GNG 1106 2020

LAB 3 REPORT

Exercise: Volume of a liquid in a horizontal cylinder

Corrected Program

```
/*-------------------------------------------------------------------
 File:  CylinderVolume.c  (Lab 3)
 Author: Gilbert Arbez, Fall 2018
 Description: Calculates how the volume changes for different depths
          in a horizontal cylinder.
-------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// Define symbolic constant
#define N       50    // number of points to compute
#define TRUE    1
#define FALSE   0

// Prototypes
void computeDisplayVolume(double, double);
/*-------------------------------------------------------------------
Function: main
Description:  Gets from the user the radius and length of the cylinder,
          calls computeDisplayVolume to compute and display a table
          of how the volume changes with depth of a liquid in
          the cylinder.
-------------------------------------------------------------------*/
void main()
{
   // Variable declarations
   double radius;  // cylinder radius
```

```
   double length;  // cylinder length
   int flag;   // sentinelle for controlling data input
   // Get input from user, the cylinder radius and length
   do
   {
      flag = 1;
      printf("Please give cylider radius and length: ");
      scanf("%lf %lf",&radius, &length);
      if(radius <= 0.0 || length <= 0.0)
       {
          printf("Both values must be greater than zero.\n");
          flag = 0;
       }
   } while(flag == 0);
   // Compute/display depth/volume data
   computeDisplayVolume(radius, length);
}
/*-------------------------------------------------------------------------
Function: computeDisplayVolume
Parameter
   radius - radius of the horizontal cylinder
   length - length of the horizontal cylinder
Description: Computes and displays depth/volume data points in a table
          that varies the depth of the liquid from 0 to the cylinder diameter.
          N such data points are computed (i.e. the increment in the
                                    value of h is 2r/N).
-------------------------------------------------------------------------*/
void computeDisplayVolume(double radius, double length)
{
   // Declaration of variables
```

```
   double increment;   // how to increment the depth
   double h;          // depth value
   double volume;     // volume value
   int i ;            // loop counter
   // setup the variables
   increment = radius/N;
   h = 0.0;
   // Table Header
   printf("The change in liquid volume of the cylinder with radius %.2f \nand length %.2f as depth
changes is as follows.\n",
         radius, length);
   printf("%10s    %10s\n","Depth", "Volume");
   printf("------------------------\n");
   // Loop for calculating each of the n depth/volume points.
   for(h=0.0 ; h< (N-0.5)*increment; h=h+2*increment)
   {
     volume = pow(radius,2)*acos((radius-h)/radius);
     volume = volume - (radius - h)*sqrt(2.0*radius*h - pow(h,2));
     volume = volume*length;
     // Display the row with ocmputed values
     printf("%10.2f    %10.2f\n", h, volume);
   }
}
```

Exercise: Rocket Speed

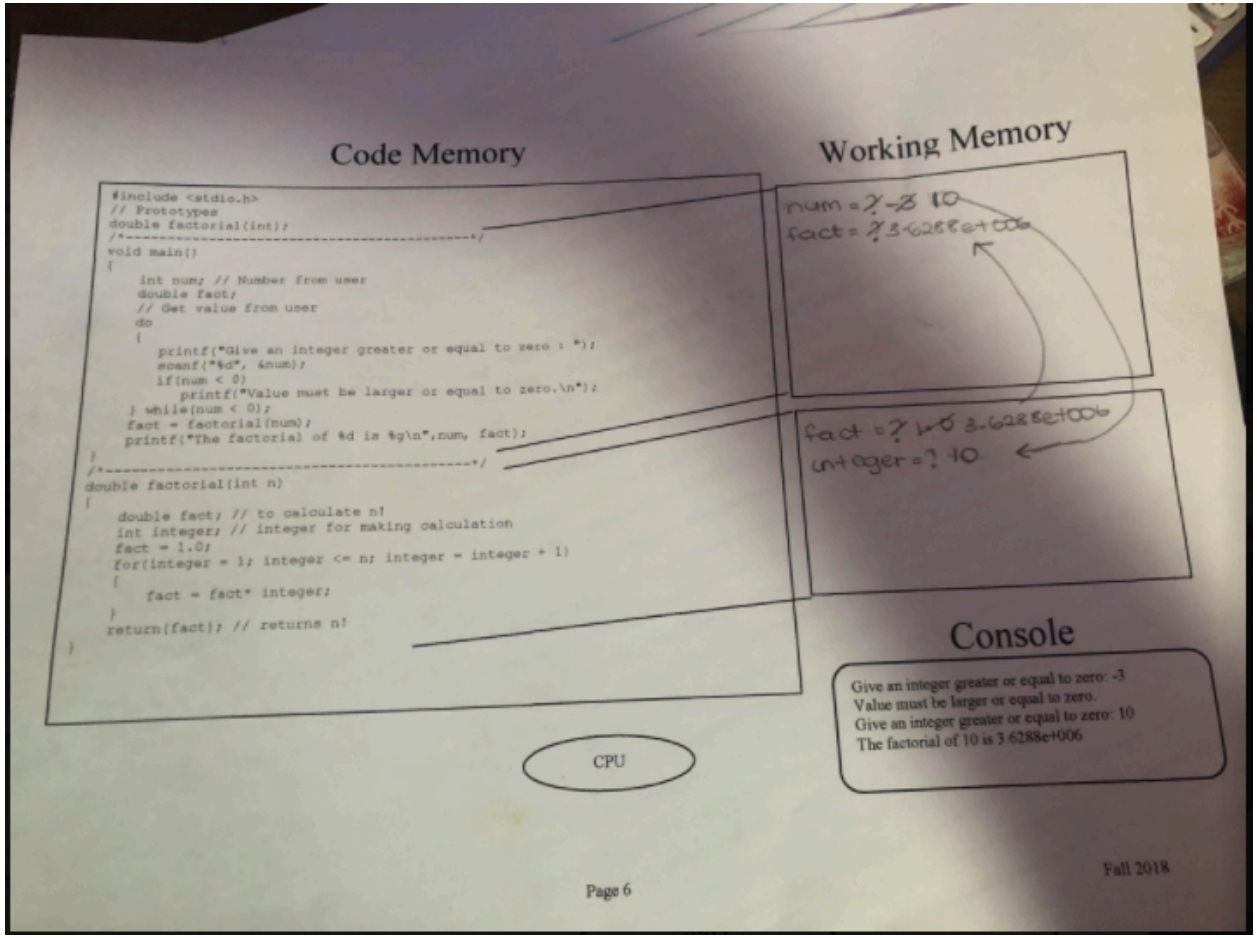
```
/*-----------------------------------------------
 File: RocketTrajectory.c
 Author: Gilbert Arbez, Fall 2018
 Description: Calculates and displays the change
 in velocity of rocket as a function of time.
-----------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
// Define symbolic constant
#define TRUE 1
#define FALSE 0
#define NUM_ROWS 25
// Function prototypes
void displayTrajectoryTable(double, double);
double computeVelocity(double);
/*-----------------------------------------------------
Function: main
Description: Gets valid input from user for start and end
        times to create table rows, then
        displayTrajectoryTable to display the trajectory
        table.
-----------------------------------------------------*/
void main()
{
   double startTime, endTime;  // times for creating table.
   int badInput=0;  // Flag used in loop to get user input
   printf("Enter start time: ");
   scanf("%lf",&startTime);
```

```
   printf("Enter end time: ");
   scanf("%lf",&endTime);
   if(startTime>=endTime)
   badInput=1;
   do
   {
      if(badInput==1)
     {
         printf("start time must be less than end time!! Enter again\n");
         printf("Enter start time: ");
         scanf(" %lf",&startTime);
         printf("Enter end time: ");
         scanf(" %lf",&endTime);
     }
     if(startTime<endTime)
     badInput=0;
   }
   while(badInput==1);
   // Display Table
   displayTrajectoryTable(startTime, endTime);
   return 0;
}
/*-------------------------------------------------
Function: displayTrajectoryTable
Parameters:
   start:  value of time at the start of the table
   end:    value of time at the end of the table
Description:  Displays on the console a table of
            NUM_ROWS rows to show how the rocket
            velocity changes between the start time
```

```
            and the end time.
-------------------------------------------------------*/
void displayTrajectoryTable(double start, double end)
{
  int i;
  double v,t=start,h;
  h=(end-start)/(NUM_ROWS-1);
  printf("%f\n",end);
  printf("Time            Velocity\n");
  printf("----------------------------\n");
  while (TRUE)
  {
     v=computeVelocity(t);
     printf("%10.2f%15.2f\n",t,v);
     t=t+h;
     if (t>end)
     break;
  }
}

/*-------------------------------------------------------------
Function: computeVelocity
Parameter:
    t - value of time
Returns: Value of velocity at time t.
Description: Computes the velocity of the rocket at time t.
-------------------------------------------------------------*/
double computeVelocity(double t)
{
   double vel;  // rocket velocity
```

```
    if(t<0)
    vel=0;
    else if(t>=0&&t<=10)
    vel=11*t*t-5*t;
    else if(t<=20)
    vel=1100-5*t;
    else if(t<=30)
    vel=50*t+2*pow((t-20),2);
    else if(t>30)
    vel=1520*exp(-.2*(t-30));
    return(vel);
}
```


Code Memory
#include <stdio.h>
// Prototypes
double factorial(int);
/*--*/
void main()
{
int num; // Number from user
double fact;
// Get value from user
do
{
printf("Give an integer greater or equal to zero : ");
scanf("%d", &num);
if(num < 0)
printf("Value must be larger or equal to zero.\n");
} while(num < 0);
fact = factorial(num);
printf("The factorial of %d is %g\n",num, fact);
}
/*--*/
double factorial(int n)
{
double fact; // to calculate n!
int integer; // integer for making calculation
fact = 1.0;
for(integer = 1; integer <= n; integer = integer + 1)
{
fact = fact* integer;
}
return(fact); // returns n!
}
Working Memory
num = ? -3 10
fact = ? 3.6288e+006
fact = ? 1 3.6288e+006
integer = ? 10
CPU
Console
Give an integer greater or equal to zero: -3
Value must be larger or equal to zero.
Give an integer greater or equal to zero: 10
The factorial of 10 is 3.6288e+006