1.SIMPLE CALCULATOR

#include<stdio.h>

#include<conio.h>

void main()

{

char op;

float a,b,result;

printf("simulation of simple calculator\n");

printf("Enter two numbers\n");

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

printf("Enter the operator[+,-,\*,/]\n");

scanf("%s",&op);

switch(op)

{

case'+':result=a+b;

break;

case'-':result=a-b;

break;

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

break;

case'/':if(b==0)

{

printf("division not possible\n");

}

else

{

result=a/b;

}

break;

default:printf("Enter in operation\n");

break;

}

printf("\n Result=%5.2f\n",result);

getch();

}

2.QUADRATIC EQUATION

#include <stdio.h>

#include <conio.h>

#include <math.h>

void main()

{

float a,b,c,d,x1,x2;

printf("enter the three co-efficents a,b,c of quadratic equations");

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

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

if(d==0)

{

printf("\*\*Roots are equal\*\*\n");

x1=x2=-b/(2\*a);

printf("Root1=Root2=%f\n");

}

else if(d>0)

{

printf("\*\*Roots are real and distinct\*\*\n");

x1=-b/2\*a+sqrt(d)/(2\*a);

x2=-b/2\*a+sqrt(d)/(2\*a);

printf("Root1=%f\n\nRoot2=%f\n\n",x1,x2);

}

else

{

printf("\*\*Roots are imaginary\*\*\n");

x1=-b/(2\*a);

x2=sqrt(fabs(d)/(2\*a));

printf("Root1=%f+i%f\n",x1,x2);

}

}

3.ELECTRICITY BILL

#include <stdio.h>

int main()

{

char name[10];

float unit,charge,rental=100;

printf("enter the name and unit consumed\n");

scanf("%s %f",&name,&unit);

if(unit<=200)

charge=unit\*0.80+rental;

else if(unit<=300)

charge=((unit-200)\*0.90)+rental;

else if(unit>300)

charge=(unit-300)\*1+250+rental;

if(charge>=400)

charge=charge+charge\*0.15;

printf("Name:%s\n units:%0.2f\n charge:%0.2f",name,unit,charge);

}

4**.READ THE NO OF ROWS**

#include <stdio.h>

int main()

{

int n,i,j;

printf("enter num of rows");

scanf("%d",&n);

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

{

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

printf(" ");

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

printf("%d",j);

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

printf("%d",j);

printf("\n");

}

}

5.BINARY SEARCH

#include<stdio.h>

int main()

{

int n,a[50],key,i,low,high,mid,found=0;

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

scanf("%d",&n);

printf("enter the elements\n");

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

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

printf("enter key element to search\n");

scanf("%d\n",&key);

low=0;

high=n-1;

while(low<=high)

{

mid=(low+high)/2;

if(a[mid]==key)

{

printf("\n%d key is found at location %d\n",key,mid+1);

found=1;

break;

}

if(key<a[mid] )

high=mid-1;

else if(key>a[mid])

{

low=mid+1;

}

}

if(low>high && found==0)

printf("%d key not found in the list",key);

return 0;

}

**6.MATRIX MULIPLICATION**

#include<stdio.h>

int main()

{ int a[10][10], b[10][10], c[10][10], i, j, k, n, m, sum;

printf(" Enter the Numbers of Row & coloumn : ");

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

printf("\n Enter the Element of First Matrix : \n");

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

{

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

{

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

}

}

printf("enter the second matrix : \n");

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

{

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

{

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

}

} printf("the a matrixs are\n");

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

{

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

{

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

}

printf("\n");

}

printf("the b matrix are\n");

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

{

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

{

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

}

printf("\n");

}

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

{

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

{

sum=0;

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

{

sum=sum+(a[i][k]\*b[k][j]);

}

c[i][j]=sum;

} }

printf("\n\n Multiplicaiton of Two Matrix are : \n");

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

{

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

{

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

} printf(" \n ");

}

return(0);

}

7.TAYLOR’S SERIES

#include<stdio.h>

#include<math.h>

int main()

{

int i,degree;

float x,sum=0,PI,term,num,deno;

printf("Enter the value of degree");

scanf("%d",&degree);

PI=3.14;

x=degree\*(PI/180);

num=x;

deno=1;

i=1;

do{

term=num/deno;

num=num\*x\*x;

deno=deno\*i\*(i+1);

sum=sum+term;

i=i+2;

printf("using built function sin(x) value of %d is %.3f\n",degree,sin(x));

}

while(fabs(term)>=0.00001);

printf("The computed sin(x) value of %d is %.3f\n",degree,sum);

printf("using library function\n");

printf("sin(%d)=(%.3f)\n",degree,sin(x));

}

**8.BUBBLE SORT**

#include <stdio.h>

int main()

{

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

printf("enterthe num of arrays\n");

scanf("%d",&n);

printf("enter the elemntes of arrays\n");

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

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

printf("the elemnts before sorting\n");

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

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

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

{

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

{

if(a[j+1]<a[j])

{

tem=a[j];

a[j]=a[j+1];

a[j+1]=tem;

}

}

}

printf("\nafter sorting is\n");

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

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

}

9.STRING OPERATION

#include <stdio.h>

#include <string.h>

void compare(char str1[],char str2[]);

void concat(char str1[],char str2[]);

void length(char str1[]);

void main()

{

char str1[50],str2[50];

int ch;

printf("Enter the choice\n1:compare\t 2:concatenate\t 3:length\n");

scanf("%d",&ch);

switch(ch)

{

case 1:printf("Enter string 1:\t");

scanf("%s",str1);

printf("Enter string 2:\t");

scanf("%s",str2);

compare(str1,str2);

break;

case 2:printf("Enter string1:\t");

scanf("%s",str1);

printf("Enter string2:\t");

scanf("%s",str2);

concat(str1,str2);

break;

case 3:printf("enter a string:\t");

scanf("%s",str1);

length(str1);

break;

default :printf("Enter invalid choice");

}

}

void compare(char str1[],char str2[])

{

if(strcmp(str1,str2)==0)

printf("given two string are same\n");

else

printf("given two string are not same\n");

}

void concat(char str1[],char str2[])

{

strcat(str1,str2);

printf("resultant string is: %s\n",str1);

}

void length(char str1[])

{

int len;

len=strlen(str1);

printf("The string length is: %d\n",len);

}

10.AVG MARKS OF STUDENT USING STRUCTURE

#include <stdio.h>

#include <string.h>

struct student

{

char name[20];

int m1,m2,m3,total;

}s[20];

void main()

{

int i,n;

float classtotal=0,avg;

printf("Enter the no of student:\t");

scanf("%d",&n);

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

{

printf("Enter the name,marks1,marks2 and marks3\t");

scanf("%s%d%d%d",&s[i].name,&s[i].m1,&s[i].m2,&s[i].m3);

s[i].total=s[i].name+s[i].m1+s[i].m2+s[i].m3;

classtotal=classtotal+s[i].total;

}

printf("student details are:\n");

printf("name\tm1\tm2\tm3\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

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

{

printf("%s\t%d\t%d\t%d\n",s[i].name,s[i].m1,s[i].m2,s[i].m3);

}

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

{

avg=classtotal/n;

printf("\n Average of class = %.2f\n",avg);

printf("\nstudent scoring above avg.2%f\n\n");

}

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

{

if(s[i].total>avg)

printf("%s %d\n",s[i].name,s[i].total);

}

printf("\n student scoring below avg.\n\n");

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

{

if(s[i].total<avg)

printf("%s %d\n",s[i].name,s[i].total);

}

}

11.SUM MEAN AND STANDARD DEVIATION USING POINTERS

#include <stdio.h>

#include <math.h>

int main()

{

float a[10],\*ptr,mean,std,sum=0,sumstd=0;

int n,i;

printf("\n Enter the no of elements\n");

scanf("%d",&n);

printf("\nEnter the arrays elements\n");

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

{

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

}

ptr=a;

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

{

sum=sum+\*ptr;

ptr++;

}

mean=sum/n;

ptr=a;

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

{

sumstd=sumstd+pow((\*ptr-mean),2);

ptr++;

}

std=sqrt(sumstd/n);

printf("\nsum=%.3f\t",sum);

printf("\n mean=%.3f\t",mean);

printf("\n standard devation=%.3f\t",std);

return 0;

}