-	Cavendish University Uganda.	
	3 id \ P = 024/37	
	Cout October 2024	
	course Unit: Principles of C Programming	
	Mame: Lakwonyero Danjel	
	Student's 1D: 203603	
	i zako esa zasto	
	Question	
	Klinte a pregram wing e programming for a calculator that	
	performs basic operations (addition, subtrection, multiplication,	
	during (2) 10 10 10 10 10 10 10 10 10 10 10 10 10	
	Touk:	
a	Implement a simple calculator classifications details	
	That is the calculator in C	
	57755 - 57755 - 57755	
	# include <stdio.n> (dia) lha = line</stdio.n>	
Ą	# include <assert-h></assert-h>	
	# include < math. h>	
	the state of the s	
	1/ calculator functions	-
	double add (double a, double b) {	
	return at b; () short = where	
	3	
	double subtract (double a, double b) {	
	: Caretum a-b;	
	3 - Company of the second of t	
	double multiply (double a double b) {	
	return axb;	
	3	
	double divide (double a, double b) {	
	if(b==0)	
	Printf ("Error: Division by zero is invalid. In");	
	return 0; }	

	else & when the prime and interests	
	return a/b; 3	
	3 The same should be the same	
	11 The main part of c calculator.	
	int main () {	
_	deuble a, b, result; 30000 1/18 200 1/18 200 1/18	7
	Char operator;	
	1/ calculator functionality	
	teath retained as no set pointer pass a point Nagarage to 17414	
	printf ("Enter an expression (for example 415): ");	
	scanf (" " stf " &c " stf", &a, & genator, &b);	_4
	: Y25T	
	Switch (Operator) &	
	case +':	
	result = add (a, b); < Nothto > atmosph =	
	break; ** ** ** ** ** ** ** ** ** ** ** ** **	
	case 1=1: - KN. Bhill > stution #	
	result = subtract (a, b);	
	break; 2000 ust setal solve	-
	i case 11: - Et d 3 plant on a script) the words	
	result = divide (a, b); of the mention	
	break;	
	default: { Cx 3/25-to 18 grating) to militar almos	
	Printf ("Invalid operator, Please try again! \n");	
	return 1; }	
	Printf ("Result: 1/24 m", result);	
	return 0; }	
	5	
	2 (d sty sot in a work) at with also of	
	\$ (0==d) 3i	
	11" at showing 2 and rapid to 1 mone" 174 m. 9	
	£ 10 couter	

b) kinto unit tests for the methods in the calculation. Here I will add a function that will compare flooding - point number with a small epsilon. int is close (double x, double y, double epsilon) & return fabs (x-y) < epilon; } Man below & Where I county out the unit tests for each method. Il assert is used in that case void testing-cole() } 11 test addition. assert (is_close (add (3,2), 5, 0.0001)); assert (is_close (add (1.5, 2.5), 4, 0.0001)); 11 testing subtraction. assert (is_close (subtract(3,2), 1,0,000)); assert (is_close (subtract (2,5), -3,0.000)); // testing multiplication assert (is-close (multiply (3,2); 6,0.0001)); assert (is_close (multiply (1.5,2), 3,00001)); 11 testing division method. accept (is close (divide (10,2), 5, 0.0001)); assert (is close (divide (5,2), 2.5, 0.0001)); 11 testing division by zero assert (divide (5,0) ==0); // error expected here Printf ("All tests passed succentrally! In"); When the tests are all successful, the program will display "All tests period successfully, however if it does not successfully pass the tests, then it will gistford an Eurol.