Software Engineering (CS401)

Practical Lab Examination

**U19CS012**

Q1.)

a.) Write a C program having structure as global variable. This structure can have more than two fields. Except for one field, you can initialize values to all fields in the structure. Run Splint for this C code and report the error generated. (This code should generate error as you have one uninitialized field in structure)

**Code**

*#include* <stdio.h>

*// Global Structure having More than Two Fields*

struct student

{

    char \*name;

    int age;

    float per;

};

float get\_percent();

int main()

{

    printf("Structure as Global Variable\n");

    printf("Except one field, you can Initialize values to all fields in the structure.\n");

    struct student o =

        {

            .name = "Nobita",

            .age = 20,

        };

*// One Field where Intialization is performed Later*

    o.per = get\_percent();

    printf("\nName    : %s", o.name);

    printf("\nAge     : %d", o.age);

    printf("\nPercent : %f", o.per);

*return* 0;

}

float get\_percent()

{

*// Marks Related to Student can be Entered (For Sample Calculation, taken as shown below)*

    float physic\_marks = 95;

    float chemistry\_marks = 90;

    float maths\_marks = 100;

    printf("Physics     : %f\n", physic\_marks);

    printf("Chemistry   : %f\n", chemistry\_marks);

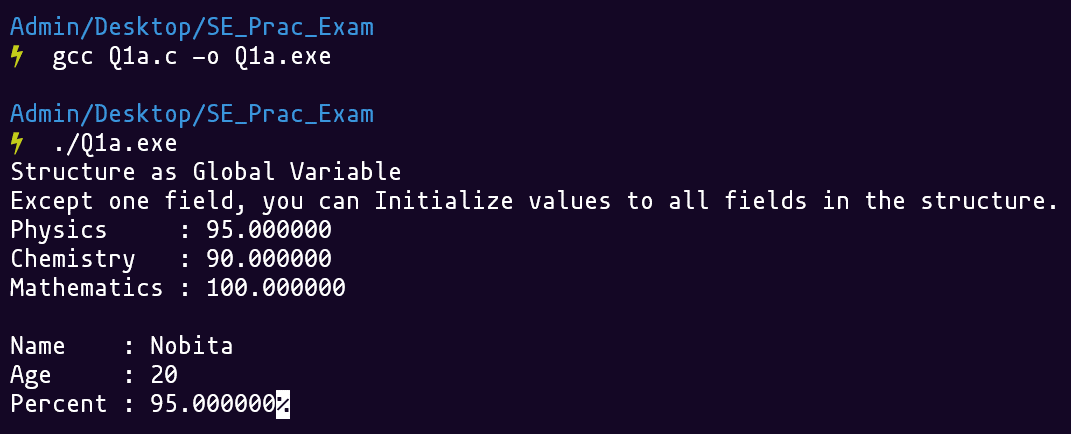
    printf("Mathematics : %f\n", maths\_marks);

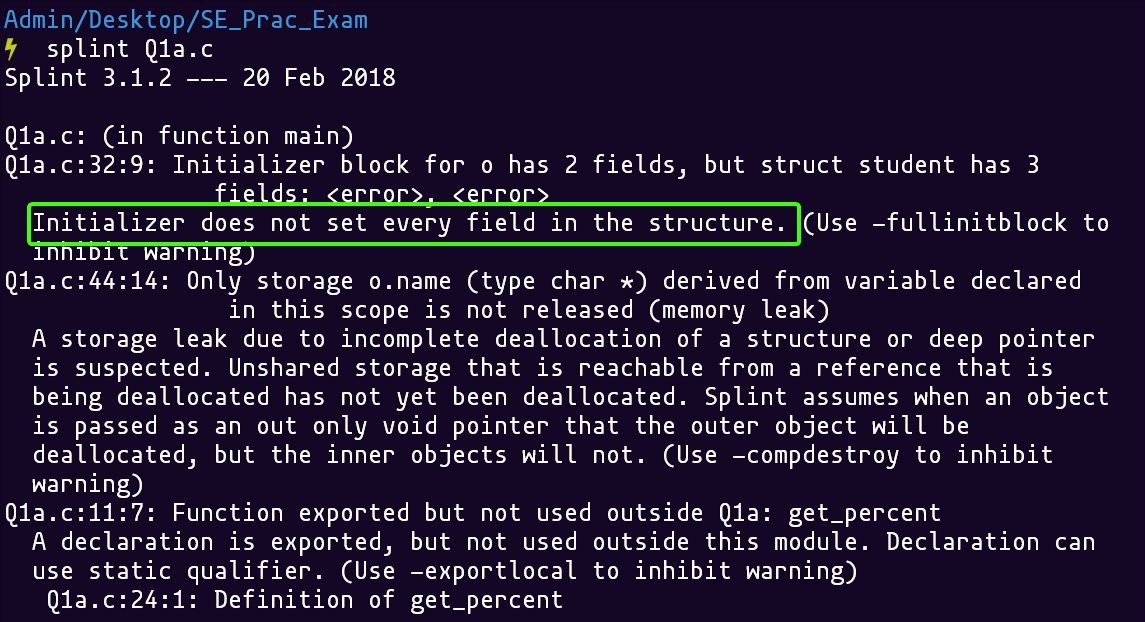
    float percent = (physic\_marks + chemistry\_marks + maths\_marks) / 3;

*return* percent;

}

**Output**





b.) Write a C program having some global variables that are declared but not used anywhere in the code. Run Splint for this C code and report the error generated.

**Code**

*#include* <stdio.h>

*// Unused variables*

*// Global Variable [Uninitialized]*

int global\_var\_1;

*// Global Variable [Initialized]*

int global\_var\_2 = 10;

int main()

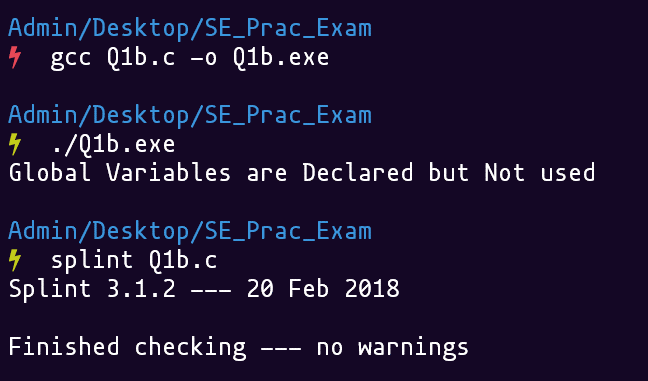
{

    printf("Global Variables are Declared but Not used\n");

*return* 0;

}

**Output**



Q2.) Consider a chocolate vending machine that contains **Four** types of chocolates

|  |  |
| --- | --- |
| **Chocolate Type** | **Price** |
| Coffeestick | Rs 10 |
| Milkbar | Rs 20 |
| Silkchocolate | Rs 50 |
| DarkChocolate | Rs 50 |

To select a coffeestick, one inserts 10 rupee note. This process is similar with other types of chocolate. For example, to collect a milkbar, one has to insert 20 rupee note.

Create a SPIN model of these **two processes**, i.e., vender for vending machine having limitless supply and **customer** having limitless appetite for chocolates.

You model will use message channels for communication between customer and vender. In the second step, you will set a limit for vending machine, i.e., at most 20 coffeesticks, 20 milkbars, 10 silkchocolates and 15 darkchocolates can be supplied in a single run (after this, vending machine needs to refill). You program should print the amount collected by vending machine at the time where it needs refill or no more chocolate (of any type) can be provided thereafter.

**Code**

#define MAX\_SUPPLY 10000

*// chan STDIN = [4] of int;*

int coffeestick\_available = MAX\_SUPPLY;

int milkbars\_available = MAX\_SUPPLY;

int silkchocolates\_available = MAX\_SUPPLY;

int darkchocolates\_available = MAX\_SUPPLY;

int coffeestick\_cost = 10;

int milkbar\_cost = 20;

int silkchocolate\_cost = 50;

int darkchocolate\_cost = 50;

int coffeestick\_count = 0;

int milkbar\_count = 0;

int silkchocolate\_count = 0;

int darkchocolate\_count = 0;

int total\_amount = 0;

proctype print\_amount\_collected()

{

*// do*

*// ::*

    printf("Chocolate Vending Machine Needs to be Refilled!\n\n");

    printf("AMOUNT COLLECTED BY MACHINE\n\n");

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

    printf("Coffee Stick Cost = %d x %d = %d\n", coffeestick\_count, coffeestick\_cost, coffeestick\_count\*coffeestick\_cost);

    printf("Milkbar Cost = %d x %d = %d\n", milkbar\_count, milkbar\_cost, milkbar\_count\*milkbar\_cost);

    printf("Silk Chocolate Cost = %d x %d = %d\n", silkchocolate\_count, silkchocolate\_cost, silkchocolate\_count\*silkchocolate\_cost);

    printf("Dark Chocolate Cost = %d x %d = %d\n", darkchocolate\_count, darkchocolate\_cost, darkchocolate\_count\*darkchocolate\_cost);

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

    printf("AMOUNT COLLECTED BY MACHINE : %d\n", total\_amount);

*// break;*

*// od*

}

proctype vender(int id)

{

*// Having Limitless Supply*

*// Having Limited Supply*

*assert*(id<=4);

*do*

    :: printf("Vender start\n");

*od*

*if*

    :: (coffeestick\_count == coffeestick\_available) -> run  print\_amount\_collected();

    :: *else* -> *if*

               :: (milkbar\_count == milkbars\_available) -> run  print\_amount\_collected();

               :: *else* -> *if*

                          :: (silkchocolate\_count == silkchocolates\_available) -> run  print\_amount\_collected();

                          :: *else* *if*

                                  :: (darkchocolate\_count<darkchocolates\_available) -> run print\_amount\_collected();

*fi*

*fi*

*fi*

*fi*

*assert*(coffeestick\_count<coffeestick\_available);

*assert*(milkbar\_count<milkbars\_available);

*assert*(silkchocolate\_count<silkchocolates\_available);

*assert*(darkchocolate\_count<darkchocolates\_available);

*if*

    :: (id == 1) -> coffeestick\_count = coffeestick\_count + 1; total\_amount = total\_amount + coffeestick\_cost;

    :: *else* -> *if*

                :: (id == 2) -> milkbar\_count = milkbar\_count + 1; total\_amount = total\_amount + milkbar\_cost;

*else* -> *if*

                        :: (id == 3) -> silkchocolate\_count = silkchocolate\_count + 1; total\_amount = total\_amount + silkchocolate\_cost;

                        :: *else* -> darkchocolate\_count = darkchocolate\_count + 1; total\_amount = total\_amount + darkchocolate\_cost;

*fi*

*fi*

*fi*

}

proctype customer()

{

    printf("Customer start\n");

*// Customer Needs to Enter the Order*

    printf("Enter your Choice\n 1 - coffeestick 2 - milkbar 3 - silkchocolates 4 - darkchocolates\n");

    int customer\_choice = 1;

*// cin >> customer\_choice;*

    run vender(customer\_choice);

}

init

{

    printf("Welcome to Chocolate Vending Machine.\n\n");

*// This is Boolean flag for Part - 2*

    bool has\_limited\_supply = false;

    has\_limited\_supply = true;

*if*

    :: (has\_limited\_supply == true) -> coffeestick\_available = 20; milkbars\_available = 20; silkchocolates\_available = 10; darkchocolates\_available = 15;

*// :: else -> break;*

*fi*

    int customer\_orders = 21;

    int i = 0;

*do*

    :: i >= customer\_orders -> *break*

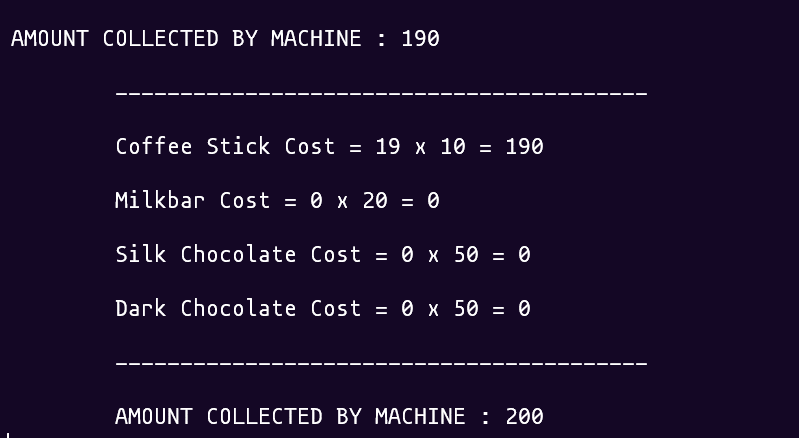
    :: *else* -> run customer();

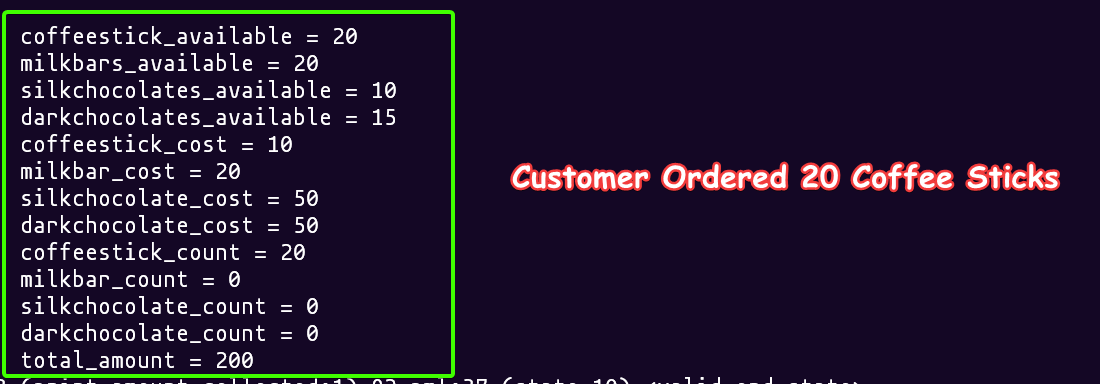
               i++

*od*

}

**Output**





**SUBMITTED BY**:

U19CS012

BHAGYA VINOD RANA