**Lab 07: Memory Debug Lab**

**Practice 1: Finding Memory Bugs**

1. Are there any memory errors in the following programs? If so, list all of them. Assume that the user enters in correct input, and that the sizes entered are at least one.   
  
Write your solution in a text or Word file and submit it.

void main() {

char \*str, \*input;

int \*ilist;

int i, size1, size2;

printf("Number of letters in word: ");

scanf("%d", &size1); /\* user inputs an integer \*/

printf("Number of integers: ");

scanf("%d", &size2); /\* user inputs an integer \*/

str = (char \*) malloc(size1);

ilist = (int \*) malloc(size2);

printf("Word: ");

scanf("%s", str); /\* user inputs a string \*/

for(i = 0; i < size2; i++) {

printf("Number %d of %d: ", i + 1, size2);

scanf("%d", ilist + i); /\* user inputs an integer \*/

}

}

2. Are there any memory errors in the following program? If so, list all of them.   
  
Write your solution in a text or Word file and submit it below.   
  
/\* return 1 if str is "1", 0 otherwise \*/  
int checkIf1(char \*str) {

char \*newstr = malloc(strlen(str) + 1);

strcpy(newstr, str); /\* set newstr to str \*/

if (strcmp(newstr, "1") == 0) { /\* newstr is "1" \*/

return 1;

}

free(newstr);

return 0;

}  
  
void main() {

char \*strArr[4] = {"1", "2", "3", "4"};

int i;

for(i = 0; i < 4; i++) {

printf("%d\n", checkIf1(strArr[i]));

}

}  
  
3. Are there any memory errors in the following program? If so, list all of them.   
  
Write your solution in a text or Word file and submit it below.   
  
struct data {

char \*str1, \*str2;

};   
  
/\* returns two strings concatenated if they are not the same, NULL otherwise \*/  
char \*mergeSingleIfDifferent(char \*s1, char \*s2) {

char \*str = (char \*) malloc(strlen(s1) + strlen(s2) + 1);  
if (strcmp(s1, s2) == 0) { /\* strings are equal \*/

str = NULL;

}  
else {

strcpy(str, s1);

strcat(str, s2);

}  
return str;

}  
  
/\* copies merged strings (or NULL) into array of strings passed in (results) \*/  
void mergeArrayIfDifferent(char \*results[], char \*strA1[], char \*strA2[], int size) {

int i;  
  
for(i = 0; i < size; i++) {

results[i] = mergeSingleIfDifferent(strA1[i], strA2[i]);

}

}

void printAndFree(int c, char \*str) {

if (str != NULL) {

printf("%d: %s\n", c, str);  
free(str);

}

}  
  
void main() {

char \*strArr1[8] = {"1", "2", "3", "4", "5", "6", "7", "8"};  
char \*strArr2[8] = {"a", "2", "c", "4", "e", "6", "g", "8"};  
char \*results[8];  
int i;  
  
mergeArrayIfDifferent(results, strArr1, strArr2, 8);  
for(i = 0; i < 8; i++) {

printAndFree(i, results);

}

}

**Practice 2: Read the code in** [**sizeof.rar**](http://swjx.scu.edu.cn/moodle/file.php/49/systemlevelprogramming/week7/sizeof.rar)**. Without running it, analysis it ，then verify it with your computer.**

**此题答案无需提交**