CIS 1201 Prefinal Mock Exam – Test 1 of 2 (25 points, 20 minutes)

Instruction: Supply the correct answers to each question. Give only what is being asked.

Place your answers at the designated areas of the answer sheet provided.

1-2. Write a C statement that is equivalent to the following: rewind(fp);

A linked list with head pointer L is populated with three elements.

3. How many pointers to nodes are found in the list?
4. How many variables are stored outside of the heap memory?
5-6. If a pointer A is declared and is currently holding the value of L, how many times does the value of A change if A is used to traverse until it detects the end of the list?

Function checkList() checks if a given list is empty or not. It will return Y if it is empty and N otherwise. The datatype of LL is a pointer to a node located in the heap.

7-9. Fill in the blank with the missing statement to complete the function. (1 statement only)

```
char checkList(LL test){
   _____________________________________
}
```

Given the declaration:

```
typedef struct A{
   char B[1000];
   float C[2];
}D;
```

```
typedef struct E{
   D F;
   struct E* G;
}H,*J;
```

```
J K;
```

How many bytes in memory are allocated to a variable of the following datatypes in Dev-C++ IDE?
10. D **11.** H **12.** J

Assume that the list declared is still empty:
13-14. Write the C statement that will allocate for a new node.
15-16. Write the C statement that will display the string of the first node of the list.
17-18. Write the C statement that will declare a pointer *ptr* which will be used to point to the given head pointer of the list.

Given the code fragment below:

```
#define N 5
FILE* fp = fopen("Test2.bin","wb");
int ctr, x, y = sizeof(int), z;
if(fp!=NULL){
  for(ctr=x=0; ctr<N; ctr++){
     x = N % (ctr + 3) + N;
     if(x==5)
        fseek(fp,0,0);
     fwrite(&x,y,1,fp);
  }
  z = ftell(fp);
  fclose(fp);
}
```

19-23. List in order the entries stored in the file after execution of the code fragment.

24-25. Determine the value of z.

CIS 1201 Prefinal Mock Exam – Test 2 of 2 (75 points, 120 minutes)

Instruction: Write the code of the functions based on the given specifications. Make your code as efficient as you can and avoid redundant statements when possible. Only one return statement shall be allowed for every function created. Conditions in the iteration and alternation statements must be relational operators.

A list of all enrolled Networking 1 students this year (which consists of only 2 groups) is stored in a single and unsorted linked list. See the definitions below.

```
typedef struct{
  char LN[16]; /* last name */
  char FN[24]; /* first name */
  char MI; /* middle initial */
}name;


typedef struct{
  char Pname[8]; /* program name
                    (e.g. BSCS) */
  char Pdesc[64]; /* program
                     description */
}program;
```

```
typedef struct{
  unsigned int year: 11;
  unsigned int month: 4;
  unsigned int day: 5;
}date; /* stores the date enrolled */


typedef struct{
  int SBAscore; /* perfect score is 100 */
  float grade[2]; /* midterm[0] and
                     final[1] grade */
}grades;
```

```
typedef struct{
  name N;
  char gender; /* M or F */
  char ID[8]; /* unique stud identifier */
  program P;
  char group; /* group, either A or B */
  date D;
  grades G;
}student;

typedef struct cell{
   student S;
   struct cell* next;
}*LL; /* linked list definition for all
         Networking 1 students (unsorted) */
```

Problem A: (15 points)

Due to the pandemic causing the delay on releasing of grades from the previous semester subjects, there are students who enrolled late in Sir G's Networking class. You are tasked to create a function that will append the record of the late enrollee at the end of sir's existing record stored using linked list implementation. The user shall be able to repeatedly call and execute this function for every late student to be added.

Create a function insertLateEnrollee() that will insert a student record of the given late enrollee after the end of the given list of student records already enrolled beforehand. Insertion shall take place only if the given student's ID does not exist yet in the list to avoid duplication of data.

Problem B: (20 points)

Having difficulty tracking the current students enrolled under Sir G, he wanted to reorganize his unified class record by separating the students according to their group, and also to have the students from each group sorted in increasing order as well according to ID in the new lists.

Create a function createComboList() which takes the data from the given generic list, and then creates a new combo list which will store every student record by having each inserted into the corresponding sub list. The two sub lists are to be sorted according to student ID, thus all insertions of records must take place in their proper positions within the sub list. The newly created combo list will be returned to the function afterwards.**

```
**Combo List definition:

typedef struct{
   LL ListA; /* will contain Group A students */
   LL ListB; /* will contain Group B students */
}ComboList;
```

Problem C: (20 points)

In preparation for the final exam, Sir G wants to identify the students that are eligible to take the exam, and that is by examining whether each student obtained a passing grade in the Skills Based Assessment (SBA) which is at least 60% of the perfect score. He wants to have a file with him containing only the names of those students who are qualified for the final exam. In addition, the records of students with failing SBA grades shall be wiped out from the original class record.

Create a function writeFinalsQualifiers() which removes all nodes in the given generic list containing the student records with failing SBA grades, and then places the names of the remaining students with passing SBA grades in a new file called "FinalsQualifiers.dat".

Problem D: (20 points)

The final exam has been conducted and Sir G now wants to update his file by placing the final grades to each of the students found in his file named "StudentGrades.dat" containing complete student records (not only the names) of those who took the final exam. Also, he wants to know how many people have passed his subject by getting the average of the midterm and final grade of each student (which should be 3.0 or below).

Create a function updateRecords() which will do the following:

Open the file mentioned above and will read the student records one entry at a time.

For each entry being read, a message will be displayed on screen prompting the user to input the student's final grade (Format: "Enter final grade of student# X: " where X represents the student ID) until a valid input is made, which will be only as follows: Between 1.0 to 3.0 inclusive, and 5.0. The entry will then be updated in the file with the inputted final grade.

Finally, the function shall return to the calling function the total number of students averaging a passing grade for the course.

-- END OF EXAM --

-- REVIEW YOUR ANSWERS!! --