**SCIENTIFIC CALCULATOR**

21CSS101J – PROGRAMMING FOR PROBLEM SOLVING

Mini Project Report

*Submitted by*

LAVANYA PAKHALE

[Reg. No.: RA2211028010132] B.Tech. CSE – Cloud Computing

&

DEEPSHIKHA KUMARI

[Reg. No.: RA2211028010134] B.Tech. CSE – Cloud Computing



**SCHOOL OF COMPUTING**

**COLLEGE OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

S.R.M. NAGAR, KATTANKULATHUR – 603 203 KANCHEEPURAM DISTRICT

**December 2022**

INDEX

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Problem statement and Objective | 3 |
| 2 | Methodology : Algorithm and Flowchart | 4-5 |
| 3 | Program | 6-9 |
| 4 | Output | 10 |
| 5 | Conclusion | 11 |

**PROBLEM STATEMENT**

A Scientific calculator with proper operator precedence is implemented including trigonometric functions, addition, subtraction, multiplication, division, pie problems, logarithm, modulus, power, exponent, history etc. The problem statement was to design a calculator that performs various operations with full accuracy and speed, which will help the administrator to customize all the changes and also allow the user to see the history of the operations performed.

The program uses C language for its complete functionality.

**OBJECTIVE**

The project aims to develop the internal programmed computational code in the form of a computer program that a scientific calculator could use to compute functions such as square root, sine, cos, tan ,exponential functions etc and display the history of the operations.

**ALGORITHM:**

**Step 1**: Start

**Step 2**: Initialize pi = 3.14 and max = 100

**Step 3**: Declare history

**Step 4**: Declare functions for calculations namely, add(x,y), mins(x,y), multiply(x,y), divide(x,y), mod(x,y), sine(x), cosine(x), tangent(x), sineh(x), cosineh(x), tangenth(x), logten(x), squareroot(x), exponent(x), power(x)

**Step 4.1**: The functions return calculation result in float value

**Step 5**: Declare function historypush(answer)

**Step 5.1**: Declare variable final[MAX]

**Step 5.2**: Call function gvct() to convert answer into String and store in final

**Step 5.3**: Call function strcat() to concat final to history

**Step 6**: Declare function main()

**Step 6.1**: Declare variable n,num1, num2, answer

**Step 6.2**:Input n

**Step 6.3**: if n>0 and n<7 is true,

**Step 6.3.1**: Input num1, num2

**Step 6.3.2**: switch case to call functions add(x,y), mins(x,y), multiply(x,y), divide(x,y), mod(x,y),power(x, y) and they return answer

**Step 6.3.3**: call function pushhistory(answer)

**Step 6.3.4**: print answer

**Step 6.3.5**: call main()

**Step 6.4**: if n>= 7 and n<= 15 is true,

**Step 6.4.1**: Input num1

**Step 6.4.2**: switch case to call functions sine(x), cosine(x), tangent(x), sineh(x), cosineh(x), tangenth(x), logten(x), squareroot(x), exponent(x) and they return answer

**Step 6.4.3:** call function pushhistory(answer)

**Step 6.4.4**: print answer

**Step 6.4.5**: call main()

**Step 6.5**: if n=16 is true,

**Step 6.5.1**: print history

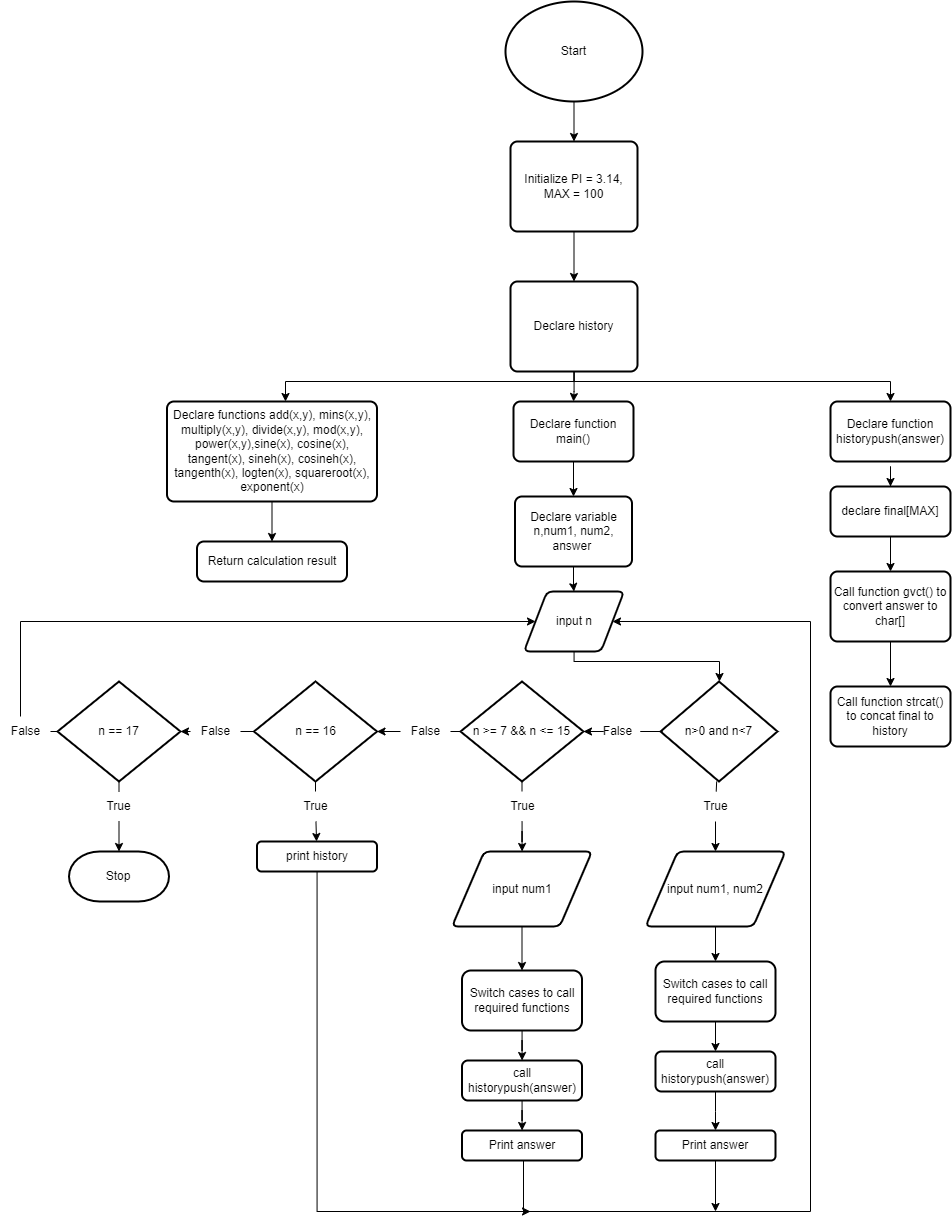
**Step 6.5.2**: call main()

**Step 6.6**: if n=17 is true,

**Step 6.6.1**: Stop

**Step 6.7:** else call function main()

**FLOW CHART:**



**PROGRAM:**

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdlib.h>

#define PI 3.14159265

#define MAX 100

char history;

float add(float a, float b)

{

return a + b;

}

float minus(float a, float b)

{

return a - b;

}

float multiply(float a, float b)

{

return a \* b;

}

float divide(float a, float b)

{

return a / b;

}

float mod(float a, float b)

{

return remainderf(a, b);

}

float sine(float x)

{

return sinf(x \* PI / 180);

}

float cosine(float x)

{

return cosf(x \* PI / 180);

}

float tangent(float x)

{

return tanf(x);

}

float sineh(float x)

{

return sinhf(x);

}

float cosineh(float x)

{

return coshf(x);

}

float tangenth(float x)

{

return tanhf(x);

}

float logten(float x)

{

return log10f(x);

}

float squareroot(float x)

{

return sqrtf(x);

}

float exponent(float x)

{

return expf(x);

}

float power(float x, float y)

{

return powf(x, y);

}

void historypush(float answer)

{

char final[MAX];

gcvt(answer, MAX, final);

strcat(&history, final);

strcat(&history, "\n");

}

int main(void)

{

int n;

float num1, num2, answer;

printf("\n What operation do you want to perform? \n");

printf("1.addition 2.subtraction 3.multiplication 4.division 5.modulus 6.power \n");

printf("7.sin 8.cos 9.tan 10.sinh 11.cosh 12.tanh 13.1og10 14.square root. 15.exponent \n");

printf("16.history 17.exit \n");

printf("Input:");

scanf("%d", &n);

if (n > 0 && n < 7)

{

printf("Enter first number :");

scanf("%f", &num1);

printf("Enter second number :");

scanf("%f", &num2);

switch (n)

{

case 1:

answer = add(num1, num2);

break;

case 2:

answer = minus(num1, num2);

break;

case 3:

answer = multiply(num1, num2);

break;

case 4:

answer = divide(num1, num2);

break;

case 5:

answer = mod(num1, num2);

break;

case 6:

answer = power(num1, num2);

break;

default:

return 0;

}

historypush(answer);

printf("The output of the operation is: %f\n", answer);

}

else if (n >= 7 && n <= 15) {

printf("Enter a number :");

scanf("%f", &num1);

switch (n)

{

case 7:

answer = sine(num1);

break;

case 8:

answer = cosine(num1);

break;

case 9:

answer = tangent(num1);

break;

case 10:

answer = sineh(num1);

break;

case 11:

answer = cosineh(num1);

break;

case 12:

answer = tangenth(num1);

break;

case 13:

answer = logten(num1);

break;

case 14:

answer = squareroot(num1);

break;

case 15:

answer = exponent(num1);

break;

default:

return 0;

}

historypush(answer);

printf("The output of the operation is: %f \n", answer);

}

else if (n == 16)

{

printf("Output history: \n");

printf("%s", &history);

}

else if (n == 17)

{

printf("Thanks for using, exiting now! \n");

return 0;

}

else

{

printf("Enter a valid input! \n");

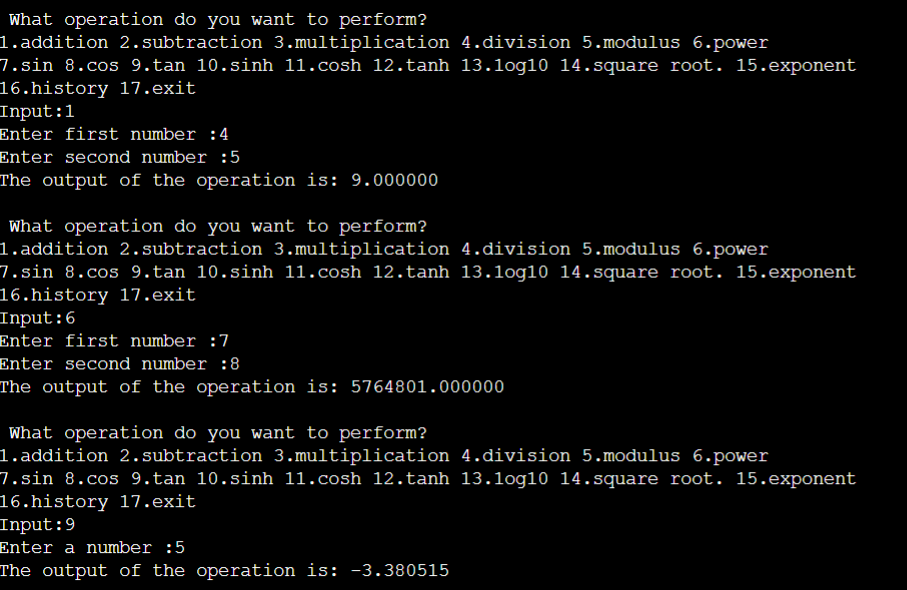
return main();

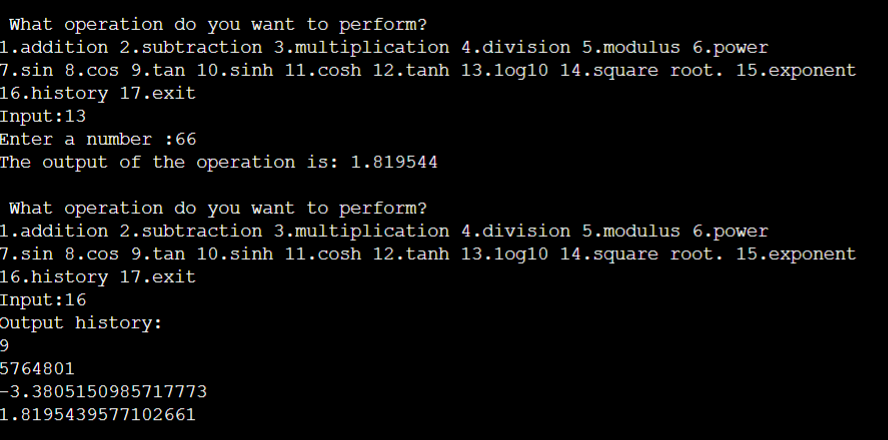
}

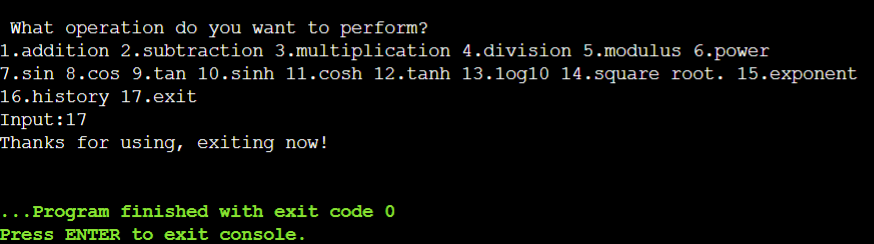
return main();

}

**OUTPUT:**







**CONCLUSION:**

The code for the scientific calculator and runs successfully. The scientific calculator is designed to help you calculate science, engineering and mathematics problems including trigonometric functions, pi, log, addition, subtraction, division etc. This program helps to understand functions.