Software Engineering

Assignment 1

Student Details

Name: Krunal Rank
Adm. No: U18CO081

4

Dereferencing a possibly null pointer

```c
char return_char(/*@null@*/ char *s)
{
   return *s;
}
```

```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/4th_Year/SE/Assignment_1$ splint ./1.c
Splint 3.1.2 --- 20 Feb 2018

1.c: (in function return_char)
1.c:4:14: 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)
   1.c:3:36: Storage s may become null

Finished checking --- 1 code warning
```

Using possibly undefined storage or returning storage that is not properly defined.

```c
extern int getVal (/*@in@*/ int *x);

int function_2(/*@out@*/ int *x)
{
   return getVal(x);
}
```

```
krhero@hellblazer:/mnt/0FB812900FB81290/BTech/Assignments/4th_Year/SE/Assignment_1$ splint ./2.c
Splint 3.1.2 --- 20 Feb 2018

2.c: (in function function 2)
2.c:5:19: Passed storage x not completely defined (*x is undefined): getVal (x)
  Storage derivable from a parameter, return value or global is not defined.
  Use /*@out@*/ to denote passed or returned storage which need not be defined.
  (Use -compdef to inhibit warning)

Finished checking --- 1 code warning
```

Type mismatches, with greater precision and flexibility than provided by C compilers and Violations of information hiding

```c
#include <stdio.h>
typedef /*@abstract@*/ char *mstring;

int isPalindrome(mstring s){
   char *current = (char *)s;
   int i, len = (int)strlen(s);
   for (i = 0; i <= (len + 1) / 2; i++){
       if (current[i] != s[len - i - 1]) return 0;
   }
   return 1;
}

bool callPal(void){
   return (isPalindrome("bob"));
}
```

```
krhero@krhero-Lenovo-ideapad-330-15IKB:/media/krhero/0FB812900FB81290/BTech/Assignments/4th_Year/SE/Assignment_1$ splint ./3.c
Splint 3.1.2 --- 20 Feb 2018

3.c: (in function callPal)
3.c:14:12: Return value type int does not match declared type bool:
             (isPalindrome("bob"))
  Types are incompatible. (Use -type to inhibit warning)
3.c:4:5: Function exported but not used outside 3: isPalindrome
  A declaration is exported, but not used outside this module. Declaration can
  use static qualifier. (Use -exportlocal to inhibit warning)
   3.c:11:1: Definition of isPalindrome

Finished checking --- 2 code warnings
```

Memory management errors including uses of dangling references and memory leaks

```c
extern /*@only@*/ int *glob;

/*@only@*/ int *

f(/*@only@*/ int *x, int *y,
```

```
 int *z)

/*@globals glob;@*/

{

   int *m = (int *)

       malloc(sizeof(int));

   glob = y;

   free(x);

   *m = *x;

   return z;
}
```

```
4.c:21:11: Variable x used after being released
  Memory is used after it has been released (either by passing as an only param
  or assigning to an only global). (Use -usereleased to inhibit warning)
   4.c:19:10: Storage x released
4.c:23:12: Implicitly temp storage z returned as only: z
4.c:23:14: Fresh storage m not released before return
  A memory leak has been detected. Storage allocated locally is not released
  before the last reference to it is lost. (Use -mustfreefresh to inhibit
  warning)
   4.c:15:29: Fresh storage m created
```

Dangerous aliasing

```
#include <string.h>

void capitalize(/*@out@*/ char *s, char *t)
{
   strcpy(s, t);
   *s = toupper(*s);
}
```

```
5.c: (in function capitalize)
5.c:5:12: Parameter 1 (s) to function strcpy is declared unique but may be
             aliased externally by parameter 2 (t)
  A unique or only parameter may be aliased by some other parameter or visible
  global. (Use -mayaliasunique to inhibit warning)
```