Splint main.c

```
varshu@varshu-VirtualBox:~/Uni_alumni/src$ splint main.c -I ../inc
Splint 3.1.2 --- 21 Feb 2021

main.c: (in function main)
main.c:8:8: Test expression for while not boolean, type int: 1
  Test expression type is not boolean or int. (Use -predboolint to inhibit
  warning)
main.c:9:3: Return value (type int) ignored: system("clear")
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
main.c:12:3: Return value (type int) ignored: scanf("%d", &ch)
main.c:16:17: Return value (type int) ignored: system("clear")
main.c:18:17: Return value (type int) ignored: scanf("%d", &alu...
main.c:22:25: Return value (type int) ignored: system("clear")
main.c:25:29: Return value (type int) ignored: signUp(0 + 1)
main.c:27:29: Return value (type int) ignored: signUp(id + 1)
main.c:30:25: Return value (type int) ignored: system("clear")
main.c:35:5: Return value (type int) ignored: scanf("%d", &adm...
main.c:39:20: Return value (type int) ignored: system("clear")
main.c:44:20: Return value (type int) ignored: system("clear")
main.c:46:20: Return value (type int) ignored: deleteAlumniInfo()
main.c:51:20: Unreachable code: break
  This code will never be reached on any possible execution. (Use -unreachable
  to inhibit warning)
main.c:57:26: Fall through case (no preceding break)
  Execution falls through from the previous case (use /*@fallthrough@*/ to mark
  fallthrough cases). (Use -casebreak to inhibit warning)
main.c:60:25: Unreachable code: break
main.c:65:11: Fall through case (no preceding break)
main.c:66:10: Return value (type int) ignored: system("clear")
main.c:67:16: Test expression for while not boolean, type int: 1
main.c:69:4: Return value (type int) ignored: scanf("%d", &stu_ch)
main.c:73:19: Return value (type int) ignored: system("clear")
```

Splint admin.c

```
varshu@varshu-VirtualBox:~/Uni_alumni/src$ splint admin.c -I ../inc
Splint 3.1.2 --- 21 Feb 2021

admin.c:9:17: Parse Error. (For help on parse errors, see splint -help
                parseerrors.)
*** Cannot continue.
```

Splint validations.c

```
varshu@varshu-VirtualBox:~/Uni_alumni/src$ splint validations.c -I ../inc
Splint 3.1.2 --- 21 Feb 2021

validations.c:12:12: Parse Error. (For help on parse errors, see splint -help
                     parseerrors.)
*** Cannot continue.
```

Splint alumni.c

```
varshu@varshu-VirtualBox:~/Uni_alumni/src$ splint alumni.c -I ../inc
Splint 3.1.2 --- 21 Feb 2021

alumni.c: (in function signUp)
alumni.c:10:2: Return value (type int) ignored: getchar()
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
alumni.c:11:10: Arrow access from possibly null pointer sn: sn->name
  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)
   alumni.c:6:15: Storage sn may become null
alumni.c:11:2: Return value (type char *) ignored: fgets(sn->name, ...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalother to inhibit warning)
alumni.c:14:3: Return value (type int) ignored: scanf("%s", sn->...
alumni.c:17:2: Return value (type int) ignored: scanf("%s", sn->...
alumni.c:18:2: Return value (type int) ignored: getchar()
alumni.c:20:2: Return value (type int) ignored: scanf("%ld", &sn...
alumni.c:23:3: Return value (type int) ignored: scanf("%ld", &sn...
alumni.c:26:2: Return value (type int) ignored: scanf("%d", &sn-...
alumni.c:28:2: Return value (type int) ignored: scanf("%d", &sn-...
alumni.c:30:2: Return value (type int) ignored: scanf("%s", sn->...
alumni.c:32:4: Arrow access from possibly null pointer ln: ln->uni_id
   alumni.c:7:14: Storage ln may become null
alumni.c:34:2: Return value (type int) ignored: scanf("%s", ln->...
alumni.c:36:2: Return value (type int) ignored: scanf("%s", ln->...
alumni.c:40:3: Return value (type int) ignored: scanf("%s", ln->...
alumni.c:46:2: Return value (type int) ignored: signUpWrite(sn, 0)
alumni.c:47:2: Return value (type int) ignored: loginWrite(ln, 0)
alumni.c:48:11: Fresh storage sn not released before return
```

Splint student.c

```
varshu@varshu-VirtualBox:~/Uni_alumni/src$ splint student.c -I ../inc
Splint 3.1.2 --- 21 Feb 2021

student.c: (in function studSignUp)
student.c:10:15: Arrow access from possibly null pointer ss: ss->name
  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)
   student.c:6:14: Storage ss may become null
student.c:10:2: Return value (type int) ignored: scanf("%s", ss->...
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
student.c:13:3: Return value (type int) ignored: scanf("%s", ss->...
student.c:16:2: Return value (type int) ignored: scanf("%s", ss->...
student.c:18:2: Return value (type int) ignored: scanf("%ld", &ss...
student.c:21:3: Return value (type int) ignored: scanf("%ld", &ss...
student.c:24:2: Return value (type int) ignored: scanf("%s", ss->...
student.c:26:4: Arrow access from possibly null pointer sl: sl->regno
   student.c:7:14: Storage sl may become null
student.c:28:2: Return value (type int) ignored: scanf("%s", sl->...
student.c:30:2: Return value (type int) ignored: scanf("%s", sl->...
student.c:34:3: Return value (type int) ignored: scanf("%s", sl->...
student.c:36:2: Return value (type int) ignored: studSignUpWrite(...
student.c:37:2: Return value (type int) ignored: studLoginWrite(s...
student.c:38:11: Fresh storage ss 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)
   student.c:6:37: Fresh storage ss created
student.c:38:11: Fresh storage sl not released before return
   student.c:7:37: Fresh storage sl created
student.c: (in function studSignUpWrite)
```