



# CS 2263 - FR01A

# Lab 5

By Ngoc Phuong Anh Nguyen - 3712361

November 2021

## Question 1:

```
1 #include <string.h>
2 #include <stdlib.h>
3 #include <stdio.h>
4 #include "Strings.h"
5 String mallocString(int stringsize)
6 {
7     return (String)malloc(sizeof(char)* (stringsize));
8 }
9 void freeString(String s)
10 {
11     free(s);
12 }
13 String duplicateString(String s)
14 {
15     String sCopy = mallocString(strlen(s));
16     if(sCopy != (String)NULL)
17     {
18         strcpy(sCopy,s);
19     }
20     return sCopy;
21 }
22 String* duplicateStringList(String* s,int n)
23 {
24     String* slCopy;
25     slCopy = (String*)malloc(sizeof(String)*n);
26     if(slCopy == (String*)NULL) return slCopy;
27     int i,j;
```

Figure 1: Source Code Of Strings.c

```
28     for(i = 0; i<n; i++)
29     {
30         slCopy[i] = duplicateString(s[i]);
31         if(slCopy[i] == (String)NULL)
32         {
33             for(j=0; j<i; j++)
34             {
35                 freeString(slCopy[j]);
36             }
37             free(slCopy);
38             slCopy = (String*)NULL;
39             break;
40         }
41     }
42     return slCopy;
43 }
44 int compareStrings(void* s1, void* s2)
45 {
46     return strcmp(s1,s2);
47 }
48 String getString(String terminators)
49 {
50     String s;
51     s = terminators;
52     return s;
53 }
54
```

Figure 2: Source Code of Strings.c

```

1  #include <string.h>
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include "Strings.h"
5
6  int main(int argc, char ** argv)
7  {
8      String a[5]= {"Hello World", "Hello Rick", "Hello You", "Hello Mom"};
9      int size = sizeof(a)/sizeof(a[0]);
10     a[4] = duplicateString(a[2]);
11     String* b = duplicateStringList(a,size);
12     int i;
13     for(i = 0; i < size; i++)
14     {
15         printf("a[%d]: %s\n",i,getString(a[i]));|
16         printf("b[%d]: %s\n",i,getString(*(b+i)));
17         if(compareStrings(a[i],*(b+i)) == 0)
18         {
19             printf("Same\n");
20         }
21         else
22         {
23             printf("Different\n");
24         }
25     }
26
27     return EXIT_SUCCESS;
28 }

```

Figure 3: Source Code of stringTest0.c

```

[anguyen5@gcl12m30 Lab 5]$ make stringTest0
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringTest0.c -o stringTest0
[anguyen5@gcl12m30 Lab 5]$ ./stringTest0
a[0]: Hello World
b[0]: Hello World
Same
a[1]: Hello Rick
b[1]: Hello Rick
Same
a[2]: Hello You
b[2]: Hello You
Same
a[3]: Hello Mom
b[3]: Hello Mom
Same
a[4]: Hello You
b[4]: Hello You
Same
[anguyen5@gcl12m30 Lab 5]$ █

```

Figure 4: Output of stringTest0.c

## Question 2:

**Without free() Function:**

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4
5  int main(int argc, char* argv[])
6  {
7      String programName;
8
9      programName = duplicateString(argv[0]);
10     if(programName == (String)NULL)
11     {
12         fprintf(stderr, "Memory failure, terminating");
13         return EXIT_FAILURE;
14     }
15     printf("%s\n", programName);
16
17     // free(programName);
18
19     return EXIT_SUCCESS;
20 }
```

Figure 5: Source Code of stringTest1.c

```
[anguyen5@gc112m30 Lab 5]$ make stringTest1
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringTest1.c -o stringTest1
[anguyen5@gc112m30 Lab 5]$ valgrind --tool=memcheck --leak-check=full --verbose --log-file=stringTest1-loga ./stringTest1
./stringTest1
[anguyen5@gc112m30 Lab 5]$
```

Figure 6: The output from running your Makefile, and valgrind command

```
[anguyen5@gc112m30 Lab 5]$ make stringTest1
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringTest1.c -o stringTest1
[anguyen5@gc112m30 Lab 5]$ ./stringTest1
./stringTest1
[anguyen5@gc112m30 Lab 5]$
```

Figure 7: Screenshot of the program running and output

```

1 ==12163== Memcheck, a memory error detector
2 ==12163== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
3 ==12163== Using Valgrind-3.15.0-608cbl1914-20190413 and LibVEX; rerun with -h for copyright info
4 ==12163== Command: ./stringTest1
5 ==12163== Parent PID: 29182
6 ==12163==
7 --12163--
8 --12163-- Valgrind options:
9 --12163-- --tool=memcheck
10 --12163-- --leak-check=full
11 --12163-- --verbose
12 --12163-- --log-file=stringTest1.log
13 --12163-- Contents of /proc/version:
14 --12163-- Linux version 3.10.0-1160.36.2.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC) ) #1 SMP Wed Jul 21 11:57:15 UTC 2021
15 --12163--
16 --12163-- Arch and hwcaps: AMD64, LittleEndian, amd64-cx16-lzcnt-rdtscp-sse3-ssse3-avx-avx2-bmi-f16c-rdrand
17 --12163-- Page sizes: currently 4096, max supported 4096
18 --12163-- Valgrind library directory: /usr/libexec/valgrind
19 --12163-- Reading syms from /home1/ugrads/anguyen5/CS 2263/Labs/Lab 5/stringTest1
20 --12163-- Reading syms from /usr/lib64/ld-2.17.so
21 --12163-- Reading syms from /usr/libexec/valgrind/memcheck-amd64-linux
22 --12163-- object doesn't have a symbol table
23 --12163-- object doesn't have a dynamic symbol table
24 --12163-- Scheduler: using generic scheduler lock implementation.
25 --12163-- Reading suppressions file: /usr/libexec/valgrind/default.supp
26 ==12163== embedded gdbserver: reading from /tmp/vgdb-pipe-from-vgdb-to-12163-by-anguyen5-on-gcll2m30.cs.unb.ca
27 ==12163== embedded gdbserver: writing to /tmp/vgdb-pipe-to-vgdb-from-12163-by-anguyen5-on-gcll2m30.cs.unb.ca
28 ==12163== embedded gdbserver: shared mem /tmp/vgdb-pipe-shared-mem-vgdb-12163-by-anguyen5-on-gcll2m30.cs.unb.ca
29 ==12163==
30 ==12163== TO CONTROL THIS PROCESS USING vgdb (which you probably
31 ==12163== don't want to do, unless you know exactly what you're doing,
32 ==12163== or are doing some strange experiment):
33 ==12163== /usr/libexec/valgrind/.../bin/vgdb --pid=12163 ...command...
34 ==12163==
35 ==12163== TO DEBUG THIS PROCESS USING GDB: start GDB like this
36 ==12163== /path/to/gdb ./stringTest1
37 ==12163== and then give GDB the following command
38 ==12163== target remote | /usr/libexec/valgrind/.../bin/vgdb --pid=12163
39 ==12163== --pid is optional if only one valgrind process is running
40 ==12163==
41 --12163-- REDIR: 0x4019e40 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c7ed5 (???
42 --12163-- REDIR: 0x4019c10 (ld-linux-x86-64.so.2:index) redirected to 0x580c7eef (???
43 --12163-- REDIR: 0x4019e40 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c7ed5 (???
44 --12163-- Reading syms from /usr/libexec/valgrind/vgpreload_core-amd64-linux.so
45 --12163-- Reading syms from /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so
46 ==12163== WARNING: new redirection conflicts with existing -- ignoring it
47 --12163-- old: 0x4019e40 (strlen) R-> (0000.0) 0x580c7ed5 ???
48 --12163-- new: 0x4019e40 (strlen) R-> (2007.0) 0x04c2d1b0 strlen
49 --12163-- REDIR: 0x4019dc0 (ld-linux-x86-64.so.2:strcmp) redirected to 0x4c2e300 (strcmp)
50 --12163-- REDIR: 0x401aa80 (ld-linux-x86-64.so.2:mempcpy) redirected to 0x4c31f90 (mempcpy)
51 --12163-- Reading syms from /usr/lib64/libc-2.17.so
52 --12163-- REDIR: 0x4ec71c0 (libc.so.6:strcasestr) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
53 --12163-- REDIR: 0x4ec3f40 (libc.so.6:strnlen) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
54 --12163-- REDIR: 0x4ec9490 (libc.so.6:strncasecmp) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
55 --12163-- REDIR: 0x4ec69a0 (libc.so.6:memset) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
56 --12163-- REDIR: 0x4ec6950 (libc.so.6:mempcpy@GLIBC 2.2.5) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
57 --12163-- REDIR: 0x4ec5930 (libc.so.6:_GI_strchr) redirected to 0x4c2cb70 (_GI_strchr)
58 --12163-- REDIR: 0x4ec3e10 (libc.so.6:strlen) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
59 --12163-- REDIR: 0x4fa67a0 (libc.so.6:_strnlen_sse2_pminub) redirected to 0x4c2d0f0 (strlen)
60 --12163-- REDIR: 0x4ebc740 (libc.so.6:malloc) redirected to 0x4c29eec (malloc)
61 --12163-- REDIR: 0x4ec3850 (libc.so.6:strcpy) redirected to 0x4a247a0 (_vgnU ifunc wrapper)
62 --12163-- REDIR: 0x4ed1770 (libc.so.6:_strcpy_sse2_unaligned) redirected to 0x4c2d1d0 (strcpy)
63 --12163-- REDIR: 0x4ec3e60 (libc.so.6:_GI_strlen) redirected to 0x4c2d110 (_GI_strlen)
64 --12163-- REDIR: 0x4ebcb60 (libc.so.6:free) redirected to 0x4c2afe0 (free)
65 ==12163==
66 ==12163== HEAP SUMMARY:
67 ==12163== in use at exit: 14 bytes in 1 blocks
68 ==12163== total heap usage: 1 allocs, 0 frees, 14 bytes allocated
69 ==12163==
70 ==12163== Searching for pointers to 1 not-freed blocks
71 ==12163== Checked 70,240 bytes
72 ==12163==
73 ==12163== 14 bytes in 1 blocks are definitely lost in loss record 1 of 1
74 ==12163== at 0x4C29F73: malloc (vg_replace_malloc.c:309)
75 ==12163== by 0x400747: mallocString (Strings.c:7)
76 ==12163== by 0x400782: duplicateString (Strings.c:15)
77 ==12163== by 0x4008F8: main (stringTest1.c:9)
78 ==12163==
79 ==12163== LEAK SUMMARY:
80 ==12163== definitely lost: 14 bytes in 1 blocks
81 ==12163== indirectly lost: 0 bytes in 0 blocks
82 ==12163== possibly lost: 0 bytes in 0 blocks
83 ==12163== still reachable: 0 bytes in 0 blocks
84 ==12163== suppressed: 0 bytes in 0 blocks
85 ==12163==
86 ==12163== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

```

Figure 8: The log-file output



## Within free() Function:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4
5  int main(int argc, char* argv[])
6  {
7      String programName;
8
9      programName = duplicateString(argv[0]);
10     if(programName == (String)NULL)
11     {
12         fprintf(stderr, "Memory failure, terminating");
13         return EXIT_FAILURE;
14     }
15     printf("%s\n", programName);
16
17     free(programName);
18
19     return EXIT_SUCCESS;
20 }
```

Figure 9: Source Code of stringTest1.c (within free() function)

```
[anguyen5@gcl12m30 Lab 5]$ make stringTest1
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringTest1.c -o stringTest1
[anguyen5@gcl12m30 Lab 5]$ valgrind --tool=memcheck --leak-check=full --verbose --log-file=stringTest1-logb ./stringTest1
./stringTest1
[anguyen5@gcl12m30 Lab 5]$
```

Figure 10: The output from running your Makefile, and valgrind command

```
[anguyen5@gcl12m30 Lab 5]$ make stringTest1
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringTest1.c -o stringTest1
[anguyen5@gcl12m30 Lab 5]$ ./stringTest1
./stringTest1
[anguyen5@gcl12m30 Lab 5]$
```

Figure 11: Screenshot of the program running and output

```

1 ==11762== Memcheck, a memory error detector
2 ==11762== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
3 ==11762== Using Valgrind-3.15.0-608cb11914-20190413 and LibVEX; rerun with -h for copyright info
4 ==11762== Command: ./stringTest1
5 ==11762== Parent PID: 29182
6 ==11762==
7 --11762--
8 --11762-- Valgrind options:
9 --11762--   --tool=memcheck
10 --11762--   --leak-check=full
11 --11762--   --verbose
12 --11762--   --log-file=stringTest1-log
13 --11762-- Contents of /proc/version:
14 --11762--   Linux version 3.10.0-1160.36.2.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC) ) #1 SMP Wed Jul 2
15 --11762--
16 --11762-- Arch and hwcaps: AMD64, LittleEndian, amd64-cx16-lzcnt-rdtscp-sse3-ssse3-avx-avx2-bmi-f16c-rdrand
17 --11762-- Page sizes: currently 4096, max supported 4096
18 --11762-- Valgrind library directory: /usr/libexec/valgrind
19 --11762-- Reading syms from /home1/ugrads/anguyen5/CS 2263/Labs/Lab 5/stringTest1
20 --11762-- Reading syms from /usr/lib64/ld-2.17.so
21 --11762-- Reading syms from /usr/libexec/valgrind/memcheck-amd64-linux
22 --11762--   object doesn't have a symbol table
23 --11762--   object doesn't have a dynamic symbol table
24 --11762-- Scheduler: using generic scheduler lock implementation.
25 --11762-- Reading suppressions file: /usr/libexec/valgrind/default.supp
26 ==11762== embedded gdbserver: reading from /tmp/vgdb-pipe-from-vgdb-to-11762-by-anguyen5-on-gc112m30.cs.unb.ca
27 ==11762== embedded gdbserver: writing to /tmp/vgdb-pipe-to-vgdb-from-11762-by-anguyen5-on-gc112m30.cs.unb.ca
28 ==11762== embedded gdbserver: shared mem /tmp/vgdb-pipe-shared-mem-vgdb-11762-by-anguyen5-on-gc112m30.cs.unb.ca
29 ==11762==
30 ==11762== TO CONTROL THIS PROCESS USING vgdb (which you probably
31 ==11762== don't want to do, unless you know exactly what you're doing,
32 ==11762== or are doing some strange experiment):
33 ==11762== /usr/libexec/valgrind/../../bin/vgdb --pid=11762 ...command...
34 ==11762==
35 ==11762== TO DEBUG THIS PROCESS USING GDB: start GDB like this
36 ==11762== /path/to/gdb ./stringTest1
37 ==11762== and then give GDB the following command
38 ==11762== target remote | /usr/libexec/valgrind/../../bin/vgdb --pid=11762
39 ==11762== --pid is optional if only one valgrind process is running
40 ==11762==
41 --11762-- REDIR: 0x4019e40 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c7ed5 (???)
42 --11762-- REDIR: 0x4019c10 (ld-linux-x86-64.so.2:index) redirected to 0x580c7eef (???)
43 --11762-- Reading syms from /usr/libexec/valgrind/vgpreload_core-amd64-linux.so
44 --11762-- Reading syms from /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so
45 ==11762== WARNING: new redirection conflicts with existing -- ignoring it
46 --11762--   old: 0x04019e40 (strlen) R-> (0000.0) 0x580c7ed5 ???
47 --11762--   new: 0x04019e40 (strlen) R-> (2007.0) 0x04c2d1b0 strlen
48 --11762-- REDIR: 0x4019dc0 (ld-linux-x86-64.so.2:strncmp) redirected to 0x4c2e300 (strncmp)
49 --11762-- REDIR: 0x401aa80 (ld-linux-x86-64.so.2:memcpy) redirected to 0x4c31f90 (memcpy)
50 --11762-- Reading syms from /usr/lib64/libc-2.17.so
51 --11762-- REDIR: 0x4ec71c0 (libc.so.6:strcasestr) redirected to 0x4a247a0 (vgnU ifunc wrapper)
52 --11762-- REDIR: 0x4ec3f40 (libc.so.6:strnlen) redirected to 0x4a247a0 (vgnU ifunc wrapper)
53 --11762-- REDIR: 0x4ec9490 (libc.so.6:strncasestr) redirected to 0x4a247a0 (vgnU ifunc wrapper)
54 --11762-- REDIR: 0x4ec69a0 (libc.so.6:memset) redirected to 0x4a247a0 (vgnU ifunc wrapper)
55 --11762-- REDIR: 0x4ec6950 (libc.so.6:memcpy@GLIBC 2.2.5) redirected to 0x4a247a0 (vgnU ifunc wrapper)
56 --11762-- REDIR: 0x4ec5930 (libc.so.6:_GI_strrchr) redirected to 0x4c2cb70 (_GI_strrchr)
57 --11762-- REDIR: 0x4ec3e10 (libc.so.6:strlen) redirected to 0x4a247a0 (vgnU ifunc wrapper)
58 --11762-- REDIR: 0x4fa67a0 (libc.so.6:strlen_sse2_pminub) redirected to 0x4c2d0f0 (strlen)
59 --11762-- REDIR: 0x4ebc740 (libc.so.6:malloc) redirected to 0x4c29eec (malloc)
60 --11762-- REDIR: 0x4ec3850 (libc.so.6:strcpy) redirected to 0x4a247a0 (vgnU ifunc wrapper)
61 --11762-- REDIR: 0x4ed1770 (libc.so.6:strcpy_sse2_unaligned) redirected to 0x4c2d1d0 (strcpy)
62 --11762-- REDIR: 0x4ec3e60 (libc.so.6:_GI_strlen) redirected to 0x4c2d110 (_GI_strlen)
63 --11762-- REDIR: 0x4ebcb60 (libc.so.6:free) redirected to 0x4c2afe6 (free)
64 ==11762==
65 ==11762== HEAP SUMMARY:
66 ==11762==   in use at exit: 0 bytes in 0 blocks
67 ==11762== total heap usage: 1 allocs, 1 frees, 14 bytes allocated
68 ==11762==
69 ==11762== All heap blocks were freed -- no leaks are possible
70 ==11762==
71 ==11762== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Figure 12: The log-file output

### Question 3:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "Strings.h"
4
5  int main(int argc, char* argv[])
6  {
7      String a[5];
8
9      int size = 5;
10     int i;
11     printf("Input string: \n");
12     for(i = 0; i < size; i++)
13     {
14         a[i] = mallocString(100);
15         printf("a[%d]: \n", i);
16         scanf("%99s", a[i]);
17     }
18
19     String* b = duplicateStringList(a, size);
20     |
21     for(i = 0; i < size; i++)
22     {
23         printf("a[%d]: %s\n", i, getString(a[i]));
24         printf("b[%d]: %s\n", i, getString(*(b+i)));
25         free(a[i]);
26     }
27
28     // a[4] = duplicateString(a[2]);
29     // String* b = duplicateStringList(a, size);
30
31     return EXIT_SUCCESS;
32 }
33
```



```
[anguyen5@gc112m30 Lab 5]$ make stringListTest
gcc -g -Wall -Wshadow -c Strings.c
gcc -g -Wall -Wshadow Strings.o stringListTest.c -o stringListTest
[anguyen5@gc112m30 Lab 5]$ valgrind --tool=memcheck --leak-check=full --verbose --log-file=stringListTest-log ./stringListTest
Input string:
a[0]:
Rick
a[1]:
Amy
a[2]:
Anna
a[3]:
Hen
a[4]:
Hal
a[0]: Rick
b[0]: Rick
a[1]: Amy
b[1]: Amy
a[2]: Anna
b[2]: Anna
a[3]: Hen
b[3]: Hen
a[4]: Hal
b[4]: Hal
[anguyen5@gc112m30 Lab 5]$
```

```
==13242== Memcheck, a memory error detector
==13242== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==13242== Using Valgrind-3.15.0-608cbl1914-20190413 and LibVEX; rerun with -h for copyright info
==13242== Command: ./stringListTest
==13242== Parent PID: 29182
==13242==
--13242--
--13242-- Valgrind options:
--13242--   --tool=memcheck
--13242--   --leak-check=full
--13242--   --verbose
--13242--   --log-file=stringListTest-log
--13242-- Contents of /proc/version:
--13242--   Linux version 3.10.0-1160.36.2.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC) ) #1 SMP Wed Jul 2:
--13242--
--13242-- Arch and hwcaps: AMD64, LittleEndian, amd64-cx16-lzcnt-rdtscp-sse3-ssse3-avx-avx2-bmi-f16c-rdrand
--13242-- Page sizes: currently 4096, max supported 4096
--13242-- Valgrind library directory: /usr/libexec/valgrind
--13242-- Reading syms from /home1/ugrads/anguyen5/CS 2263/Labs/Lab 5/stringListTest
--13242-- Reading syms from /usr/lib64/ld-2.17.so
--13242-- Reading syms from /usr/libexec/valgrind/memcheck-amd64-linux
--13242--   object doesn't have a symbol table
--13242--   object doesn't have a dynamic symbol table
--13242-- Scheduler: using generic scheduler lock implementation.
--13242-- Reading suppressions file: /usr/libexec/valgrind/default.supp
==13242== embedded gdbserver: reading from /tmp/vgdb-pipe-from-vgdb-to-13242-by-anguyen5-on-gc112m30.cs.unb.ca
==13242== embedded gdbserver: writing to /tmp/vgdb-pipe-to-vgdb-from-13242-by-anguyen5-on-gc112m30.cs.unb.ca
==13242== embedded gdbserver: shared mem /tmp/vgdb-pipe-shared-mem-vgdb-13242-by-anguyen5-on-gc112m30.cs.unb.ca
==13242==
==13242== TO CONTROL THIS PROCESS USING vgdb (which you probably
==13242== don't want to do, unless you know exactly what you're doing,
==13242== or are doing some strange experiment):
==13242==   /usr/libexec/valgrind/../../bin/vgdb --pid=13242 ...command...
==13242==
==13242== TO DEBUG THIS PROCESS USING GDB: start GDB like this
```

```

36 ==13242== /path/to/gdb ./stringListTest
37 ==13242== and then give GDB the following command
38 ==13242== target remote | /usr/libexec/valgrind/../../bin/vgdb --pid=13242
39 ==13242== --pid is optional if only one valgrind process is running
40 ==13242==
41 --13242-- REDIR: 0x4019e40 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c7ed5 (???)
42 --13242-- REDIR: 0x4019c10 (ld-linux-x86-64.so.2:index) redirected to 0x580c7eef (???)
43 --13242-- Reading syms from /usr/libexec/valgrind/vgpreload_core-amd64-linux.so
44 --13242-- Reading syms from /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so
45 ==13242== WARNING: new redirection conflicts with existing -- ignoring it
46 --13242-- old: 0x04019e40 (strlen) R-> (0000.0) 0x580c7ed5 ???
47 --13242-- new: 0x04019e40 (strlen) R-> (2007.0) 0x04c2d1b0 strlen
48 --13242-- REDIR: 0x4019dc0 (ld-linux-x86-64.so.2:strcmp) redirected to 0x4c2e300 (strcmp)
49 --13242-- REDIR: 0x401aa80 (ld-linux-x86-64.so.2:mempcpy) redirected to 0x4c31f90 (mempcpy)
50 --13242-- Reading syms from /usr/lib64/libc-2.17.so
51 --13242-- REDIR: 0x4ec71c0 (libc.so.6:strcasecmp) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
52 --13242-- REDIR: 0x4ec3f40 (libc.so.6:strlen) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
53 --13242-- REDIR: 0x4ec9490 (libc.so.6:strncasecmp) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
54 --13242-- REDIR: 0x4ec69a0 (libc.so.6:memset) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
55 --13242-- REDIR: 0x4ec6950 (libc.so.6:mempcpy@GLIBC 2.2.5) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
56 --13242-- REDIR: 0x4ec5930 (libc.so.6: _GI strrchr) redirected to 0x4c2cb70 ( _GI strrchr)
57 --13242-- REDIR: 0x4ec3e60 (libc.so.6: _GI strlen) redirected to 0x4c2d110 ( _GI strlen)
58 --13242-- REDIR: 0x4ebc740 (libc.so.6:malloc) redirected to 0x4c29eec (malloc)
59 --13242-- REDIR: 0x4ecd450 (libc.so.6:strchrnul) redirected to 0x4c31ab0 (strchrnul)
60 --13242-- REDIR: 0x4ec6b70 (libc.so.6: _GI mempcpy) redirected to 0x4c31cc0 ( _GI mempcpy)
61 --13242-- REDIR: 0x4ebcb60 (libc.so.6:free) redirected to 0x4c2afe6 (free)
62 --13242-- REDIR: 0x4ec3e10 (libc.so.6:strlen) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
63 --13242-- REDIR: 0x4fa67a0 (libc.so.6: strlen sse2 pminub) redirected to 0x4c2d0f0 (strlen)
64 --13242-- REDIR: 0x4ec3850 (libc.so.6:strcpy) redirected to 0x4a247a0 ( _vgnU ifunc wrapper)
65 --13242-- REDIR: 0x4ed1770 (libc.so.6: strcpy sse2_unaligned) redirected to 0x4c2d1d0 (strcpy)
66 ==13242==
67 ==13242== HEAP SUMMARY:
68 ==13242== in use at exit: 40 bytes in 1 blocks
69 ==13242== total heap usage: 11 allocs, 10 frees, 557 bytes allocated
70 ==13242==
71 ==13242== Searching for pointers to 1 not-freed blocks
72 ==13242== Checked 74,368 bytes
73 ==13242==
74 ==13242== 40 bytes in 1 blocks are definitely lost in loss record 1 of 1
75 ==13242== at 0x4C29F73: malloc (vg_replace_malloc.c:309)
76 ==13242== by 0x4007E7: duplicateStringList (Strings.c:25)
77 ==13242== by 0x400985: main (stringListTest.c:19)
78 ==13242==
79 ==13242== LEAK SUMMARY:
80 ==13242== definitely lost: 40 bytes in 1 blocks
81 ==13242== indirectly lost: 0 bytes in 0 blocks
82 ==13242== possibly lost: 0 bytes in 0 blocks
83 ==13242== still reachable: 0 bytes in 0 blocks
84 ==13242== suppressed: 0 bytes in 0 blocks
85 ==13242==
86 ==13242== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
87

```

Figure 13: The log-file output