All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 1: Hello world program.

Notes:

This program uses the stdio.h library header.

It introduced the main function (int main void)

It also demonstrated the printf function (printf("");)

It demonstrated statement terminators (;)

I learned that a main function needs a return statement i.e. an int as its int main void

Program 2: Hello world program with newline character

Notes:

Continuing from program 1 this program demonstrated the newline character

This character (\n) creates a newline for content after it in the printf statement

Program 3: Add 2 numbers

Notes:

In this program I created 3 integer variables.

I added these variables together and assigned the sum to a new variable

I then printed the result.

This program does not ask for user input it just prints result

Program 4: Size of different variable types

Notes:

This program introduced arrays of different variable types (float, char, int, double).

It also introduced the sizeof function which returns the byte size of a variable

I created different variable types and measured there size and then printed the result

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 5: Ask User for favourite colour

Notes:

This program asks user for favourite colour by using the scanf function.

This function gets inserts user input into a variable of any type in this case a char array (%s).

I then displayed the users response on screen by printing the variable to make sure in had been

Input correctly.

I also used %15s to stop buffer overflow as the array was size 16.

Program 6: Print first letter of Favourite Colour

Notes:

Continuing from program 5 after asking user for favourite colour the task was to display on the

first letter.

I had 3 different solutions

The first solution was to access the specific element in the array and print this on screen

The second was similar to function five except that when I was printing it to the screen I put %c instead of %s and this only prints the first char

The third solution was to limit the character overflow to only 1 character i.e. %1s then there would only be 1 character in the array and it would print the first letter of the users favourite colour

Program 7: Buffer overflow example

Notes:

This program we were tasked with overflowing an array. I set the array to a very low number

I then held a on the keyboard for 2 minutes. This caused the program to crash as expected,

thus showing the importance of stopping buffer overflow.

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 8: User inputs 2 numbers add and display result

Notes:

This program combined scanf function with adding the variables from earlier programs

The user input 2 numbers. There values were added and assigned to a third result variable

The result was then printed on the screen

Program 9: User inputs 2 numbers into integer array add/subtract and display result

Notes:

This program is a modification of program 8 using an integer array instead of separate int variables

To input I used scanf and specified the array element to input for each number i.e. a [1] a [2] etc.

I then added the variables using the correct element specifies and assigned the result to a sum array. I then subtracted the variables and assigned to the next element in sum array. I then printed the result on screen

Program 10: User inputs 3 variables and program prints the biggest

Notes:

This program asked the user to input 3 different numbers

The program then used an if statement and 2 else if statements to decide which number to output.

The program used Boolean logic operators including the && operator i.e. AND

Using the && operator means it would only print if both conditions of statement were true

Program 11: Demonstration of getchar function

Notes:

This program introduces a while loop and its conditional statement.

It also demonstrates how char strings can have a newline character at the end

This program removes this from the array and if not it prints a warning

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 12: Calculator Program

Notes:

This program allows user to input 2 numbers and an operator (char)

It then uses if loops based on what character is chosen to deliver the right calculation (*, \,-, +)

Program 13: Calculator Program allowing more than 1 calculation

Notes:

This program continues from program 12 and allows more than calculation

The condition while (operation! = q) means that if uses q as operator program will end otherwise it will keep going allowing calculation after calculation on an infinite loop

Program 14: Difference between Signed and Unsigned Bytes (integers)

Notes:

Up until now we have used %d to print numbers it allows signed and unsigned numbers

However if we want to make sure that only signed numbers are used %u allows that to happen

This program subtracts 2 from 1 (1 – 2) and displays result using %u (unsigned byte) as it now cannot be a negative number it becomes 4292067295 which is the maximum size of an integer in C

Program 15: Demonstrate that Chars have a byte size of 1

Notes:

In this program we create a loop that loops 10000 times.

I also created a char variable c and have it equal to 0, then add 1 to the char every time

Then the value is printed in signed byte format so the program outputs the numbers -128 to 127 inclusive again and again until it has looped 10000 times. This shows that a char has a range of 0 - 255 and a byte size of 1

Program 16: User inputs a word a count number of characters

Notes:

In this program the character loops until the end of the word is reached i.e. a '\0' character as strings end with a space.

For every letter the loop increments a counter

The value of the counter is then displayed to show the amount of characters in the word

Program 17: Calculate length of word using a for loop

Notes:

This program has the same function as program 16 except that it uses a for loop

The for loop initiates a variables value sets the condition and increments variable for each loop inside in the same brackets. These 3 are separated by (;)

I++ is short hand for I = I +1 and it increments I by 1 after each loop

Program 18: User Enters Secret Word to end Program

Notes:

This program uses a constant to set secret word set above main function.

It then uses the strcmp function to compare if the users word is the same as the one that is entered. If strcmp is 1 it is false if it returns a 0 it is true

The program displays a message if right or wrong using if statements

Program 19: Enter the secret word to end the program user has 3 guesses

Notes:

This program asks the user for a secret word. It then compares the word using strcmp to the word entered if true it displays a message and the program ends if not it gives the user the option to have another guess. It uses a for loop to allow for only 3 guesses.

A break statement is used to exit the for loop if guess is correct this in turn allows the program to end

I have also set the program to show how many guesses out of 3 the user has taken by setting the counter variable to 1 which changes as it increments.

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 20: Demonstration of fgets function

Notes:

This program demonstrates the fgets function which allows the user to input into a character array. fgets allows the program to copy everything entered including spaces so it can store a sentence

Fegts consists of 3 parts

The variable name

The variable size

The stream or method of input in this case stdin which stands for standard input.

All 3 are separated by a comma.

Program 21: Reverse the letters in the word for each word of a sentence and print

Notes:

This program asks users to input a sentence using fgets

It then reverses each word in the sentence but keeps them in the same position and prints.

So the sentence: I love release notes (BECOMES) I evol esaeler seton

This program introduces the <string.h> header which can call the library functions strlen etc. strlen measures the string length which needs to be measured before reversing the word

It uses 2 for loops which cycle through the letters in the word and when the reach a space it changes the first character of a word to the last character in a word by using a temporary variable and incrementing /decrementing counters to go through elements it then sets the new variable to have a space and moves on to the next word

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 22: Change vowels of user inputted sentence to $ sign

Notes:

This program allows user to input a sentence into a character array

It then uses a for loop to cycle through each letter.

Inside the for loop is an if statement with conditions separated by || operator. This is the OR operator in C if which allows the function to be through if 1 condition or another is true.

This program uses the or operator to check if each element in array is a vowel (lowercase and capitals) and replaces it with a $ symbol

Program 23: Secret word program again with wait for user input function

Notes:

This program has 3 functions include with main.

A clear screen function clears screen by print many newlines

I have then moved earlier functionality of program 19 to a function that read user input. This function has the character the character array word passed to it. After user input function is done the wait for user input function is activated this uses a dummy fgets with local variable dummy to stop the program until the user hits the return key to continue

Program 24: Reverse letters in each word using a function

Notes:

This program has the same functionality as program one except reversing takes place inside a function. For practice I also used pointers in this function.

The pointer to the character array of user inputted sentence is passed along with the reverse of the sentence. An integer start is also declared in the function header

Malloc is used to assign memory to each variable in this case malloc (100)

Program 25: Count how many letters appears in a user inputted sentence

Notes:

This program creates 2 character arrays to store capital and lower case letters. It also creates an int array to act as a counter for each letter a – z with both capital and lowercase types incrementing the same counter i.e. A or a mans that count[0] = 1

The program uses a for loop to cycle through each element in the user inputted for each letter in the alphabet arrays i.e. 1 loop to see if sentence chars are a's and another for b's etc.

A Final for loop is used to display each letter and how many times it appeared.

Program 26: Demonstrate Second Main Function

Notes:

This program introduces the concept of the second main function which allows user inputted content on the cmd line to be used in the program. The second main consists of the argument count and the argument value. The argument count is the number of things on the cmd (separated by spaces) and the argument value is the content of each element. It is important to remember that the first element will be the executable

These can then be displayed and manipulated within the program

Program 27: Demonstrate Pointers

Notes:

This program demonstrates the concept of pointers. These are address of variables

You can assign as much memory as you want to pointers and manipulate the values they point to

It is important to remember to free pointers after use as otherwise it can cause a memory leak which leads to runtime errors in your program

To use pointers the library header <stdlib.h> is used in C programs

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 28: Difference between Arrays and Pointers

Notes:

This program shows the difference between arrays and pointers

By comparing its output to that of program 28.

In this program we output the value of a string aswel as its address.

The Difference is that arrays are conceptual data representations which have elements

Pointers are references to variables but are not actually variables themselves

Program 29: Create a calculator program using the second main function and no loops

Notes:

This program allows the user to input a calculation e.g. 10 x 12 on the cmd line after executable

It then uses sscanf to change elements 2 and 4 of argv[] to integers as argv is of char*

The program then uses if statements based on the condition of the operator sign being the same as the sum that it carries out. The result is displayed and the program ends

Program 30: Demonstrate how to access elements inside pointers and change them

Notes:

This program uses pointers it assigns memory to them and uses fgets in allow input.

Pointers are then accessed the same as arrays and there values can be changed

After this I freed the pointers and then printed out their contents. This showed that the values changed not fully but random characters did appear.

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 31: Guess the secret word program (word in text file)

Notes:

This program introduces file input/output. File pointers are created by using FILE *fp etc.

Files are then opened using the fopen function and can be opened in different modes. In the case of this program in read mode “r”. The file is checked if it exits by seeing if fopen returns a NULL pointer. If the file does not exist it uses exit function to leave the program

The program then acts the same as earlier versions of the secret word programme except that a loop is made to read content line by line. Content is read by fgets from the file by changing stream to file pointers name. It reads until it returns and end of file character i.e. outside the signed byte range0-255 i.e. -1.

The program compares the inputted word by the user to this word from the file and gives the user 3 guesses.

An alternate solution although slower and inefficient is reading each character into a variable to store the word and the comparing this to the user inputted word

Program 32: Program that can copy a text file

Notes:

This program opens two files one to read from and the other to write to.

If the program to read to does not exist the program ends.

The program writes to a file set to write mode even it does not exist already

It then uses a while loop to write each character to the second file until the end of the file you are writing from is finished.

Program 33: Read secret word from file with 20 secret words after each is guessed correctly move on to next word (3 guesses per word)

Notes:

This program reads a word from the file with 20 secret words using fgets which reads the program line by line. The 20 secret words each have their own line so only that is entered

User input is then compared to the word and if correct it moves on to next word. The user is informed how many guesses they have completed out of 3 if they get it wrong 3 times the program ends

Program 34: Allow user to compare 2 files on the cmd line

Notes:

Using the second main function the user enters the executable and 2 files separated by a space on the command line the files are then read character by character if even 1 character differs the program will end with a message warning they are not the same if not a message saying they are the same will appear

Program 35: Program that displays the contents 20 lines at a time

Notes:

This program opens a file containing text if text file does not exist it exits the program

The program then prints each character of the file on the screen using putchar () function. If it reaches a newline character it increments a counter by 1

After counter reaches 20 the counter is reset to 0 and user is given the option to quit or print next 20 lines. This cycle continues until the end of file is reached

Program 36: Program that checks if equal amounts of brackets () {} []. File to test can be entered on the cmd line

Notes:

This program creates a variable for each type of bracket [] {} () it then cycles through each character in the file once and when it encounters a bracket it increments its type counter by 1.

At the end of the program there are 4 if statements 1 for success and 3 telling the user if those type of brackets are not equal and how many of them there are in the program.

This program is limited in that it does not tell the user if the brackets have a closing counterpart in order i.e.)(will appear as if there are equal amounts of brackets but as you can see it leaves 2 open brackets in the file without informing the user

All programs contain correct files needed to run in file folders **Peter Hanily (12252788)**

Program 37: Allow user to input a record into a struct, write this to file and then reload back into sturct.

Notes:

This program creates a struct with 3 char arrays for name, telephone and student number (must be char if you want to input numbers that start with 0)

The user is then presented with a menu with 4 options using a switch statement.

0 to quit the program, 1 to edit struct contents, 2 to save struct to file and 3 to load structure.

To save the struct it is written to the file using fprintf and fscanf is used to load the record.

Program 38: Allow user to input up to 20 records into a struct array and save these to a file then reload back into struct array on demand

Notes:

The program creates a struct array of 20. It allows the user to input data write data to file or load data from file back into struct array.

It uses a for loop to cycle through the struct array when saving a loading.

It also gives the user an option to print struct on screen to print the struct array on screen.

Program 39: Allow user to input as many student records as they want, save to file and load from file

Notes:

To allow as many structs a linked list needs to be implemented. A pointer to the next struct is placed in the struct definition this allows the list to be cycled through. The last pointer is a Null pointer signifying the end of the linked list.

This program allows the user to insert as many records as needed, remove records, save records and load records form file.

It will continue until the user choses to quit to program by pressing 0