CST8234 – C Programming
Sample Questions Midterm II

Name:_____

Section I **[35 Points]**

Write a small C-like function **`revstr( )`** that reverses a string **`s`** in the same string.

Function Prototype:

```
void revstr( char * s );
```

Example:

```
   Memory Position          String
+----------------------------------------+
| s [ 0x7fffc22b8550 ] --> Hello World    |
| s [ 0x7fffc22b8550 ] --> dlroW olleH    |
+----------------------------------------+
```

`int strlen( char * s )` from the library `string.h` returns to you the length of the string **`s`**.

To obtain all the points in this question, you need to use the same function prototype given above.

[10 Points]

CST8234 – C Programming
Sample Questions Midterm II

Write a small C-like function **`reverseFile( )`** that reverses each line of a file, and writes the new file into a new file. Your function should return 0 if it was able to reverse the file or **`EXIT_FAILURE`** if it encountered any problems.

Function Prototype:

```
int reverseFile( char * filein, char *fileout );
```

Pre-Conditions:

`filein`: It is an existing file
`fileout`: A file may exist or not

Post-Conditions:

`filein`: It is an existing file
`fileout`: It is a duplicate of filein, but each line is has been reversed. `Filein` and `fileout` have the same size. The function returns 0 if it was able to reverse the file or `EXIT_FAILURE` if not

Example:

```
root@luna:/13W_CST8234# more
test
This is simple example
of how to reverse a complete
line in a file
```

```
root@luna:/13W_CST8234# more
output
elpmaxe elpmis si sihT
etelpmoc a esrever ot woh fo
elif a ni enil
```

[10 Points]

CST8234 – C Programming
Sample Questions Midterm II

Assume you have a function **`display_usage( )`** in a file **`usage.c`** with the following function prototype in the file **`usage.h`**:

```
/*********************************************************************
 * displays in stderr a usage message for a program prog that needs
 * arguments arg
 *********************************************************************/
void display_usage( const char *prog, cons char *arg );
```

You **DO NOT** need to write this function.

Write a complete program that receives from the command line argument two arguments `filea`, `fileb`. Your program should check that the correct number of arguments are passed and display the appropriate usage message if not. If you have the right number of arguments, your program should reverse `filea` into `fileb`.

[5 Points]

CST8234 – C Programming
Sample Questions Midterm II

Assuming that you wrote your **`main()`** function and your **`reverseFile()`** and your **`revstr()`** in a file called **`main.c`**, and that you use the **`display_usage()`** function from the file **`usage.c`**, how would you manually compile your program to create an executable called **`reverse`**. You want your code to be ANSI C.

[4 Points]

Jared finished to write his program and his Makefile. After giving the **`make`** command, he got the following error:

```
root@luna:/13W_CST8234# make
tmp/ccU7gSMg.o: In function `reverseFile':
ReverseFile.c:(.text+0x195): undefined reference to `revsr'
collect2: ld returned 1 exit status
```

Please notice that some information has been omitted in the above `make` command.

a) What does the error means,
(b) from which stage of the compilation process is coming and
(c) what can you do to fix it.

[6 Points]

CST8234 – C Programming
Sample Questions Midterm II

Section II **[20 Points]**

Giving the following data structure:

```
struct node {
      int data;
      struct node * next;
}
```

Write a small C-like function `insertN()` which will insert a new node at any index within a list. The caller may specify any index in the range [0..n], and the new node should be inserted so as to be at that index.

Function prototype:

```
void insertN(struct node** headRef, int index, int data);
```

Example

```
root@luna:/13W_CST8234# ./insertN
[ HEAD ]-->[ 0 ]-->[ 15 ]-->[ 10 ]-->[ 5 ]-->[ NULL ]
insertN( &head, 3, -44 )
[ HEAD ]-->[ 0 ]-->[ 15 ]-->[ 10 ]-->[ -44 ]-->[ 5 ]-->[ NULL ]
insertN( &head, 40, -55 )
[ HEAD ]-->[ 0 ]-->[ 15 ]-->[ 10 ]-->[ -44 ]-->[ 5 ]-->[ NULL ]
insertN( &head, 0, -66 )
[ HEAD ]-->[ -66 ]-->[ 0 ]-->[ 15 ]-->[ 10 ]-->[ -44 ]-->[ 5 ]-->[ NULL ]
```

Be sure to carefully test your boundary conditions.

[10 Points]

CST8234 – C Programming
Sample Questions Midterm II

Write a small C-like function `removeN()` which will removes a node at any index within a list. The caller may specify any index in the range [0..n], and the node should be removed.

Function prototype:

```
void removeN(struct node** headRef, int index);
```

Example:

```
[ HEAD ]-->[ -66 ]-->[ 0 ]-->[ 15 ]-->[ 10 ]-->[ -44 ]-->[ 5 ]-->[ NULL ]
removeN( &head, 2 )
[ HEAD ]-->[ -66 ]-->[ 0 ]-->[ 10 ]-->[ -44 ]-->[ 5 ]-->[ NULL ]
removeN( &head, 3 )
[ HEAD ]-->[ -66 ]-->[ 0 ]-->[ 10 ]-->[ 5 ]-->[ NULL ]
removeN( &head, 30 )
[ HEAD ]-->[ -66 ]-->[ 0 ]-->[ 10 ]-->[ 5 ]-->[ NULL ]
removeN( &head, 0 )
[ HEAD ]-->[ 0 ]-->[ 10 ]-->[ 5 ]-->[ NULL ]
```

Be sure to carefully test your boundary conditions.
[10 Points]

CST8234 – C Programming
Sample Questions Midterm II

Section II **[15 Points]**

Giving the following data structure:

```
struct node {
      int data;
      struct node * next;
}
```

Write a small C-like function `Merge()` which merges two list nodes together to make one list, taking nodes alternately between the two lists. If either list runs out, all the nodes should be taken from the other list.

Function prototype:

```
struct node * Merge( struct node *a, struct node *b );
```

Example:

a = { 78 -> 10 -> 6 -> 3 -> NULL } **b** = { 44 -> 34 -> 10 -> NULL }

c = { 78 -> 44 -> 10 -> 34 -> 6 -> 10 -> 3 -> NULL }

[15 Points]

CST8234 – C Programming
Sample Questions Midterm II

Section III **[15 Points]**

You are to write a small C-like function countChars() that counts how many times each letter from 'A' to 'Z' occurs in a text file.

Function prototype:

```
void countChars( const char *file, int *count );
```

Pre-Conditions:

`file`	Text file, it may or may not exits
`count`	An int array of 26 positions, each position representing a letter from A to Z count memory has already been allocated

Post-Conditions:

`count`	count[0] contains the number of 'A's in the file count[1] contains the number of 'B's in the file

[12 Points]