John Akujobi

Oct 10th, 2022

CSC 150 SO3

**Q 1**

In what ways are the initialization, repetition test, and update steps alike for a sentinel-controlled loop and an end-file-controlled loop?

How are they different?

* The initialization and update steps are the same for the two loops
* However, the repetition tests are different for both
  + In sentinel-control loops the control variable is checked if it is not equivalent value (variable != sentinel\_value). The while loop continues until the sentinel value is entered
  + while for the End file-controlled loop, the expression will be (variable != EOF).
  + The loop stops after the end of the file is detected by using the EOF() function which returns true when the end-of-file is reached.

**Q 2**

Write a program that computes and displays the sum of a collection of Celsius temperatures entered at the terminal until a sentinel value of -275 is entered.

/\*

Program that computes and displays the sum of a collection of Celsius temperatures entered at the terminal until a sentinel value of -275 is entered.

Created by: John Akujobi

Date: 30th October 2022

CSC 150 S03

\*/

#include <stdio.h>

void ProgramGreeting(void);//Function prototype of the greetings function

int main ()

{

//Variable declaration

int temp, temp\_sum = 0;

//Call the greetings function

ProgramGreeting();

//it checks and makes sure that the temp is not the sentinel value of -275 is entered.

while (temp != -275)

{

//Asks the user to enter a temperature

printf("\\nEnter a temperature in Celsius --->: ");

scanf("%d", &temp);

//Tried to apply input validation but it didn't work

//while (temp/temp != 1 && check = 1)

//{

// printf("Invalid input. Please enter a valid temperature: ");

//}

//increments temp\_sum by the value of temp

temp\_sum += temp;

printf("The sum so far is %5d\\n",temp\_sum);

}

//prints the final sum

printf("\\nThe total sum of the temperatures is %8d\\n\\n", temp\_sum);

return 0;

}

//Greets the user and tells them what the program does

void ProgramGreeting()

{

printf("\\nHello\\nThis program computes the sum of a collection of Celsius temperatures entered at the terminal\\n");

printf("Until a value of -275 is entered.\\nAnd it gives you the total sum\\n\\n");

return 0;

}

**Q 3**

Hand trace the program that follows given the following data:

4 2 8 4 1 4 2 1 9 3 3 1 -22 10 8 2 3 3 4 5

#include <stdio.h>

#define SPECIAL\_SLOPE 0.0

int main(void)

{

double slope, y2, y1, x2, x1;

printf("Enter 4 numbers separated by spaces.");

printf("\\nThe last two numbers cannot be the same,");

printf("\\nbut the program terminates if ");

printf("the first two are.\\n");

printf("\\nEnter four numbers> ");

scanf("%lf%lf%lf%lf", &y2, &y1, &x2, &x1);

//4 2 8 4

//y1=4, y2=2, x2=8, x1=4

for (slope = (y2 - y1) / (x2 - x1);

slope != SPECIAL\_SLOPE;

slope = (y2 - y1) / (x2 - x1))

{

printf("Slope is %5.2f.\\n", slope); //Slope is 0.50.

printf("\\nEnter four more numbers> ");

scanf("%lf%lf%lf%lf", &y2, &y1, &x2, &x1);

}

return (0);

}

/\*

4 2 8 4

y1=4, y2=2, x2=8, x1=4

(2-4)/(8-4)

---> "Slope is o.50."

1 4 2 1

y1=1, y2=4, x2=2, x1=1

(4-1)/(2-1)

---> "Slope is -3.00."

9 3 3 1

y1=9, y2=3, x2=3, x1=1

(3-9)/(3-1)

---> "Slope is 3.00."

-22 10 8 2

y1=-22, y2=10, x2=8, x1=2

(10-(-22))/(8-2)

---> "Slope is -5.33."

3 3 4 5

y1=3, y2=3, x2=4, x1=5

(3-3)/(4-5) //SPECIAL\_SLOPE

[No output because the slope is 0 and the loop ends]

\*/

**Q 4**

Rewrite the program in Review Question 3 so that it uses a while loop .

/\*

This program calculates the slope of a line given 4 numbers and prints the slope

until the user enters 4 numbers where the first two are the same.

Created by: John Akujobi

Date: 30th October 2022

CSC 150 S03

\*/

#include <stdio.h>

#define SPECIAL\_SLOPE 0.0

int main(void)

{

//Variable declaration

double slope, y2, y1, x2, x1;

//Tells the user what the program does and how to use it

printf("Enter 4 numbers separated by spaces.\\n");

printf("E.g 2 4 6 8 which represents y2 y1 x2 x1\\n");

printf("\\nThe last two numbers cannot be the same,");

printf("\\nbut the program terminates if the first two are.\\n");

//Asks the user to enter 4 numbers

printf("\\nEnter four numbers --> ");

scanf("%lf%lf%lf%lf", &y2, &y1, &x2, &x1);

//calculates the slope

slope = (y2 - y1) / (x2 - x1);

while (slope != SPECIAL\_SLOPE)

{

//prints the slope from the last numbers entered

printf("Slope is %5.2f.\\n", slope); //Slope is 0.50.

//asks the user to enter 4 more numbers to calculate another slope

printf("\\nEnter four more numbers> ");

scanf("%lf%lf%lf%lf", &y2, &y1, &x2, &x1);

//calculates the slope

slope = (y2 - y1) / (x2 - x1);

}

return (0);

}

**Q 5**

Rewrite the program segment that follows, using a for loop:

count = 0;

i = 0;

while (i < n)

{

scanf("%d", &x);

if (x == i)

{

++count;

}

++i;

}

//With a For loop

for (i = 0, count = 0; i < n; i++)

{

scanf("%d", &x);

if (x == i)

{

++count;

}

}

**Q 6**

Rewrite this for loop heading, omitting any invalid semicolons. for (i = n; i < max; ++i;);

for ( i = n; i < max; ++i)

**Q 7**

Write a do-while loop that repeatedly prompts for and takes input until a value in the range 0 through 15 inclusive is input.

Include code that prevents the loop from executing forever on input of a wrong data type.

do

{

//initializes or resets num to 0

//In order to stop an infinite loop on entry of other data types

int num = 0;

//Asks the user to enter a number for the first time

printf("Enter a number between 0 and 15: ");

scanf("%d", &num);

//Checks if the number is in the range 0 to 15

if (num\*1 < 0 || num\*1 > 15)

{

//Print error message

//And ask the user to enter a number again

printf("Whoops! Invalid input.\\n");

printf("Please Enter a number between 0 and 15: ");

scanf("%d", &num);

}

//printf("The number you entered is %d\\n", num);

} while (num\*1 < 0 || num\*1 > 15);