# Programming Problems

Notes to Applicant:

All coding to be written in C, compiled using gcc. If you don’t have gcc readily available, there are online IDEs that could be used to compile and execute C programs (such as <https://www.codechef.com/ide> ).

Please set the language to C.

For each of the problems listed please provide in full the detailed description of the approach you took and the related steps to addressing the problem statements. In addition, please describe the test scenarios you used to verify your solutions. Assume that the code will execute in a real-time embedded system environment.

Submit the source code, including the syntax of the compile command used, as well as the written solution details for each problem. Please ensure you provide all work products in a zipped file archive. Thank you.

# Programming Problem 1:

You are given a set of measurements over time of water levels in a tank.  Over time, the water levels fluctuate and a measurement of the level is taken every 5 minutes.  Write a function in C that, given an array of water level values and a threshold level (as integer values), will determine:

* the number of times the water has crossed the threshold (from below the threshold to above the threshold), and
* the amount of water that has exceeded the threshold.

The function should have the following signature:

int checkLevel( const int levels[],const int numSamples, int threshold,  unsigned int \*numOfCrossings, unsigned int \*excessWater);

The number of instances where the threshold was crossed should be captured in the numOfCrossings variable.

If a measurement exceeds the threshold, the difference between the measured value and the threshold value should be captured in the excessWater value.

For example, given the following array of samples:

[-1, 2, 5, 7, 3, 6, 1 ]

The numSamples would be 7 and if a threshold level of 4 is specified, then there would be 2 crossings.  The excess amount of water would be 6.

If the calculation has been completed, the function should return 1. On any error, the function should return 0.

# BugFix Problem 2:

Are there any potential issues with the following code?

#include <stdio.h>

typedef enum  
{  
 ID\_1,  
 ID\_2,  
 ID\_3,  
} ciena\_id\_t;

void parseInput(int argc, char \*argv[], ciena\_id\_t \*output)  
{  
 /\*Convert the number 1 to ciena ID\_1,  
 Convert the number 2 to ciena ID\_2,  
 For all other values, convert to ID 3\*/  
  if (argv[1] == "1")  
 {  
 \*output = ID\_1;  
 }  
 else if (argv[1] == "2")  
 {  
 \*output = ID\_2;  
 }  
 else  
 {  
 \*output = ID\_3;  
 }  
}

int main(int argc, char \*argv[])  
{  
 ciena\_id\_t i;

   /\*Convert the user input to the ciena id value\*/

if (argc < 2)  
 {  
 printf("Not enough arguments");  
 }  
  
 parseInput(argc, argv, &i);

if (i == ID\_1 || ID\_2)  
 {  
 printf("ID is ID\_1 or ID\_2\n");  
 }  
 else  
 {  
 printf("ID is something else\n");  
 }

return 0;  
}

# BugFix Problem 3:

Are there any potential issues with the following code snippet?

#include <stdio.h>

typedef struct {  
 int\* a;  
 int\* b;  
 int\* c;  
} result\_t;

void inc(result\_t \*t)  
{  
 int c;  
 c = \*(t->a) + \*(t->b);  
 t->c = &c;  
}

int main(void)  
{  
 result\_t t;  
 int a = 5;  
 int b = 10;  
   
 t.a = &a;  
 t.b = &b;  
   
 inc(&t);

printf("The incrmeneted value is %d\n", \*t.c);

return 0;  
}

# Programming Problem 4:

Given a string that contains sentences, write a function that will parse and enumerate each word and return a count of each word in a singly linked list. Essentially the linked list is a dictionary of all the words and a count is associated with each word. The words should be normalized to contain only lowercase characters and punctuation such as hyphens (e.g. follow-up) and apostrophes (e.g. Ciena’s). Periods, commas and exclamation marks should be filtered out.

The linked list structure should, at a minimum, have the following members:

typedef struct wordCount

{  
 char \*word;

unsigned int count;

struct wordCount \*ptr;

} wordCount\_t;

(More members may be added for internal processing if necessary. If more members are added, please add them after ptr).

The function should return a pointer to the head of the linked list, or NULL if there are no characters or if the input cannot be processed.  The last entry in the linked list should have a ptr value of NULL to indicate that this is the end of the list. The function signature should align to the following:

wordCount\_t \*countStrings(char \*input);

As an example, if the following string is supplied as input to the function,

“Ciena corporation is a global supplier of telecommunications networking equipment software and services that support the delivery and transport of voice video and data service. With nearly 25 years of industry leadership we support more than 1300 of the world’s largest most reliable networks. As a follow-up, please contact us to discuss Ciena’s strengths!”

a linked list would be returned indicating the following words (note that the order of the words should not matter). The linked list would be similar to the following (the number indicates the count for each word):

ciena - 1

corporation - 1

is - 1

a - 2

global - 1

supplier - 1

of - 4

telecommunications - 1

networking - 1

equipment - 1

software - 1

and - 3

services - 1

that - 1

support - 2

the - 2

delivery - 1

voice - 1

video - 1

data - 1

service - 1

with - 1

nearly - 1

25 - 1

years - 1

industry - 1

leadership - 1

we - 1

more - 1

than - 1

1300 - 1

world’s - 1

largest - 1

most - 1

reliable - 1

networks – 1

as – 1

follow-up – 1

please – 1

contact – 1

us – 1

to – 1

discuss – 1

ciena’s – 1

strengths - 1

The performance of the solution will be measured as part of the evaluation. Standard C library functions such as strtok\_r() may be used as part of your solution. If additional header files are required, please incorporate the necessary #include statements in your submission.