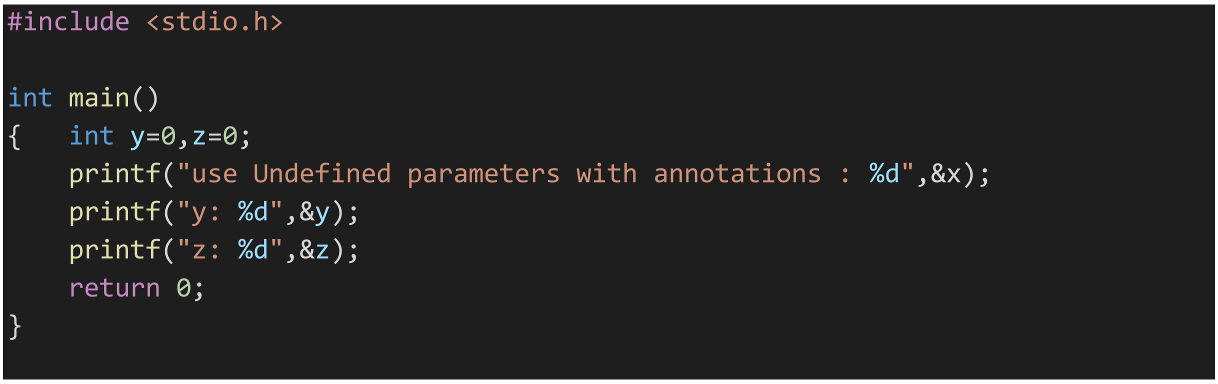
SE PRACTICAL EXAM

U19CS082 SOURABH PATEL

Odd Number

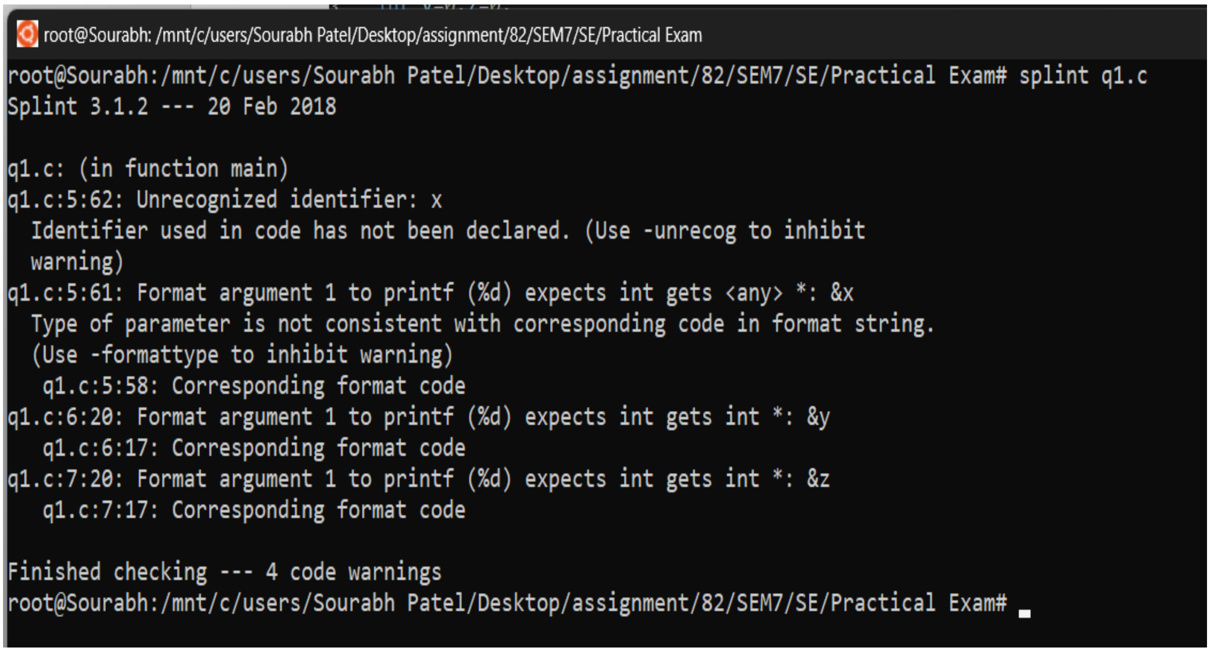
1) Demonstrate use of Undefined parameters with annotations in Splint tool and C Compiler.

CODE:



```c
#include <stdio.h>

int main()
{   int y=0,z=0;
    printf("use Undefined parameters with annotations : %d",&x);
    printf("y: %d",&y);
    printf("z: %d",&z);
    return 0;
}
```

OUTPUT:

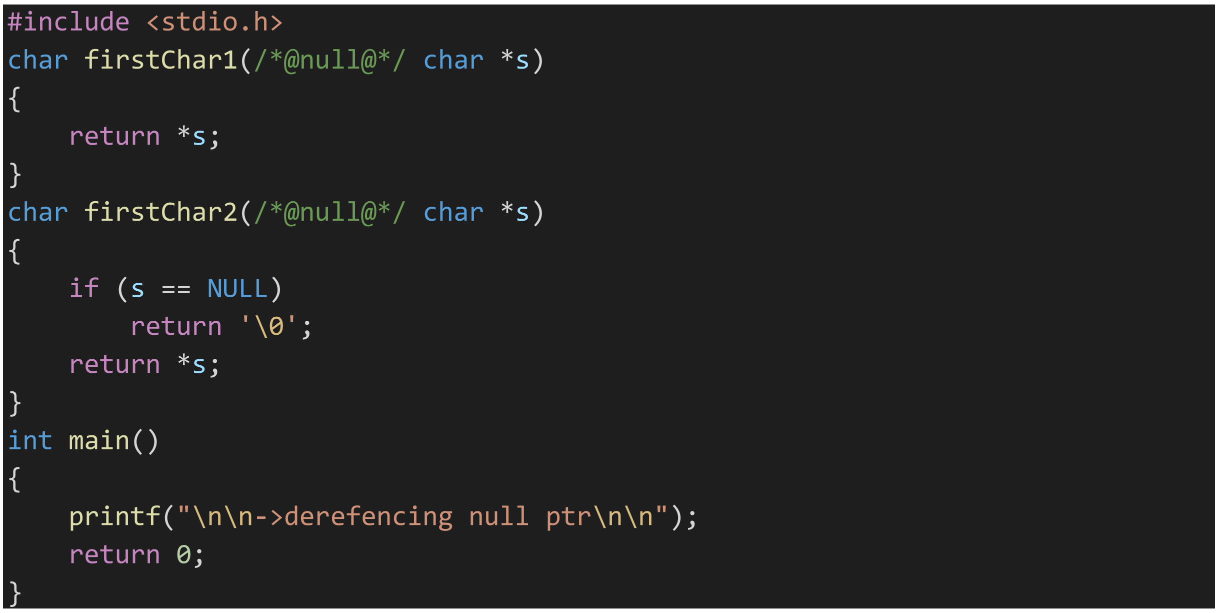


root@Sourabh: /mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam

```
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam# splint q1.c
Splint 3.1.2 --- 20 Feb 2018

q1.c: (in function main)
q1.c:5:62: Unrecognized identifier: x
  Identifier used in code has not been declared. (Use -unrecog to inhibit
  warning)
q1.c:5:61: Format argument 1 to printf (%d) expects int gets <any> *: &x
  Type of parameter is not consistent with corresponding code in format string.
  (Use -formattype to inhibit warning)
   q1.c:5:58: Corresponding format code
q1.c:6:20: Format argument 1 to printf (%d) expects int gets int *: &y
   q1.c:6:17: Corresponding format code
q1.c:7:20: Format argument 1 to printf (%d) expects int gets int *: &z
   q1.c:7:17: Corresponding format code

Finished checking --- 4 code warnings
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam#
```

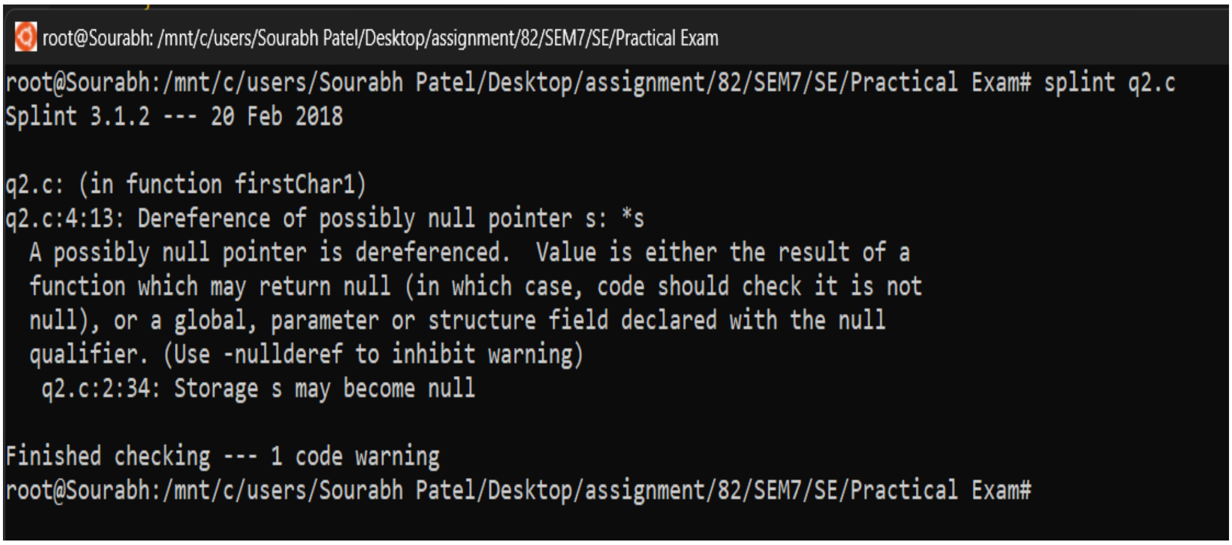
2) Write a C program to indicate null pointer and dereferenced pointer and Run Splint for this C code and report the error generated.

CODE:



```c
#include <stdio.h>
char firstChar1(/*@null@*/ char *s)
{
    return *s;
}
char firstChar2(/*@null@*/ char *s)
{
    if (s == NULL)
        return '\0';
    return *s;
}
int main()
{
    printf("\n\n->derefencing null ptr\n\n");
    return 0;
}
```

OUTPUT:



root@Sourabh: /mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam

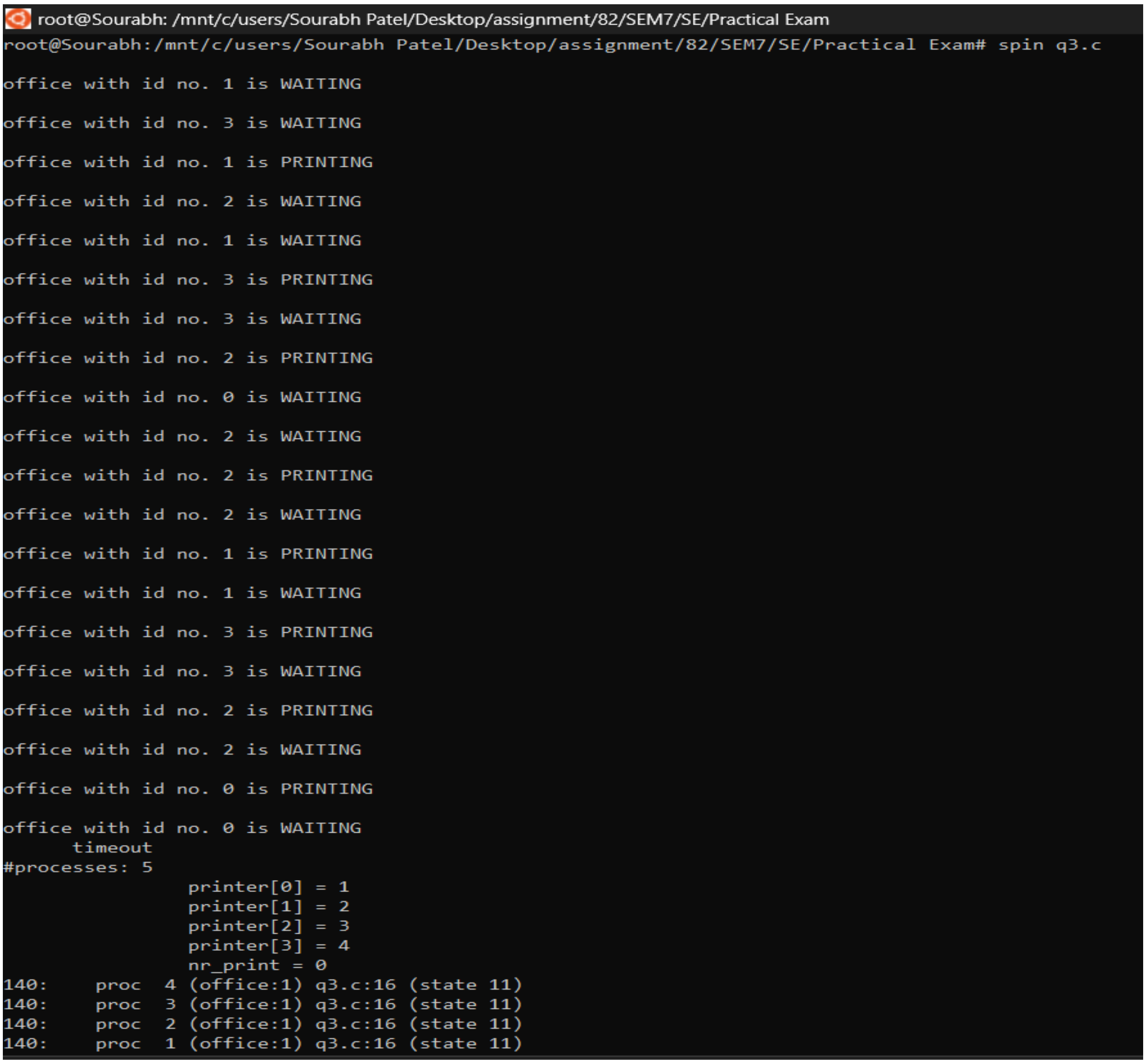
```
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam# splint q2.c
Splint 3.1.2 --- 20 Feb 2018

q2.c: (in function firstChar1)
q2.c:4:13: Dereference of possibly null pointer s: *s
  A possibly null pointer is dereferenced.  Value is either the result of a
  function which may return null (in which case, code should check it is not
  null), or a global, parameter or structure field declared with the null
  qualifier. (Use -nullderef to inhibit warning)
   q2.c:2:34: Storage s may become null

Finished checking --- 1 code warning
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam#
```

3) Model an office where two computer systems are hooked up to a single printer. Establish mutual exclusion between the two computer systems using a global variable and atomic "test & set" operations. Verify in SPIN model that it is never the case that both systems print at the same time.

CODE:

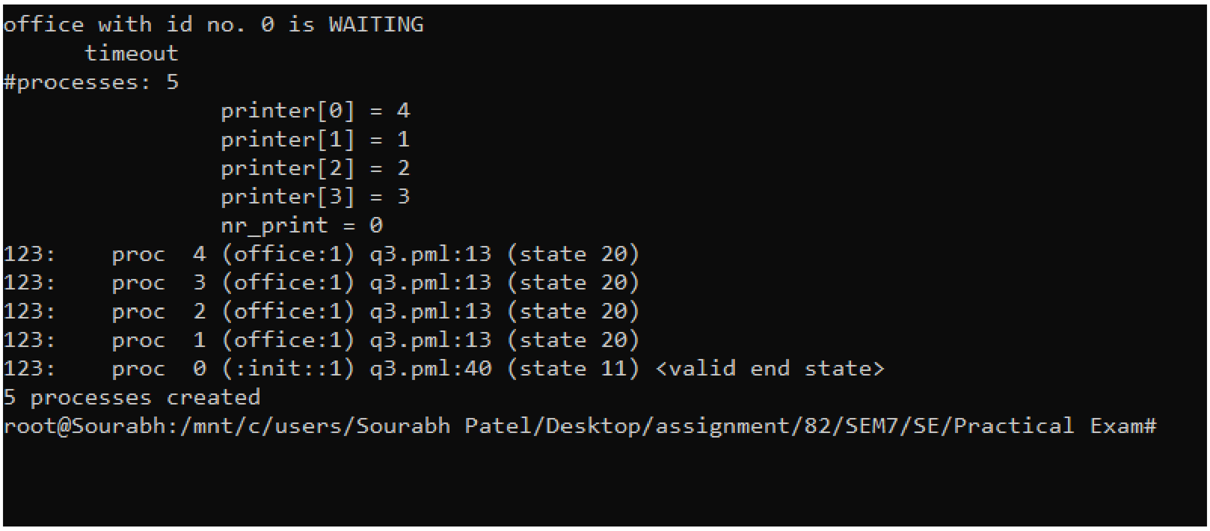
```
#define N 4
byte printer[N];
byte nr_print;
proctype office(byte id)
{
    Think:
        printf("\noffice with id no. %d is WAITING\n",id);
        if
        :: atomic { printer[id] == 0 -> printer[id] = id + 1; };
        :: atomic { printer[(id + 1)%N] == 0 -> printer[(id +
1)%N] = id + 1; };
        fi;
    One:
        if
        :: atomic
        {
            printer[id] == id + 1 -> printer[(id + 1)%N] == 0 ->
printer[(id + 1)%N] = id + 1;
            nr_print++;
        }
        :: atomic
        {
            printer[id] == 0 -> printer[(id + 1)%N] == id + 1 ->
printer[id] = id + 1;
            nr_print++;
        }
        fi;
    Eat:
        printf("\noffice with id no. %d is PRINTING\n",id);
        d_step { nr_print--; printer[(id + 1)%N] = 0; printer[id]
= 0;}
    goto Think;
}

init {
    atomic
    {
        byte i = 0;
        do
        :: i < N -> run office(i); i++;
        :: else -> break;
        od;
    }
}
```

OUTPUT:



root@Sourabh: /mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam

```
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam# spin q3.c

office with id no. 1 is WAITING

office with id no. 3 is WAITING

office with id no. 1 is PRINTING

office with id no. 2 is WAITING

office with id no. 1 is WAITING

office with id no. 3 is PRINTING

office with id no. 3 is WAITING

office with id no. 2 is PRINTING

office with id no. 0 is WAITING

office with id no. 2 is WAITING

office with id no. 2 is PRINTING

office with id no. 2 is WAITING

office with id no. 1 is PRINTING

office with id no. 1 is WAITING

office with id no. 3 is PRINTING

office with id no. 3 is WAITING

office with id no. 2 is PRINTING

office with id no. 2 is WAITING

office with id no. 0 is PRINTING

office with id no. 0 is WAITING
      timeout
#processes: 5
                printer[0] = 1
                printer[1] = 2
                printer[2] = 3
                printer[3] = 4
                nr_print = 0
140:    proc  4 (office:1) q3.c:16 (state 11)
140:    proc  3 (office:1) q3.c:16 (state 11)
140:    proc  2 (office:1) q3.c:16 (state 11)
140:    proc  1 (office:1) q3.c:16 (state 11)
```



```
office with id no. 0 is WAITING
      timeout
#processes: 5
                printer[0] = 4
                printer[1] = 1
                printer[2] = 2
                printer[3] = 3
                nr_print = 0
123:    proc  4 (office:1) q3.pml:13 (state 20)
123:    proc  3 (office:1) q3.pml:13 (state 20)
123:    proc  2 (office:1) q3.pml:13 (state 20)
123:    proc  1 (office:1) q3.pml:13 (state 20)
123:    proc  0 (:init::1) q3.pml:40 (state 11) <valid end state>
5 processes created
root@Sourabh:/mnt/c/users/Sourabh Patel/Desktop/assignment/82/SEM7/SE/Practical Exam#
```