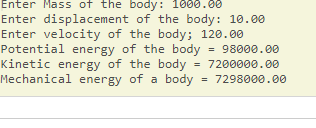
**1) C Program to find Mechanical Energy of a particle using E = mgh+1/2 mv 2 .**

#include <stdio.h>  
int main(void)  
{  
float m,h,v,p,k,e;  
printf("Enter Mass of the body\n");  
scanf("%f",&m );  
printf("Enter displacement of the body\n");  
scanf("%f",&h );  
printf("Enter velocity of the body\n");  
scanf("%f",&v );  
p=m\*9.8\*h; //To calculate Potential energy  
k=0.5\*m\*(v\*v); //To calculate Kinetic energy  
e=p+k;  
printf("Potential energy of the body = %f\n",p );  
printf("Kinetic energy of the body = %f\n",k );  
printf("Mechanical energy of a body = %f\n" , e);  
}  
**OUTPUT:**



**2) C Program to convert Kilometers into Meters and Centimeters.**

#include < stdio.h >

int main()

{

float km, cm, mm, m;

printf("Enter distance in Kelometer\n");

scanf("%f", &km);

m = km \* 1000.0;

cm = km \* 100000.0;

mm = km \* 1000000.0;

printf("Distance in Meter is %f\n", m);

printf("Distance in Centimeter is %f\n", cm);

printf("Distance in Milimeter is %f\n", mm);

getch();

return 0;

}

**OUTPUT:**

Enter Length in Kilometer :

10

Distance in Meter is 10000.000000

Distance in Centimeter is 1000000.000000

Distance in Milimeter is 10000000.000000

**3) C Program To Check the Given Character is Lowercase or Uppercase or Special Character.**

#include<stdio.h>

int main()

{

//for initialize of character

char c;

//to take user input

printf("Enter any character : ");

scanf("%c",&c);

//to find true of upper case value.

if(c>='A' && c<='Z')

printf("character is an upper case");

//to check of lowercase character

else if(c>='a' && c<='z')

printf("character is a lower case");

//to check it is not a character

else if(c>='0'&& c<='9')

printf("it is not a character");

//all condition false than

else

printf("character is a special character");

getch();

return 0;

}

**OUTPUT:**

Enter any character : A

character is an upper case

Enter any character : a

character is a lower case

Enter any character : 9

it is not a character

Enter any character @

character is a special character

**4) Program to balance the given Chemical Equation values x, y, p, q of a simple chemical equation of the type:The task is to find the values of constants b1, b2, b3 such that the equation is balanced on both sides and it must be the reduced form.**

#include<stdio.h>

#include<conio.h>

int gcd(int a,int b)

{

if(b==0)

return a;

return gcd(b,a%b);

}

void balance(int x,int y,int p,int q)

{

int b1,b2,b3,temp;

if(p%x==0 && q%y==0)

{

b1=p/x;

b2=q/y;

b3=1;

}

else

{

p=p\*y;

q=q\*x;

b3=x\*y;

temp=gcd(p,gcd(q,b3));

b1=p/temp;

b2=q/temp;

b3=b3/temp;

}

printf("b1=%d\n b2=%d\n b3=%d",b1,b2,b3);

}

int main()

{

int x=2,y=3,p=4,q=5;

clrscr();

balance(x,y,p,q);

getch();

return 0;

}

**OUTPUT:**

b1=6

b2=5

b3=3

**5) Implement Matrix multiplication and validate the rules of multiplication.**

#include <stdio.h>

int main()

{

int m, n, p, q, c, d, k, sum = 0;

int first[10][10], second[10][10], multiply[10][10];

printf("Enter number of rows and columns of first matrix\n");

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

printf("Enter elements of first matrix\n");

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

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

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

printf("Enter number of rows and columns of second matrix\n");

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

if (n != p)

printf("The multiplication isn't possible.\n");

else

{

printf("Enter elements of second matrix\n");

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

for (d = 0; d < q; d++)

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

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

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

for (k = 0; k < p; k++) {

sum = sum + first[c][k]\*second[k][d];

}

multiply[c][d] = sum;

sum = 0;

}

}

printf("Product of the matrices:\n");

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

for (d = 0; d < q; d++)

printf("%d\t", multiply[c][d]);

printf("\n");

}

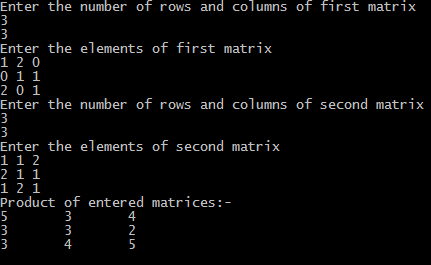
}

getch();

return 0;

}

**OUTPUT:**

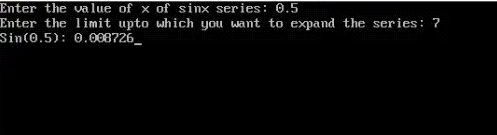


**6) Compute sin(x) using Taylor series approximation.Compare you result with the built-in library function .Print both the result with approximate inferences.**

#include <stdio.h>  
#include <math.h>  
#define PI 3.142  
int main()  
{  
    int n, i;  
    float deg, x, sum = 0, term = 0;  
    printf(“Enter number of terms, say n\n”);  
    scanf(“%d”, &n);  
    printf(“Enter the degree\n”);  
    scanf(“%f”, &deg);  
    x = (deg \* PI) / 180;  
    printf(“In Radians = %f \n”, x);  
    term = x;  
    sum = term;  
    for (i = 3; i <= n; i += 2)  
    {  
        term = (-term \* x \* x) / (i \* (i – 1));  
        sum = sum + term;  
    }  
    printf(“sin(%f)=%f\n”, deg, sum);  
    printf(“Inbuilt function Sin(%f) = %f \n”, deg, sin(x));  
    printf(“User function Sin(%f) = %f”, deg, sum);

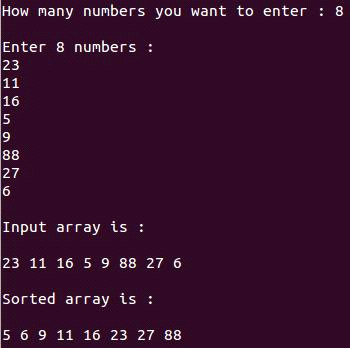
getch();  
}

**OUTPUT:**



**7) Sort the given set of N numbers using Bubblesort.**

#include<stdio.h>  
#define maxsize 20  
int main()  
{  
      int array[maxsize];  
      int i, j, num, t;  
      printf("How many numbers you want to enter : ");  
      scanf("%d", &num);  
      printf("\nEnter %d numbers :\n",num);  
      for (i = 0; i < num; i++)  
      {  
            scanf("%d", &array[i]);  
      }  
      printf("\nInput array is : \n\n");  
      for (i = 0; i < num; i++)  
      {  
            printf("%d ", array[i]);  
      }  
      /\* Bubble sorting begins \*/  
      for (i = 0; i < num; i++)  
      {  
            for (j = 0; j < (num - i - 1); j++)  
            {  
                  if (array[j] > array[j + 1])  
                  {  
                        t = array[j];  
                        array[j] = array[j + 1];  
                        array[j + 1] = t;  
                  }  
            }  
      } // end of sorting  
      printf("\n\nSorted array is : \n\n");  
      for (i = 0; i < num; i++)  
      {  
            printf("%d ", array[i]);  
      }  
      return 0;  
}

**OUTPUT:**  
  


**8)write function to implement string operation such as compare,concatenate,string length.convince the parameters passing techniques.**

**#include <stdio.h>  
#include <string.h>  
int main()  
{  
    int length(char str[100]);  
    int compare(char s1[100], char s2[100]);  
    void concat(char s1[100], char s2[100]);  
      
      
        int option, result;  
        char str[100], s1[100], s2[100];  
        do  
        {  
            printf(“1.String length \n”);  
            printf(“2.string comparision \n”);  
            printf(“3.string concatenation \n”);  
            printf(“4.quit \n”);  
            printf(“enter your choice \n”);  
            scanf(“%d”, &option);  
            switch (option)  
            {  
            case 1:  
                printf(“enter string \n”);  
                scanf(“%s”, &str);  
                result = length(str);  
                printf(“the length of string= %d\n”, result);  
                break;  
            case 2:  
                printf(“enter 1st string\n”);  
                scanf(“%s”, &s1);  
                printf(“enter 2nd string\n”);  
                scanf(“%s”, &s2);  
                result = compare(s1, s2);  
                if (result == 0)  
                {  
                    printf(“strings are equal \n”);  
                }  
                else  
                {  
                    printf(“strings are not equal \n”);  
                    break;  
                }  
  
            case 3:  
                printf(“enter two strings\n”);  
                scanf(“%s%s”, s1, s2);  
                concat(s1, s2);  
                printf(“result=%s \n”, s1);  
                break;  
            }  
        } while (option <= 3);  
        return 0;  
}  
    int length(char str[100])  
    {  
        int i = 0;  
        while (str[i] != ‘\0’)  
            i++;  
        return (i);  
    }  
    int compare(char s1[100], char s2[100])  
    {  
        int i = 0;  
        while (s1[i] != ‘\0’)  
        {  
            if (s1[i] > s2[i])  
                return (1);  
            else if (s1[i] < s2[i])  
                return (-1);  
            i++;  
        }  
        return 0;  
    }  
    void concat(char s1[100], char s2[100])  
    {  
        int i, j;  
        i = 0;  
        while (s1[i] != ‘\0’)  
            i++;  
        for (j = 0; s2[j] != ‘\0’; i++, j++)  
            s1[i] = s2[j];  
        s1[i] = ‘\0’;  
    }**

**OUTPUT:**

Enter 1st string:softech

Enter IInd string:softech

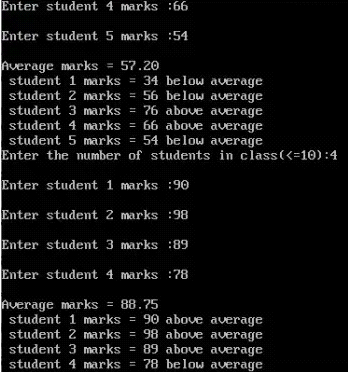
Given two strings:are EQUEL:

Concatenated String=softech softech

Length of Concatenated String=15

**9 a) Implement structures to read,write and compute averages-marks and the student scoring above and below the average marks for a class of N students**

#include<stdio.h>  
#include<conio.h>  
  
struct student{  
int marks;  
}st[10];  
void main()  
{  
int i,n;  
float total=0,avgmarks;  
//clrscr();  
printf("\nEnter the number of students in class(<=10):");  
scanf("%d",&n);  
for(i=0;i<n;i++)  
{  
printf("\nEnter student %d marks :",i+1);  
scanf("%d",&st[i].marks);  
}  
for(i=0;i<n;i++)  
{  
total = total + st[i].marks;  
}  
avgmarks=total/n;  
printf("\nAverage marks = %.2f",avgmarks);  
for(i=0;i<n;i++)  
{  
if(st[i].marks>=avgmarks)  
{  
printf("\n student %d marks = %d above average",i+1,st[i].marks);  
}  
else  
{  
printf("\n student %d marks = %d below average",i+1,st[i].marks);  
}  
}  
getch();  
}**OUTPUT:**



**9 b)Develope a program using pointers to compute the sum,mean and standard deviation of all elements stored in an array of N real numbers**

#include<stdio.h>

#include<math.h>

int main()

{

int n , i;

float x[20],sum,mean;

float variance , deviation;

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

scanf("%d",&n);

printf("enter %d real values \n",n);

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

{

scanf("%f",(x+i));

}

sum=0;

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

{

sum= sum+\*(x+i);

}

printf("sum=%f\n",sum);

mean=sum/n;

sum=0;

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

{

sum=sum+(\*(x+i)-mean)\*(\*(x+i)-mean);

}

variance=sum/n;

deviation=sqrt(variance);

printf("mean(Average)=%f\n",mean);

printf("variance=%f\n",variance);

printf("sum=%f\n",sum);

printf("standard deviation=%f\n",deviation);

}

**OUTPUT:**

Enter the value of n

5

Enter the 5 real values

3

7

23

1

4

Sum = 38.0000

Mean ( Average ) = 7.6000

Variance = 63.039997

Standard deviation = 7.9397