INTRODUCTION

Scientific Calculator :

The calculator was written by Rolf Howarth in early 1996.

A fully featured scientific calculator with proper operator precedence is implemented, including trig functions and logarithms, factorials, 12 levels of parentheses, logs to base 2 (a handy function for information entropists!), bitwise logical operators, hex, octal, binary and ASCII display.

The calculator is written in JavaScript and you are welcome to view the JavaScript source (visible within the HTML page) for personal educational purposes as long as you recognize that it is copyrighted and not in the public domain. This calculator is now available as part of Hummingbird's Enterprise Information Portal. All enquiries regarding licensing the calculator should be directed to Hummingbird Ltd.

BASIC FUNCTIONS

Addition

The addition (sum function) is used by clicking on the "+" button or using the keyboard. The function results in a+b.

Subtraction

The subtraction (minus function) is used by clicking on the "-" button or using the keyboard.

The function results in a-b

Multiplication

The multiplication (times function) is used by clicking on the "x" button or using the keyboard "\*"key. The function results in a\*b.

Division

The division (divide function) is used by clicking on the "/" button or using the keyboard"/" key. The function results in a/b.

Sign

The sign key (negative key) is used by clicking on the "(-)" button. The function results in -1\*x.

Square

The square function is used by clicking on the "x"2" button or type "^2". The function results in

Square Root

The square root function is used by clicking on the "x" button or type "sqrt()". This

function represents x^.5 where the result squared is equal to x.

Raise to the Power

The raise to the power (y raised to the x function) is used by clicking on the "y"x" button or type "A"

Natural Exponential

The natural exponential (e raised to the x) is used by clicking on the "e"x" button or type "exp()". The result is e (2.71828...) raised to x.

Logarithm

The logarithm (LOG) is used by clicking on the "LOG" button or type "LOG()".

Natural Logarithm

The Natural logarithm (LN) is used by clicking on the "LN" button or type "LN()".

Inverse

Multiplicative inverse (reciprocal function) is used by pressing the "1/x" button or

typing "inv()". This function is the same as x^-1 or dividing 1 by the number.

Exponent

Numbers with exponents of 10 are displayed with an "e", for example 4.5e+100 or 4.5e-100, This function represents 10^x. Numbers are automatically displayed in the format when the number is too large or too small for the display. To enter a number in this format use the exponent key "EEX". To do this enter the mantissa (the non exponent part) then press "EEX" or type"e" and then enter the exponent.

**SYSTEM DESIGN**

Then we began with the design phase of the system. System design is a solution, a "HOW TO" approach to the creation of a new system. It translates system requirements into ways by which they can be made operational. It is a translational from a user oriented document to a document oriented programmers. For that, it provides the understanding and procedural details necessary for the implementation. Here we use Flowchart to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs. It should overcome all the drawbacks of the Old existing system and most important of all meet the user requirements.

**CODING**

#include <stdio.h>

#include <math.h>

#define PI 3.141592654

long long intResult = 0;

double k = 0, result = 0;

int menu()

{

int ch;

printf("\n1. Addition");

printf("\n2. Subtraction");

printf("\n3. Multiplication");

printf("\n4. Division");

printf("\n5. Remainder");

printf("\n6. Factorial");

printf("\n7. Sine");

printf("\n8. Cosine");

printf("\n9. Tangent");

printf("\n10.log(base e)");

printf("\n11.log(base 10)");

printf("\n12.e^x");

printf("\n13.SquareRoot");

printf("\n14.CubeRoot");

printf("\n15.Power");

printf("\n16.Absolute Value");

printf("\n17.Sine Inverse");

printf("\n18.Cosine Inverse");

printf("\n19.Tangent Inverse");

printf("\n20.Ceil Function");

printf("\n21.Floor Function");

printf("\n22.Permutation (nPr)");

printf("\n23.Combination (nCr)");

printf("\n24.Clear");

printf("\n25.Exit");

printf("\nEnter your choice: ");

scanf("%d", &ch);

return ch;

}

void addition()

{

double a, b;

if(k)

{

printf("\nEnter a number: ");

scanf("%lf", &a);

result += a;

printf("\nResult = %lf", result);

}

else

{

printf("\nEnter two numbers: ");

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

result = a + b;

printf("\nResult = %lf", result);

}

}

void subtraction()

{

double a, b;

if(k)

{

printf("\nEnter a number: ");

scanf("%lf", &a);

result -= a;

printf("\nResult = %lf", result);

}

else

{

printf("\nEnter two numbers: ");

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

result = a - b;

printf("\nResult = %lf", result);

}

}

void multiplication()

{

double a, b;

if(k)

{

printf("\nEnter a number: ");

scanf("%lf", &a);

result \*= a;

printf("\nResult = %lf", result);

}

else

{

printf("\nEnter two numbers: ");

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

result = a \* b;

printf("\nResult = %lf",result);

}

}

void division()

{

double a, b;

if(k)

{

printf("\nEnter a number: ");

scanf("%lf", &a);

if(a!=0)

{

result /= a;

printf("\nResult = %lf", result);

}

else

{

printf("Math Error\n");

}

}

else

{

printf("\nEnter two numbers: ");

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

if (b!=0)

{

result = a / b;

printf("\nResult = %lf", result);

}

else

{

printf("Math Error\n");

}

}

}

void mod()

{

long long a, b;

if(k)

{

printf("\nEnter a number: ");

scanf("%lld", &a);

intResult %= a;

printf("\nResult = %d",intResult);

}

else

{

printf("\nEnter two numbers: ");

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

intResult = a % b;

printf("\nResult = %lld", intResult);

}

}

void factorial()

{

long long n, f, i;

printf("\nEnter a number: ");

scanf("%lld", &n);

f = 1;

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

{

f = f \* i;

}

intResult = f;

printf("\nResult = %lld", intResult);

}

long long factorialReturn(long long n)

{

long long f, i;

f = 1;

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

{

f = f \* i;

}

return f;

}

void sine()

{

double a;

printf("Enter angle in radians: ");

scanf("%lf", &a);

result = sin(a);

printf("\nResult = %lf", result);

}

void cosine()

{

double a;

printf("Enter angle in radians: ");

scanf("%lf", &a);

result = cos(a);

printf("\nResult = %lf", result);

}

void tangent()

{

double a;

printf("Enter angle in radians: ");

scanf("%lf", &a);

result = tan(a);

printf("\nResult = %lf", result);

}

void logBasee()

{

double a;

printf("Enter a number: ");

scanf("%lf", &a);

if(a<=0.0)

{

printf("Math Error\n");

}

else

{

result = log(a);

printf("\nResult = %lf", result);

}

}

void logBase10()

{

double a;

printf("Enter a number: ");

scanf("%lf", &a);

if(a<=0.0)

{

printf("Math Error\n");

}

else

{

result = log10(a);

printf("\nResult = %lf", result);

}

}

void eToPowerX()

{

double a;

printf("Enter a number: ");

scanf("%lf", &a);

result = exp(a);

printf("\nResult = %lf", result);

}

void squareRoot()

{

int n;

printf("\nEnter a number: ");

scanf("%d",&n);

if (n<0)

{

printf("Math Error\n");

}

else

{

result = sqrt(n);

printf("\nResult = %lf", result);

}

}

void cubeRoot()

{

int n;

printf("\nEnter a number: ");

scanf("%d",&n);

result = cbrt(n);

printf("\nResult = %lf", result);

}

void power()

{

double base, expo;

printf("Enter a base number: ");

scanf("%lf", &base);

printf("Enter an exponent: ");

scanf("%lf", &expo);

result = pow(base, expo);

printf("%.1lf^%.1lf = %.2lf", base, expo, result);

}

void absolute()

{

int n;

printf("\nEnter a number: ");

scanf("%lld",&n);

intResult = abs(n);

printf("\nResult = %lld", intResult);

}

void sineInverse()

{

double n;

printf("\nEnter a number: ");

scanf("%lf",&n);

if(n>1 || n<-1)

{

printf("Not in Range");

}

else

{

result = asin(n);

printf("Inverse of sin(%.2f) = %.2lf in radians\n", n, result);

result = asin(n)\*180/PI;

printf("Inverse of sin(%.2f) = %.2lf in degrees\n", n, result);

}

}

void cosineInverse()

{

double n;

printf("\nEnter a number: ");

scanf("%lf",&n);

if(n>1 || n<-1)

{

printf("\nNot in Range");

}

else

{

result = acos(n);

printf("\nInverse of cos(%.2f) = %.2lf in radians\n", n, result);

result = acos(n)\*180/PI;

printf("\nInverse of cos(%.2f) = %.2lf in degrees\n", n, result);

}

}

void tangentInverse()

{

double n;

printf("\nEnter a number: ");

scanf("%lf",&n);

result = atan(n);

printf("\nInverse of tan(%.2f) = %.2f in radians", n, result);

result = (result \* 180) / PI;

printf("\nInverse of tan(%.2f) = %.2f in degrees", n, result);

}

void ceilF()

{

double n;

printf("\nEnter a number: ");

scanf("%lf",&n);

result = ceil(n);

printf("Ceiling integer of %.2f = %f", n, result);

}

void floorF()

{

double n;

printf("\nEnter a number: ");

scanf("%lf",&n);

result = floor(n);

printf("Floor integer of %.2f = %f", n, result);

}

void npr()

{

long long n, r;

printf("\nEnter two numbers: ");

scanf("%lld%lld", &n, &r);

if(n>=r && n>0 && r>=0)

{

intResult = factorialReturn(n) / factorialReturn(n - r);

printf("\nResult = %lld", intResult);

}

else

{

printf("Math Error\n");

}

}

void ncr()

{

long long n, r;

printf("\nEnter two numbers: ");

scanf("%lld %lld", &n, &r);

if(n>=r && n>0 && r>=0)

{

intResult = factorialReturn(n) / (factorialReturn(r) \* factorialReturn(n - r));

printf("\nResult = %lld", intResult);

}

else

{

printf("Math Error\n");

}

}

void clear()

{

printf("\nOld Data Cleared");

intResult = 0;

result = 0;

k = 0;

}

void main()

{

int l = 0;

while(1)

{

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

printf("\n Old Integer Result = %d", intResult);

switch(menu())

{

case 1: addition();

k = 1;

break;

case 2: subtraction();

k = 1;

break;

case 3: multiplication();

k = 1;

break;

case 4: division();

k = 1;

break;

case 5: mod();

k = 1;

break;

case 6: factorial();

k = 1;

break;

case 7: sine();

k = 1;

break;

case 8: cosine();

k = 1;

break;

case 9: tangent();

k = 1;

break;

case 10:logBasee();

k = 1;

break;

case 11:logBase10();

k = 1;

break;

case 12:eToPowerX();

k = 1;

break;

case 13:squareRoot();

k = 1;

break;

case 14:cubeRoot();

k = 1;

break;

case 15:power();

k = 1;

break;

case 16:absolute();

k = 1;

break;

case 17:sineInverse();

k = 1;

break;

case 18:cosineInverse();

k = 1;

break;

case 19:tangentInverse();

k = 1;

break;

case 20:ceilF();

k = 1;

break;

case 21:floorF();

k = 1;

break;

case 22:npr();

k = 1;

break;

case 23:ncr();

k = 1;

break;

case 24:clear();

break;

case 25:l = 1;

break;

default:

printf("\nInvalid Choice !");

}

printf("\nPress any button to continue......");

getch();

system("cls");

if(l == 1)

break;

}

}