

```

Compiling mentry.c
mentry.c: In function me_get:
mentry.c:18:2: warning: implicit declaration of function malloc [-Wimplicit-func
tion-declaration]
    entry = (MEntry *) malloc(sizeof(MEntry));
    ^
mentry.c:18:21: warning: incompatible implicit declaration of built-in function mal
loc
    entry = (MEntry *) malloc(sizeof(MEntry));
                        ^
mentry.c:24:3: warning: implicit declaration of function free [-Wimplicit-functi
on-declaration]
    free(entry);
    ^
mentry.c:24:3: warning: incompatible implicit declaration of built-in function free
mentry.c:34:2: warning: implicit declaration of function strncpy [-Wimplicit-fun
ction-declaration]
    strncpy(entry->full_address, string, 350);
    ^
mentry.c:34:2: warning: incompatible implicit declaration of built-in function strn
cpy
mentry.c:35:2: warning: implicit declaration of function strtok [-Wimplicit-func
tion-declaration]
    strncpy(entry->surname, strtok(string, ","), 400);
    ^
mentry.c:35:26: warning: passing argument 2 of strncpy makes pointer from intege
r without a cast
    strncpy(entry->surname, strtok(string, ","), 400);
                        ^
mentry.c:35:26: note: expected const char * but argument is of type int
mentry.c:41:2: warning: implicit declaration of function strcat [-Wimplicit-func
tion-declaration]
    strcat(entry->full_address, string2);
    ^
mentry.c:41:2: warning: incompatible implicit declaration of built-in function strc
at
mentry.c:56:2: warning: implicit declaration of function strpbrk [-Wimplicit-fun
ction-declaration]
    zip = strpbrk(string3, num);
    ^
mentry.c:56:8: warning: incompatible implicit declaration of built-in function strp
brk
    zip = strpbrk(string3, num);
           ^
mentry.c:58:17: warning: assignment makes pointer from integer without a cast
    entry->zipcode = strtok(entry->zipcode, "\n");
                  ^
mentry.c: In function me_hash:
mentry.c:68:2: warning: implicit declaration of function strdup [-Wimplicit-func
tion-declaration]
    t = strdup(me->surname);
    ^
mentry.c:68:6: warning: incompatible implicit declaration of built-in function strd
up
    t = strdup(me->surname);
        ^
mentry.c:69:2: warning: incompatible implicit declaration of built-in function free
mentry.c:69:2: warning: incompatible implicit declaration of built-in function free
    free(t); //strdup returns malloced pointer, must free to avoid leak
    ^
mentry.c:77:2: warning: implicit declaration of function atoi [-Wimplicit-functi
on-declaration]
    hash += atoi(me->zipcode);

```

```

^
mentry.c: In function 'me_compare':
mentry.c:93:2: warning: implicit declaration of function 'strcasecmp' [-Wimplicit-
function-declaration]
    int result = strcasecmp(mel->surname, me2->surname);
    ^
mentry.c:99:3: warning: implicit declaration of function 'strcmp' [-Wimplicit-func
tion-declaration]
    result = strcmp(mel->zipcode, me2->zipcode);
    ^
mentry.c: In function 'me_destroy':
mentry.c:106:2: warning: incompatible implicit declaration of built-in function 'fre
e'
    free(me->zipcode);
    ^

```

Linking mecat

```
gcc -g -o mecat mecat.o mentry.o
```

Checking to see if me_get and me_print work correctly on 1000 entry file

```
./mecat <tento3.txt | diff - tento3.txt | cat -A | head -20
```

```
0a1,3000$
```

```

> Kennedy, Mr Q$
> 29 Quince Ln$
> Junction City, OR 97256$
> Jackson, Mr D$
> 51 Oak Ln$
> Junction City, OR 97393$
> Williams, Mr H$
> 32 Maple St$
> Coos Bay, OR 97868$
> Jones, Miss Z$
> 92 Apple Blvd$
> Corvallis, OR 97427$
> Smith, Ms B$
> 20 Orange Ln$
> Albany, OR 97484$
> Lincoln, Mrs O$
> 36 Maple Ln$
> Salem, OR 97397$
> Lincoln, Mr C$

```

Checking to see if me_get and me_print work correctly on 1000000 entry file

```
./mecat <tento6.txt | diff - tento6.txt | cat -A | head -20
```

```
0a1,3000000$
```

```

> Kennedy, Mr Q$
> 29 Quince Ln$
> Junction City, OR 97256$
> Jackson, Mr D$
> 51 Oak Ln$
> Junction City, OR 97393$
> Williams, Mr H$
> 32 Maple St$
> Coos Bay, OR 97868$
> Jones, Miss Z$
> 92 Apple Blvd$
> Corvallis, OR 97427$
> Smith, Ms B$
> 20 Orange Ln$
> Albany, OR 97484$
> Lincoln, Mrs O$
> 36 Maple Ln$
> Salem, OR 97397$
> Lincoln, Mr C$

```

Checking for memory leaks in mentry

```
valgrind ./mecat <tento4.txt >/dev/null
```

```
==17288== Memcheck, a memory error detector
```

```

==17288== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==17288== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==17288== Command: ./mecat
==17288==
==17288== Invalid read of size 1
==17288==    at 0x4C2DB30: __strncpy_sse2_unaligned (in /usr/lib/valgrind/vgpreload_me
mcheck-amd64-linux.so)
==17288==    by 0x400B18: me_get (mentry.c:35)
==17288==    by 0x400A0D: main (mecat.c:17)
==17288== Address 0xfffffffff000320 is not stack'd, malloc'd or (recently) free'd
==17288==
==17288== Process terminating with default action of signal 11 (SIGSEGV)
==17288== Access not within mapped region at address 0xFFFFFFFFf000320
==17288==    at 0x4C2DB30: __strncpy_sse2_unaligned (in /usr/lib/valgrind/vgpreload_me
mcheck-amd64-linux.so)
==17288==    by 0x400B18: me_get (mentry.c:35)
==17288==    by 0x400A0D: main (mecat.c:17)
==17288== If you believe this happened as a result of a stack
==17288== overflow in your program's main thread (unlikely but
==17288== possible), you can try to increase the size of the
==17288== main thread stack using the --main-stacksize= flag.
==17288== The main thread stack size used in this run was 8388608.
==17288==
==17288== HEAP SUMMARY:
==17288==    in use at exit: 1,282 bytes in 4 blocks
==17288==    total heap usage: 4 allocs, 0 frees, 1,282 bytes allocated
==17288==
==17288== LEAK SUMMARY:
==17288==    definitely lost: 0 bytes in 0 blocks
==17288==    indirectly lost: 0 bytes in 0 blocks
==17288==    possibly lost: 0 bytes in 0 blocks
==17288==    still reachable: 1,282 bytes in 4 blocks
==17288==    suppressed: 0 bytes in 0 blocks
==17288== Rerun with --leak-check=full to see details of leaked memory
==17288==
==17288== For counts of detected and suppressed errors, rerun with: -v
==17288== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
./test88.sh: line 29: 17288 Segmentation fault      (core dumped) valgrind ./mecat < t
ento4.txt > /dev/null
Linking findduplLL
gcc -g -o findduplLL finddupl.o mentry.o mlistLL.o
Checking to see if findduplLL works correctly on S.txt
./findduplLL <S.txt | diff - S.out | cat -A | head -20
0a1,20$
> Potential duplicate$
> =====$
> Rogers, Fred$
> 123 Neighbor Place$
> Latrobe, PA 15650$
> =====$
> Rogers, Mr$
> 123 Neighbor Drive$
> Latrobe, PA 15650$
> $
> Potential duplicate$
> =====$
> Sventek, Joe$
> Computer and Information Science Department$
> Eugene, OR 97403$
> =====$
> Sventek, Prof J$
> University of Oregon$
> Eugene, OR 97403$

```

```

Checking to see if findduplLL works correctly on M.txt
./findduplLL <M.txt | diff - M.out | cat -A | head -20
0a1,510$
> Potential duplicate$
> =====$
> Johnson, Ms I$
> 34 Fir Ln$
> Salem, OR 97620$
> =====$
> Johnson, Mrs Z$
> 34 Orange St$
> Coos Bay, OR 97620$
> $
> Potential duplicate$
> =====$
> Adams, Mr H$
> 61 Pine Dr$
> Salem, OR 97232$
> =====$
> Adams, Dr Z$
> 61 Maple Blvd$
> Eugene, OR 97232$
Compiling mlist.c
mlist.c: In function 'ml_create':
mlist.c:26:6: warning: unused variable 'i' [-Wunused-variable]
    int i;
        ^
mlist.c: In function 'ml_add':
mlist.c:43:6: warning: unused variable 'prev_size' [-Wunused-variable]
    int prev_size;
        ^
mlist.c:40:17: warning: unused variable 'm' [-Wunused-variable]
    MListNode *q, *m;
                  ^
mlist.c:39:13: warning: variable 'r' set but not used [-Wunused-but-set-variable]
    MList *p, *r;
                ^
mlist.c: In function 'ml_destroy':
mlist.c:112:16: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
    for (i = 0; i < ml->list_size; i++) {
                  ^
Linking finddupl
gcc -g -o finddupl finddupl.o mentryTrue.o mlist.o
Checking to see if finddupl works correctly on S.txt
./finddupl <S.txt | diff - S.out | head -20
Checking to see if finddupl works correctly on M.txt
./finddupl <M.txt | diff - M.out | head -20
Checking to see if finddupl works correctly on 10,000 entry file
./finddupl <tento4.txt | diff - tento4.out | head -20
Checking to see if finddupl works correctly on 100,000 entry file
./finddupl <tento5.txt | diff - tento5.out | head -20
Checking to see if finddupl works correctly on 1,000,000 entry file
./finddupl <tento6.txt | diff - tento6.out | head -20
Checking for memory leaks
valgrind ./finddupl <tento5.txt >/dev/null
==17327== Memcheck, a memory error detector
==17327== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==17327== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==17327== Command: ./finddupl
==17327==
==17327== Conditional jump or move depends on uninitialised value(s)
==17327==    at 0x401308: ml_lookup (mlist.c:98)
==17327==    by 0x400B41: main (finddupl.c:28)

```

```
==17327==
==17327== Conditional jump or move depends on uninitialised value(s)
==17327==    at 0x401308: ml_lookup (mlist.c:98)
==17327==    by 0x4011BC: ml_add (mlist.c:45)
==17327==    by 0x400B5F: main (finddupl.c:30)
==17327==
==17327== Conditional jump or move depends on uninitialised value(s)
==17327==    at 0x401230: ml_add (mlist.c:55)
==17327==    by 0x400B5F: main (finddupl.c:30)
==17327==
==17327== Conditional jump or move depends on uninitialised value(s)
==17327==    at 0x401384: ml_destroy (mlist.c:114)
==17327==    by 0x400BE8: main (finddupl.c:41)
==17327==
==17327==
==17327== HEAP SUMMARY:
==17327==    in use at exit: 0 bytes in 0 blocks
==17327==    total heap usage: 496,430 allocs, 496,430 frees, 11,829,008 bytes allocated
==17327==
==17327== All heap blocks were freed -- no leaks are possible
==17327==
==17327== For counts of detected and suppressed errors, rerun with: -v
==17327== Use --track-origins=yes to see where uninitialised values come from
==17327== ERROR SUMMARY: 203616 errors from 4 contexts (suppressed: 0 from 0)
Finished.
```