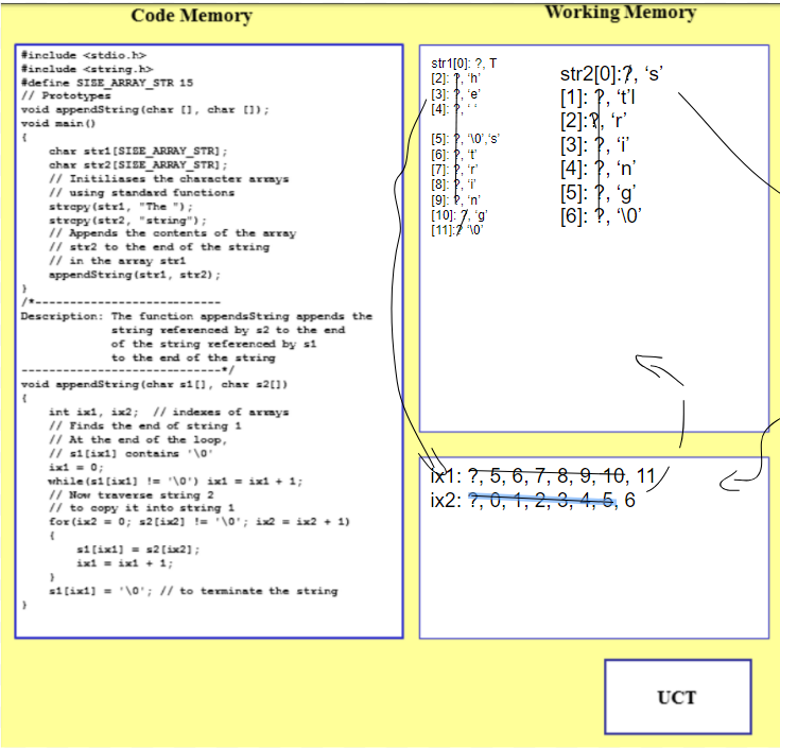
GNG1106

Assignment 5

Amit Nayak

300066780

Question 1:

Question 2:

/\*

Amit Nayak 300066780 GNG1106 Assignment 4

October 21, 2018

Average Velocity of Water Program:

This program takes in channel characteristics from the user to calculate and return the average velocity of water.

It returns 25 incremental velocities from the max depth given.

It prints the depths and velocites in a table for the user to see.

\*/

#include <stdio.h>

#include <math.h>

// Define symbolic constant

#define D\_IX 0

#define V\_IX 1

#define LINES 25

typedef struct

{

char name[15];

double n;

double slope;

double width;

double maxDepth;

} CHANNEL;

// Prototypes

double getPositiveValue();

void displayTable(CHANNEL \*, double [2][LINES]);

void getInput(CHANNEL \*);

void fillArray(CHANNEL \*, double [2][LINES]);

double computeVelocity(CHANNEL \*, double );

/\*--------------------------------------------------------------------

Function: main

Description: Gets from the user values for the equation.

----------------------------------------------------------------------\*/

int main()

{

CHANNEL chan; //structure variable for CHANNEL.

CHANNEL \*chanPtr;

chanPtr = &chan;

double points[2][LINES];

getInput(chanPtr);

fillArray(chanPtr, points);

displayTable(chanPtr, points);

return 0;

}

/\*------------------------------------------------------------------------

Function: getPositiveValue

Returns: A value strictly positive (>0)

Description: Reads a real value from the user, checks that it is strictly

positive, and returns the value.

------------------------------------------------------------------------\*/

double getPositiveValue()

{

double value;

do

{

scanf("%lf",&value);

if(value <= 0.0)

printf("The value must be greater than zero: \n");

}

while(value <= 0.0);

return(value);

}

/\*-----------------------------------------------------------------------

Function: displayTable

Description: calls the computeVelocity function to calculate all the

velocities at each increment of depth. Then it prints the values in a table

with the the channel characterstics above.

------------------------------------------------------------------------\*/

void displayTable(CHANNEL \*chanPtr, double points[2][LINES])

{

//This section of code prints out the table.

printf("\nChannel Data for: %s", (\*chanPtr).name);

printf("Coefficient of Roughness: %0.4f\n", (\*chanPtr).n);

printf("Slope: %0.5f\n", (\*chanPtr).slope);

printf("Width: %10.2f\n", (\*chanPtr).width);

printf("Maximum Depth: %0.2f\n", (\*chanPtr).maxDepth);

printf("%10s %10s\n","Depth", "Velocity");

printf("------------------------\n");

for(int i = 0; i < LINES; i++)

printf("%10.4f %10.6f\n", points[D\_IX][i], points[V\_IX][i]); //print depth and velocity using array values.

}

/\*------------------------------------------------------------------------

Function: computeVelocity

Parameters: CHANNEL chan, depth.

Description: Passes in the CHANNEL chan members plus the depth.

Calculates using a series of step the average velocity. returns the value.

------------------------------------------------------------------------\*/

double computeVelocity(CHANNEL \*chanPtr, double depth)

{

double u = 0; //Value used for calculation.

double temp = 0; //Temporary value used in calculation.

//series of steps for final equation given in the assignment instructions.

u = sqrt((\*chanPtr).slope);

u = u / (\*chanPtr).n;

temp = (((\*chanPtr).width/ depth)/((\*chanPtr).width + (2 \* depth)));

temp = pow(temp, (2.0/3.0));

u = u \* temp;

return u;

}

//Series of prompts ask the user to enter the input then stores the input via pointer.

void getInput(CHANNEL \*chanPtr)

{

//This section of code gets all of the user data to fill all the members of CHANNEL struct.

printf("Please give the name of the channel: ");

fgets((\*chanPtr).name, 15, stdin);

printf("Please enter the coefficient of roughness: ");

(\*chanPtr).n = getPositiveValue();

printf("Please enter the slope: ");

(\*chanPtr).slope = getPositiveValue();

printf("Please enter the channel width: ");

(\*chanPtr).width = getPositiveValue();

printf("Please enter the maximum depth of the channel: ");

(\*chanPtr).maxDepth = getPositiveValue();

}

//Calculates and stores the values in a 2d array via pointer.

void fillArray(CHANNEL \*chanPtr, double points[2][LINES])

{

//Initialization of variables.

double depth = 0; //Depth variable for each calculation of each increment.

float inc; //Increment variable

inc = (\*chanPtr).maxDepth/(LINES); //Increment based on max depth and number of lines.

depth = inc; //set depth to first increment

//runs the calculation for each increment of depth.

//calls the computeVelocity function for each depth increment.

for(int i = 0; i < LINES; i++)

{

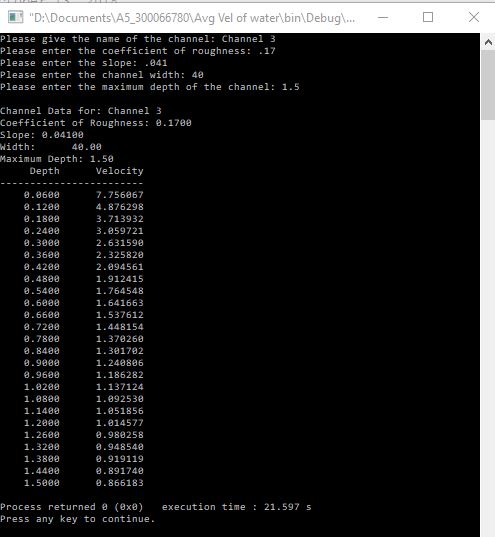
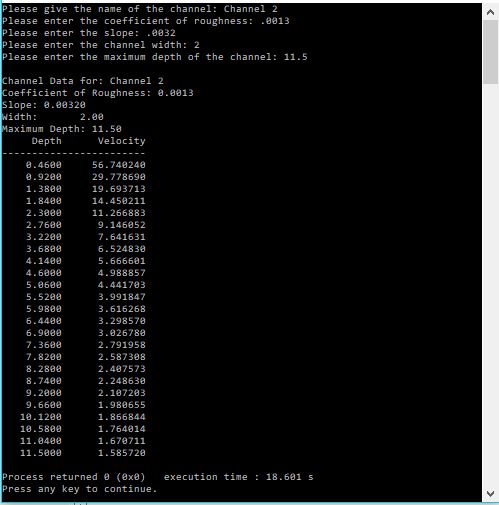
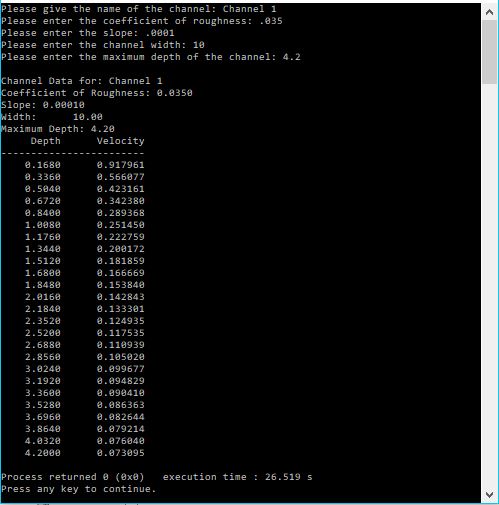
points[D\_IX][i] = depth; //set array points to depth value

points[V\_IX][i] = computeVelocity(chanPtr, depth); //set array points to velocity value

depth = depth + inc; //increment depth

}

}



Question 3

