*/\*  
 \*main.c  
 \*  
 \* Created: 8/8/2018 11:58:54 PM  
 \* Authors : Apollo Aggrey  
 \* Victor Okumu  
 \* Charles Mutua  
 \*Description: solve for a specified unknown in one of the 3 Newtons equations of motion.  
 \*  
 \*  
 \*/*#pragma clang diagnostic push  
#pragma clang diagnostic ignored **"-Wdeprecated-declarations"**#define **True** 1  
#define **False** 0  
#define **bool int**#define **or** ||  
  
  
#include **<stdio.h>** *// NOLINT***char** prompt\_unknown(**int**);  
  
**int** choose\_eqn();  
  
**void** calculate(**int**, **char**, **double** \*);  
  
**void** output\_result(**int** eqn, **char** unknown, **double** \*variables);  
  
**void** get\_inputs(**int** eqn, **char** unknown\_var, **double** \*variables);  
  
**double** Newton\_sqrt(**double** x);*//find square root using Newtons method to 6d.p***double** abs(**double** x); *// fond absolute vaalue of a number***int** main() {  
 **bool** proceed;  
 proceed = **True**;  
 **while** (proceed) {  
 **int** eqn; *//to store equation to be solved* **char** var; *//to store unknown variable name* eqn = choose\_eqn();  
 var = prompt\_unknown(eqn);  
 **double** variables[7];  
 get\_inputs(eqn, var, variables);  
 calculate(eqn, var, variables);  
 output\_result(eqn, var, variables);  
 printf(**"Do you want to continue? \n "  
 "[y] : yes\n "  
 "[n] : No\n"  
 ">>>"**);  
 **char** choice[8];  
 scanf(**"%s"**, choice); *// NOLINT* **if** (choice[0] == **'y' or** choice[0] == **'Y'**)proceed = **True**;  
 **else** proceed = **False**;  
 }  
 **return** 0;  
}  
  
*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/***double** abs(**double** x) {  
 **if** (x < 0) {  
 **return** x \* -1;  
 }  
 **return** x;  
}  
  
**double** Newton\_sqrt(**double** x) *//Newtons mtd*{  
 **const double** eps = 1e-6;  
 **double** t = x;  
 **while** (abs(t \* t - x) >= eps)  
 t = (t + x / t) / 2;  
 **return** t;  
}  
  
**void** calculate(**int** eqn, **char** unknown, **double** \*variables)*//[a,s,t,u,v]*{ *//variables mappings[0,1,2,3,4]* **if** (eqn == 1) {  
 **if** (unknown == **'v'**) {  
 variables[4] = variables[0] \* variables[2] + variables[3];  
 } **else if** (unknown == **'a'**) {  
 variables[0] = (variables[4] - variables[3]) / variables[2];  
 } **else if** (unknown == **'t'**) {  
 variables[2] = (variables[4] - variables[3]) / variables[0];  
 } **else if** (unknown == **'u'**) {  
 variables[3] = variables[4] - variables[0] \* variables[2];  
 }  
  
 } **else if** (eqn == 2) {  
 **if** (unknown == **'s'**) {  
 variables[1] = 0.5 \* (variables[0]) \* variables[2] \* variables[2] + variables[3] \* variables[2];  
 } **else if** (unknown == **'a'**) {  
 variables[0] = (variables[1] - (variables[3] \* variables[2])) / (0.5 \* variables[2] \* variables[2]);  
 } **else if** (unknown == **'t'**) {  
 **if** ((variables[3] \* variables[3] - (2 \* variables[0] \* variables[1])) > 0) {  
 variables[2] =  
 (variables[3] \* -1 +  
 Newton\_sqrt(variables[3] \* variables[3] - (2 \* variables[0] \* variables[1]))) /  
 (2 \* variables[0]); *//first sln to quad eqn -b(+)Newton\_sqrt( ...* variables[6] =  
 (variables[3] \* -1 -  
 Newton\_sqrt(variables[3] \* variables[3] - (2 \* variables[0] \* variables[1]))) /  
 (2 \* variables[0]); *//second sln to quad eqn -b(-)Newton\_sqrt( ...* variables[7] = 0; *//flag to show t is REAL* } **else** {  
 variables[7] = 1; *//flag to show t is undefined (i.e imaginary)* variables[2] = 0;  
 variables[6] = 0;  
 }  
 } **else if** (unknown == **'u'**) {  
 variables[3] = (variables[1] - (0.5 \* variables[0] \* variables[2] \* variables[2])) / variables[2];  
 }  
 } **else if** (eqn == 3) {  
 **if** (unknown == **'v'**) {  
 variables[4] = Newton\_sqrt(2 \* variables[0] \* variables[1] + variables[3] \* variables[3]);  
 } **else if** (unknown == **'a'**) {  
 variables[0] = (variables[4] \* variables[4] - variables[3] \* variables[3]) / (2 \* variables[1]);  
 } **else if** (unknown == **'s'**) {  
 variables[1] = (variables[3] \* variables[3] - variables[4] \* variables[4]) / (2 \* variables[1]);  
 } **else if** (unknown == **'u'**) {  
 variables[3] = Newton\_sqrt(variables[4] \* variables[4] - 2 \* variables[0] \* variables[1]);  
 }  
 }  
}  
  
  
**void** output\_result(**int** eqn, **char** unknown, **double** \*variables) {  
 **double** out;  
 printf(**"The value of "**);  
 printf(**"%c"**, unknown);  
 printf(**" is "**);  
 **if** (unknown == **'a'**) {  
 out = variables[0];  
 printf(**"%lf"**, out);  
 } **else if** (unknown == **'s'**) {  
 out = variables[1];  
 printf(**"%lf"**, out);  
 } **else if** ((unknown == **'t'**) & (variables[7] == 0))*// check if its not imaginary* {  
 out = variables[2];  
 printf(**"%lf"**, out);  
 } **else if** ((unknown == **'t'**) & (variables[7] == 1))*// check if its imaginary* {  
 printf(**" UNDEFINED "**); *//TIME CANT BE IMAGINARY* } **else if** (unknown == **'u'**) {  
 out = variables[3];  
 printf(**"%lf"**, out);  
 } **else if** (unknown == **'v'**) {  
 out = variables[4];  
 printf(**"%lf"**, out);  
 }  
  
 **if** (eqn == 2) {  
 **if** ((unknown == **'t'**) & (variables[7] == 0))*// check if its not imaginary* {  
 printf(**" and/or "**);  
 printf(**"%lf"**, variables[6]);  
 }  
  
 }  
 printf(**"\n"**);  
}  
  
  
**int** choose\_eqn() {  
 **int** eqn = 1;  
 printf(**"choose an equation to solve?\n"  
 " [1]: first equation( v = at + u)\n"  
 " [2]: second equation( s = 0.5(at^2) + ut)\n"  
 " [3]: third equation(v^2 = 2as + u^2)\n"  
 ">>> "**);  
 scanf(**"%d"**, &eqn); *// NOLINT* **return** eqn;  
}  
  
**char** prompt\_unknown(**int** eqn) *//what variable should be solved for? takes the chosen eqn as args*{  
 **char** unknown[8];*// to store unknown* printf(**"choose a variable to solve for\n"**);  
 **if** (eqn == 1) {  
 printf(**"[v,a,t,u] \n>>"**);  
 } **else if** (eqn == 2) {  
 printf(**"[s,a,t,u] \n>>"**);  
 } **else if** (eqn == 3) {  
 printf(**"[v,a,s,u] \n>>"**);  
 }  
 scanf(**"%s"**, unknown); *// NOLINT* **return** unknown[0];  
}  
  
**void** get\_inputs(**int** eqn, **char** unknown\_var, **double** \*variables) {  
 *//[a,s,t,u,v]  
 //variables mappings[0,1,2,3,4]* **if** (eqn == 1) {  
 **if** (unknown\_var == **'v'**) {  
 printf(**"Enter values of [a,t,u] \n>>"**);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[2]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'a'**) {  
 printf(**"Enter values of [v,t,u] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[2]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'t'**) {  
 printf(**"Enter values of [v,a,u] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'u'**) {  
 printf(**"Enter values of [v,a,t] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[2]);  
 }  
  
 } **else if** (eqn == 2) {  
 **if** (unknown\_var == **'s'**) {  
 printf(**"Enter values of [a,t,u] \n>>"**);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[2]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'a'**) {  
 printf(**"Enter values of [s,t,u] \n>>"**);  
 scanf(**"%lf"**, &variables[1]);  
 scanf(**"%lf"**, &variables[2]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'t'**) {  
 printf(**"Enter values of [s,a,u] \n>>"**);  
 scanf(**"%lf"**, &variables[1]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'u'**) {  
 printf(**"Enter values of [s,a,t] \n>>"**);  
 scanf(**"%lf"**, &variables[1]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[2]);  
 }  
 } **else if** (eqn == 3) {  
 **if** (unknown\_var == **'v'**) {  
 printf(**"Enter values of [a,s,u] \n>>"**);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[1]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'a'**) {  
 printf(**"Enter values of [v,s,u] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[1]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'s'**) {  
 printf(**"Enter values of [v,a,u] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[3]);  
 } **else if** (unknown\_var == **'u'**) {  
 printf(**"Enter values of [v,a,s] \n>>"**);  
 scanf(**"%lf"**, &variables[4]);  
 scanf(**"%lf"**, &variables[0]);  
 scanf(**"%lf"**, &variables[1]);  
 }  
 }  
}