printf("remainder of two numbers is %d", c);

break;

case '%': c = a % b;

printf("quotient of two numbers is %d", c);

break;

default: printf("Invalid operator");

break;

}

return 0;

}

**OUTPUT:**

Enter your operator(+, -, /, \*, )

/

Enter the values of a and b

3 5

remainder of two numbers is 0

**Write a program that finds if a given number is a prime number**

#include <stdio.h>

int main()

{

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d",&n);

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

}

OUTPUT:

Enter a positive integer: 9

9 is not a prime number.

**Write a C program to find the sum of individual digits of a positive integer**

#include <stdio.h>

int main()

{

int num,rem,sum=0;

printf("enter a positive integer");

scanf("%d",&num);

while(num>0)

{

rem=num%10;

sum=sum+rem;

num=num/10;

}

printf("sum of digits of %d",sum);

return 0;

}

OUTPUT:

enter a positive integer123

sum of digits of 6

**Write a C program to test given number is palindrome.**

#include <stdio.h>

int main()

{

int num,rem,sum=0,temp;

printf("enter a positive integer");

scanf("%d",&num);

temp=num;

while(num>0)

{

rem=num%10;

sum=sum\*10+rem;

num=num/10;

}

if(temp==sum)

printf("%d is pallindrome",temp);

else

printf("%d is not pallindrome",temp);

return 0;

}

OUTPUT:

enter a positive integer123

123 is not palindrome

**A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence**

#include <stdio.h>

int main()

{

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

printf("enter the limit of fibinocci series");

scanf("%d",&n);

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

{

c=a+b;

a=b;

b=c;

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

}

return 0;

}

OUTPUT:

enter the limit of fibinocci series7

1 2 3 5 8 13 21

**Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.**

#include<stdio.h>

int main()

{

int i,j,n,prime;

printf("\nEnter the range");

scanf("%d",&n);

printf("Prime no.s");

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

{

prime=1;

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

if(i%j == 0)

{

prime=0;

break;

}

if(prime==1)

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

}

return 0;

}

**OUTPUT:**

Enter the range5

Prime no.s

2

3

5

**Write a C program to find the roots of a Quadratic equation.**

#include <stdio.h>

#include <math.h>

void main()

{

float a,b,c,r1,r2,d;

printf("Enter the value for equation: ");

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

if(a==0)

{

printf("\nEnter the value should not be Zero");

}

else{

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

if(d>0)

{

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

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

printf("\nRoots are Real and Unequaal");

printf("%f \n%f \n",r1,r2);

}

else if(d==0){

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

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

printf("\nRoots are Real and Equal\n");

printf("\nRoot=%f \n",r1);

printf("\nRoot=%f \n",r2);

}

else{

printf("\nRoots are Imaginary");

}

}

}

**OUTPUT:**

Enter the value for equation: 2 4 6

Roots are Imaginary

**Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: 1+xx^2+x^+3+………….+x^n.**

#include <stdio.h>

#include <math.h>

void main()

{

int n, x, i, sum = 0;

printf("Enter the limit\n");

scanf("%d", &n);

printf("Enter the value of x\n");

scanf("%d", &x);

if(x < 0 || n < 0)

{

printf("illegal value");

}

else

{

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

sum=sum + pow(x, i);

}

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

}

**OUTPUT:**

Enter the limit

5

Enter the value of x

2

sum=63

**Arrays and Pointers and Functions:**

**Write a C program to find the minimum, maximum and average in an array of integers.**

#include<stdio.h>

int main()

{

int array[100], max=0,min=0,n,i,sum=0,avg;

printf("Enter the number of elements in array\n");

scanf("%d", &n);

printf("enter the elements into an array");

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

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

max = array[0];

min=array[0];

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

{

if (array[i] > max)

{

max = array[i];

}

}

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

{

if (array[i] < min)

{

min = array[i];

}

}

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

{

sum=sum+array[i];

}

avg=sum/n;

printf("Maximum element=%d",max);

printf("\nMinimun element=%d",min);

printf("\n average of elements in an array =%d",avg);

return 0;

}

**OUTPUT:**

Enter the number of elements in array

5

enter the elements into an array1 4 2 7 8

Maximum element=8

Minimum element=1

average of elements in an array =4

**Write a C program that uses functions to perform the following:**

**i. Addition of Two Matrices**

**ii. Multiplication of Two Matrices**

**iii. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be same.**

#include <stdio.h>

int read\_matrix(int a[10][10], int m, int n);

void main()

{

int ch,i,j,m,n,p,q,k,r1,c1,a[10][10],b[10][10],c[10][10];

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t menu");

printf("\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n[1]addition of two matrices");

printf("\n[2]multiplication of two matrces");

printf("[0]exit");

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

printf("\n\t enter your choice:\n");

scanf("%d",&ch);

if(ch<=2&ch>0)

{

printf("valid choice\n");

}

switch(ch)

{

case 1:printf("input rows and columns of A&B matrix:");

scanf("%d%d",&r1,&c1);

printf("enter elements of matrix A(r1\*c1):\n");

for(i=0;i<r1;i++)

for(j=0;j<c1;j++)

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

printf("enter elements of matrix B(r1\*c1):\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

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

}

printf("\n=====matrix addition=====\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

printf("%5d",a[i][j]+b[i][j]);

printf("\n");

}

break;

case 2:

printf("input rows and columns of A matrix:");

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

printf("input rows and colums of B matrix:");

scanf("%d%d",&p,&q);

if(n==p)

{

printf("matrices can be multiplied\n");

printf("resultant matrix %d\*%d\n",m,q);

printf("input A matrix\n");

read\_matrix(a,m,n);

printf("input B matrix\n");

read\_matrix(b,p,q);

printf("\n=====maltiplication=====\n");

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

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

{

c[i][j]=0;

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

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

printf("resutant of two matrices:\n");

}

else

{

printf("matrices cannot be multiplied.");

}

break;

case 0:printf("\n choice terminated");

exit();

break;

default:

printf("\n invalid choice");

}

}

int read\_matrix(int a[10][10], int m, int n)

{

int i,j;

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

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

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

return 0;

}

int write\_matrix(int a[10][10],int m,int n)

{

int i,j;

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

{

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

printf("%5d",a[i][j]);

printf("\n");

}

return 0;

}

**Write C programs that use non- recursive function to find the factorial of a given integer.**

#include <stdio.h>

int factorial(int );//fndeclaration

void main()

{

int i,n,f;

printf("enter the n value");

scanf("%d",&n);

f=factorial(n);

printf("the factorial value is %d",f);

}

int factorial(int n)

{

int i;

int fact=1;

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

fact=fact\*i;

return fact;

}

**OUTPUT:**

enter the n value5

the factorial value is 120

**Write C programs that use recursive function to find the factorial of a given integer**.

#include <stdio.h>

int factorial(int );//fndeclaration

int main()

{

int n,res;

printf("Enter a value to find its Factorial:\n");

scanf("%d",&n);

res=factorial(n);

printf("Factorial of %d= %d\n",n,res);

return 0;

}

int factorial(int n)

{

if((n==0)||(n==1))

return 1;

else

return n\*factorial(n-1);

}

**OUTPUT:**

Enter a value to find its Factorial:

5

Factorial of 5= 120

**Write C programs that use non- recursive function to find the GCD of a two numbers**

#include <stdio.h>

int gcd(int,int);

int main()

{

int a,b;

printf("enter a &b values");

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

if(a>b)

printf("%d\n",gcd(a,b));

else

printf("%d\n",gcd(b,a));

return 0;

}

int gcd(int x,int y)

{

if(x%y==0)

return y;

else

return gcd(y,x%y);

}

**OUTPUT:**

enter a &b values5 10

5

**Write C programs that uses recursive function to find the GCD of a two numbers**

#include<stdio.h>

int main()

{

int n1,n2,gcd;

printf("\nEnter two numbers: ");

scanf("%d %d",&n1,&n2);

gcd=findgcd(n1,n2);

printf("\nGCD of %d and %d is: %d",n1,n2,gcd);

return 0;

}

int findgcd(int x,int y)

{

while(x!=y)

{

if(x>y)

return findgcd(x-y,y);

else

return findgcd(x,y-x);

}

return x;

}

**OUTPUT:**

Enter two numbers: 5 10

GCD of 5 and 10 is: 5

**Write C programs that use non- recursive function to find x power n**

**#include<stdio.h>**

**#include<math.h>**

double power(double num,int result);

void main()

{

int x,y;

double result;

printf("enter any number");

scanf("%d",&x);

printf("enter power of the number");

scanf("%d",&y);

result=power(x,y);

printf("\n %d raised to %d=%.2f\n",x,y,result);

}

double power(double num,int result)

{

return pow(num,result);

}

**OUTPUT:**

enter any number5

enter power of the number2

5 raised to 2=25.00

**Write a program for reading elements using pointer into array and display the values using array.**

#include<stdio.h>

void main()

{

int a[50],\*p,i,n;

p=a;

printf("Enter size of array:");

scanf("%d",&n);

printf("Enter elements of array:");

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

scanf("%d",p+i);

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

printf("%d ",\*(p+i));

}

**OUTPUT:**

Enter size of array:5

Enter elements of array:2 4 6 1 23

2 4 6 1 23

**Write a program for display values reverse order from array using pointer.**

#include <stdio.h>

void main()

{

int size, i, arr[10];

int \*ptr;

ptr = &arr[0];

printf("\nEnter the size of array : ");

scanf("%d", &size);

printf("\nEnter %d integers into array: ", size);

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

{

scanf("%d", ptr);

ptr++;

}

ptr = &arr[size - 1];

printf("\nElements of array in reverse order are :");

for (i = size - 1; i >= 0; i--) {

printf("\nElement%d is %d : ", i, \*ptr);

ptr--;

}

}

**OUTPUT:**

Enter the size of array : 5

Enter 5 integers into array: 1 4 2 7 8

Elements of array in reverse order are :

Element4 is 8 :

Element3 is 7 :

Element2 is 2 :

Element1 is 4 :

Element0 is 1 :

**Write a program through pointer variable to sum of n elements from array**

#include<stdio.h>

void main() {

int A[10];

int i, sum = 0;

int \*ptr;

printf("\nEnter 10 elements : ");

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

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

ptr = A; /\* a=&a[0] \*/

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

sum = sum + \*ptr;

ptr++;

}

printf("The sum of array elements : %d", sum);

}

**OUTPUT:**

Enter 10 elements: 2 5 6 1 8 3 4 9 7 0

The sum of array elements : 45

**Write a C program to display the contents of a file to standard output device**.

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int main()

{

FILE \*fp1, \*fp2;

char ch;

fp1 = fopen("source.txt", "r");

if (fp1 == NULL)

{

puts("File does not exist..");

exit(1);

}

fp2 = fopen("target.txt", "w");

if (fp2 == NULL)

{

puts("File does not exist..");

fclose(fp1);

exit(1);

}

while((ch=fgetc(fp1))!=EOF)

{

ch = toupper(ch);

fputc(ch,fp2);

}

printf("\nFile successfully copied..");

return 0;

}

**OUTPUT:**

File successfully copied..

Write a C program to merge two files into a third file

#include <stdio.h>

#include <stdlib.h>

int main()

{

// Open two files to be merged

FILE \*fp1 = fopen("file1.txt", "r");

FILE \*fp2 = fopen("file2.txt", "r");

// Open file to store the result

FILE \*fp3 = fopen("file3.txt", "w");

char c;

if (fp1 == NULL || fp2 == NULL || fp3 == NULL)

{

puts("Could not open files");

exit(0);

}

// Copy contents of first file to file3.txt

while ((c = fgetc(fp1)) != EOF)

fputc(c, fp3);

// Copy contents of second file to file3.txt

while ((c = fgetc(fp2)) != EOF)

fputc(c, fp3);

printf("Merged file1.txt and file2.txt into file3.txt");

fclose(fp1);

fclose(fp2);

fclose(fp3);

return 0;

}

**OUTPUT:**

Merged file1.txt and file2.txt into file3.txt

**Write a C** **program to convert a Roman numeral ranging from I to L to its decimal equivalent.**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

void main()

{

char rom[30];

int a[30], l, i, k, dec;

printf("Enter the roman number\n");

scanf("%s", &rom);

l =strlen(rom);

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

{

switch (rom[i])

{

case 'I': a[i] = 1;

break;

case 'V': a[i] = 5;

break;

case 'X': a[i] = 10;

break;

case 'L': a[i] = 50;

break;

case 'C': a[i] = 100;

break;

case 'D': dec = dec + 500;

break;

case 'M': a[i] = 1000;

break;

default : printf("Invalid choice");

break;

}

}

k = a[l - 1];

for(i = l - 1; i > 0; i--)

{

if(a[i] > a[i - 1])

{

k = k - a[i - 1];

}

if(a[i] <= a[i - 1])

{

k = k + a[i - 1];

}

}

printf("decimal equivalent is %d", k);

}

**OUTPUT:**

Enter the roman number

VI

Decimal equivalent is 6

**Write a C program that converts a number ranging from 1 to 50 to Roman equivalent**

#include <stdio.h>

int main () {

int num;

printf("Enter your input:");

scanf("%d", &num);

printf("Roman Number: ");

while (num > 0) {

if (num >= 1000) {

/\* M - 1000 \*/

printf("M");

num = num - 1000;

} else if (num >= 500) {

/\*

\* D is 500. CM is 900

\* CM = M - C = 1000 - 100 => 900

\*/

if (num >= 900) {

printf("CM");

num = num - 900;

} else {

printf("D");

num = num - 500;

}

} else if (num >= 100) {

/\*

\* C is 100. CD is 400

\* CD = D - C = 500 - 100 => 400

\*/

if (num >= 400) {

printf("CD");

num = num - 400;

} else {

printf("C");

num = num - 100;

}

} else if (num >= 50) {

/\*

\* L is 50. XC is 90

\* XC = C - X = 100 - 10 => 90

\*/

if (num >= 90) {

printf("XC");

num = num - 90;

} else {

printf("L");

num = num - 50;

}

} else if (num >= 9) {

/\*

\* XL is 40. IX is 9. X is 10

\* XL = L - X = 50 - 10 = 40

\* IX = X - I = 10 - 1 = 9

\*/

if (num >= 40) {

printf("XL");

num = num - 40;

} else if (num == 9) {

printf("IX");

num = num - 9;

} else {

printf("X");

num = num - 10;

}

} else if (num >= 4) {

/\*

\* V is 5 and IV is 4

\* IV = V - I = 5 - 1 => 4

\*/

if (num >= 5) {

printf("V");

num = num - 5;

} else {

printf("IV");

num = num - 4;

}

} else {

printf("I");

num = num - 1;

}

}

printf("\n");

}

**OUTPUT:**

Enter your input:50

Roman Number: L

**Write a C program to determine if the given string is a palindrome or not**

#include <stdio.h>

#include <string.h>

int main()

{

char s[25], rev[25];

int len,i;

printf("Enter the string to check if it is a palindrome\n");

gets(s);

len=strlen(s);

for(i=len-1;i>=0;i--)

{

rev[len-i-1]=s[i];

}

rev[len-i-1]='\0';

printf("Reverse of %s is %s \n",s,rev);

if(strcmp(s,rev)==0)

printf("%s and %s are palindrome\n",s,rev);

else

printf("%s and %s are not palindrome\n",s,rev);

return 0;

}

**OUTPUT:**

Enter the string to check if it is a palindrome

civic

Reverse of civic is civic

civic and civic are palindrome

**Write a C program that displays the position of a character ch in the string S or – 1 if S doesn‘t contain ch.**

#include<stdio.h>

#include<string.h>

#include<conio.h>

void main()

{

char s[30], t[20];

char \*found;

puts("Enter the first string: ");

gets(s);

puts("Enter the string to be searched: ");

gets(t);

found = strstr(s, t);

if(found)

{

printf("Second String is found in the First String at %d position.\n", found - s);

}

else

{

printf("-1");

}

}

**OUTPUT:**

Enter the first string:

hello

Enter the string to be searched:

h

Second String is found in the First String at 0 position.

**Write a C program to count the lines, words and characters in a given text**

#include <stdio.h>

#include <conio.h>

#include <string.h>

void main()

{

char str[100];

int i = 0, l = 0, f = 1;

puts("Enter any string\n");

gets(str);

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

{

l = l + 1;

}

printf("The number of characters in the string are %d\n", l);

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

{

if(str[i] == ' ')

{

f = f + 1;

}

}

printf("The number of words in the string are %d", f);

}

**OUTPUT:**

Enter any string

hello girls how are u

The number of characters in the string are 21

The number of words in the string are 5

**Write a c program to construct a pyramid of numbers as follows**

1

2 2

3 3 3

4 4 4 4

#include <stdio.h>

int main()

{

int i=1,j,n;

printf("Enter a number:\n");

scanf("%d",&n);

while(i<=n)

{

j=1;

while(j<=i)

{

printf("%3d",i);

j++;

}

i++;

printf("\n");

return 0;

}

**OUTPUT:**

Enter a number:

4

1

2 2

3 3 3

4 4 4 4

**Write a c program to construct a pyramid of numbers as follows**

\*

\* \*

\* \* \*

#include <stdio.h>

int main()

{

int i=1,j,n;

printf("Enter a number:\n");

scanf("%d",&n);

while(i<=n)

{

j=1;

while(j<=i)

{

printf("\* ");

j++;

}

i++;

printf("\n");

}

return 0;

}

**OUTPUT:**

Enter a number:

5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**Write a c program to construct a pyramid of numbers as follows**

#include <stdio.h>

int main()

{

int i=1,j,n;

printf("Enter a number:\n");

scanf("%d",&n);

while(i<=n)

{j=1;

while(j<=i)

{

printf("%d ",j);

j++;

}

i++;

printf("\n");

}

return 0;

}

**OUTPUT:**

Enter a number:

3

1

1 2

1 2 3

**write a c program that uses non recursive function to search the key value in the list of integers use linear search method**

#include<stdio.h>

void main( )

{

int i,j,n,a[10],key;

printf("\n enter range for array:");

scanf("%d",&n);

printf("\n enter elements into array:");

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

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

printf("enter the search element:");

scanf("%d",&key);

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

{

if(key==a[i])

{

printf("\n element %d found at %d",key,i+1);

break;

}

}

if(i==n)

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

}

**OUTPUT:**

enter range for array:5

enter elements into array:1 4 6 2 3

enter the search element:6

element 6 found at 3

**write a cprogram that uses non recursive function to search the keyvalue in the list of integers use binary search method**

#include<stdio.h>

int BinarySearch(int arr[20], int max, int element);

void main()

{

int i,element,a[50],position,n,j,temp;

printf("enter the limit of elements in array:\t");

scanf("%d",&n);

printf("enter %d elements in array:\n",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])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("\n the array in sorted order:-\n");

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

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

printf("enter element to search:\t");

scanf("%d",&element);

position=BinarySearch(a,n,element);

if(position==-1)

{

printf("element %d not found\n",element);

}

else

{

printf("element %d found at position %d\n",element, position+1);

}

}

int BinarySearch(int arr[20], int max, int element)

{

int low=0, high=max-1,middle;

while(low<=high)

{

middle=(low+high)/2;

if(element>arr[middle])

low=middle+1;

else if(element<arr[middle]

high=middle-1;

else

return middle;

}

return -1;

}

**OUTPUT:**

enter the limit of elements in array: 5

enter 5 elements in array:

2 5 1 7 9

the array in sorted order:-

1

2

5

7

9

enter element to search: 5

element 5 found at position 3

**write a c program that implements the bubble sort to sort a given list o integers in ascending order**

#include <stdio.h>

void main()

{

int i,j,t,a[5],n;

printf("\n enter the range of array");

scanf("%d",&n);

printf("\n enter elements into array:");

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

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

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

{

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

{

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

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

printf("\n the sorted order is:");

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

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

}

**OUTPUT:**

enter the range of array5

enter elements into array:23 45 12 89 67

the sorted order is: 12 23 45 67 89

**write a c program that sorts the given the array of integers using selection sort in descending order**

#include <stdio.h>

int main()

{

int array[100], n, c, d, position, swap;

printf("Enter number of elements\n");

scanf("%d", &n);

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

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

scanf("%d", &array[c]);

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

{

position = c;

for ( d = c + 1 ; d < n ; d++ )

{

if ( array[position] <array[d] )

position = d;

}

if ( position != c )

{

swap = array[c];

array[c] = array[position];

array[position] = swap;

}

}

printf("Sorted list in descending order:\n");

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

printf("%d\n", array[c]);

return 0;

}

**OUTPUT:**

Enter number of elements

5

Enter 5 integers

23 44 78 90 456

Sorted list in descending order:

456

90

78

44

23

**write a c program that sorts the given the array of integers using insertion sort in ascending order**

#include <stdio.h>

int main()

{

int n, array[1000], c, d, t;

printf("Enter number of elements\n");

scanf("%d", &n);

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

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

scanf("%d", &array[c]);

}

for (c = 1 ; c <= n - 1; c++)

{

d = c;

while ( d > 0 && array[d-1] > array[d])

{

t = array[d];

array[d] = array[d-1];

array[d-1] = t;

d--;

}

}

printf("Sorted list in ascending order:\n");

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

{

printf("%d\n", array[c]);

}

return 0;

}

**OUTPUT:**

Enter number of elements

5

Enter 5 integers

23 33 4 1 6

Sorted list in ascending order:

1

4

6

23

33

**write a c program that sorts the given array of names**

#include<stdio.h>

#include<string.h>

int main(){

int i,j,count;

char str[25][25],temp[25];

puts("How many strings u are going to enter?: ");

scanf("%d",&count);

puts("Enter Strings one by one: ");

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

gets(str[i]);

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

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

{

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

{

strcpy(temp,str[i]);

strcpy(str[i],str[j]);

strcpy(str[j],temp);

}

}

printf("Order of Sorted Strings:");

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

puts(str[i]);

return 0;

}

**OUTPUT:**

How many strings u are going to enter?:

3

Enter Strings one by one:

sheetal

sita

neeta

Order of Sorted Strings:

neeta

sheetal

sita