CS 135502 Final

1. (10%) What are the outputs of the following program? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>
#define SIZE 5

void x ( int b[], int size );
void y ( int e );

int main()
{
   int a[ SIZE ] = { 1, 2, 3, 4, 5 };
   int i;

   for ( i = 0; i < SIZE; i++ ) {
      printf( "%3d", a[ i ] );
   }
   printf( "\n" );

   x ( &a[0], SIZE );

   for ( i = 0; i < SIZE; i++ ) {
      printf( "%3d", a[ i ] );
   }
   printf( "\n" );

   y ( a[0] );

   for ( i = 0; i < SIZE; i++ ) {
      printf( "%3d", a[ i ] );
   }
   printf( "\n" );

   x ( a, SIZE );
```

```
   for ( i = 0; i < SIZE; i++ ) {
      printf( "%3d", a[ i ] );
   }
   printf( "\n" );

   return 0;
}

void x( int b[], int size )
{
   int j;

   for ( j = 0; j < size; j++ ) {
      b[ j ] *= 2;
   }
}

void y( int e )
{
   e *= 2;
}
```

2. (10%) What are the outputs of the following program? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>

void staticArrayInit( void );
void automaticArrayInit( void );

int main()
{
   int array[ 3 ] = { 10, 20, 30 };
   int i;

   staticArrayInit();
```

```
   automaticArrayInit();

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] );
   }
   printf ("\n");

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] += 5 );
   }
   printf ("\n\n");

   staticArrayInit();
   automaticArrayInit();

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] );
   }
   printf ("\n");

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] += 5 );
   }
   printf ("\n\n");

   return 0;
}

void staticArrayInit( void )
{
   static int array[ 3 ] = { 1, 2, 3 };
   int i;

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] );
   }
   printf ("\n");
```

```
   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] += 5 );
   }
   printf ("\n\n");
}

void automaticArrayInit( void )
{
   int array[ 3 ] = { 1, 2, 3 };
   int i;

   for ( i = 0; i <= 2; i++ ) {
      printf("%d  ", array[ i ] );
   }
   printf ("\n");

   for ( i = 0; i <= 2; i++ ) {
      printf( "%d  ", array[ i ] += 5 );
   }
   printf ("\n\n");
}
```

3. (10%) The following program should calculate the cube of a variable by using call-by-reference. Identify and correct the errors in the program. The program should function as the comments indicate. You will be deducted some points if you point out anything which is correct.

```
/* Cube a variable using call-by-reference with a pointer
argument */

#include <stdio.h>

void cubeByReference( int *nPtr ); /* prototype */

int main()
{
   int number = 5; /* initialize number */
```

```
   printf( "The original value of number is %d", number );

   /* pass address of number to cubeByReference */
   cubeByReference( number );

   printf( "\nThe new value of number is %d\n", number );

   return 0; /* indicates successful termination */

} /* end main */

/* calculate cube of *nPtr; modifies variable number in main
*/
void cubeByReference( int *nPtr )
{
   nPtr = nPtr * nPtr * nPtr;  /* calculate cube */
} /* end function cubeByReference */
```

4. (10%) What are the outputs of the following program? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>

void printCharacters( const char *sPtr );

int main()
{
   char string[] = "I love programming";

   printf( "The string is:\n" );
   printCharacters( string );
   printf( "\n" );

   return 0;
}

void printCharacters( const char *sPtr )
```

```
{
   for ( ; *sPtr != '\0'; sPtr++ ) {
      printf( "%c", *sPtr );
   }
}
```

5. (10%) What are the outputs of the following program? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>

struct card {
   char *face;
   char *suit;
};

int main()
{
   struct card aCard;
   struct card *cardPtr;

   aCard.face = "Ace";
   aCard.suit = "Spades";

   cardPtr = &aCard;

   printf( "%s%s%s\n%s%s%s\n%s%s%s\n", aCard.face, " of ",
aCard.suit, cardPtr->face, " of ", cardPtr->suit,
      ( *cardPtr ).face, " of ", ( *cardPtr ).suit );

   return 0;
}
```

6. (10%) What does the following program do? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>

void mystery1( char *s1, const char *s2 );

int main()
{
   char string1[ 80 ];
   char string2[ 80 ];

   printf( "Enter two strings: " );
   scanf( "%s%s" , string1, string2 );

   mystery1( string1, string2 );

   printf("%s", string1 );

   return 0;

}

void mystery1( char *s1, const char *s2 )
{
   while ( *s1 != '\0' ) {
      s1++;
   }

   for ( ; *s1 = *s2; s1++, s2++ ) {
      ;
   }
}
```

7. (10%) What does the following program do? If you think something is wrong in this program, identify and correct the errors. (You will be deducted some points if you point out anything which is correct.)

```
#include <stdio.h>

int mystery3( const char *s1, const char *s2 );
```

```
int main()
{
   char string1[ 80 ];
   char string2[ 80 ];

   printf( "Enter two strings: " );
   scanf( "%s%s", string1 , string2 );

   printf( "The result is %d\n", mystery3( string1,
string2 ) );

   return 0;
}

int mystery3( const char *s1, const char *s2 )
{
   for ( ; *s1 != '\0' && *s2 != '\0'; s1++, s2++ ) {

     if ( *s1 != *s2 ) {
       return 0;
     }

   }

   return 1;
}
```

8. (10%) (a) In the following program, which "printf" will not be executed? Why? (b) Rewrite "`void gameover ()`" in the following program so that the result of the function will be blinked at least four times.

```
#include <stdio.h>
#include <windows.h>

static HANDLE hConsole = 0;

void gotoxy(int X, int Y)
```

```
{
        COORD coord;
        hConsole = GetStdHandle(STD_OUTPUT_HANDLE);

        coord.X = X;
        coord.Y = Y;
        SetConsoleCursorPosition(hConsole, coord);
}

void gameover ()
{
        system ("cls");

        gotoxy (15, 8);
        printf ("------------------------------");
        gotoxy (15, 9);
        printf ("|      GAME OVER              |");
        gotoxy (15, 10);
        printf ("------------------------------");

        gotoxy (0, 23);
        exit (0);
}

int main ()
{
        gameover ();

        printf ("Game Over");
        return 0;
}
```

9. (10%) The following code is part of a completed program which can insert an element into a linked list or delete an element from a linked list. The program can also print out the element(s) in the linked list. In the main program, the valuable of "`startPtr`" points to the first element of the linked list. Write the missing lines in the function of "`insert`" mark as **`/* write this line */`**. The program should be able to insert a character in the order of its ASCII code, that is,

alphabetically. The program should function as the comments indicate.

```
#include <stdio.h>
#include <stdlib.h>

struct listNode {   /* self-referential structure */
   char data;
   struct listNode *nextPtr;
};

typedef struct listNode ListNode;
typedef ListNode *ListNodePtr;

void insert( ListNodePtr *, char );
char delete( ListNodePtr *, char );
int isEmpty( ListNodePtr );
void printList( ListNodePtr );
void instructions( void );

int main()
{
   ListNodePtr startPtr = NULL;
   int choice;
   char item;

   instructions();  /* display the menu */
   printf( "? " );
   scanf( "%d", &choice );

   while ( choice != 3 ) {

      switch ( choice ) {
         case 1:
            printf( "Enter a character: " );
            scanf( "\n%c", &item );
            insert( &startPtr, item );
            printList( startPtr );
            break;
```

```
         case 2:
            if ( !isEmpty( startPtr ) ) {
               printf( "Enter character to be deleted: " );
               scanf( "\n%c", &item );

               if ( delete( &startPtr, item ) ) {
                  printf( "%c deleted.\n", item );
                  printList( startPtr );
               }
               else
                  printf( "%c not found.\n\n", item );
            }
            else
               printf( "List is empty.\n\n" );

            break;
         default:
            printf( "Invalid choice.\n\n" );
            instructions();
            break;
      }

      printf( "? " );
      scanf( "%d", &choice );
   }

   printf( "End of run.\n" );
   return 0;
}

/* Print the instructions */
void instructions( void )
{
   printf( "Enter your choice:\n"
           "   1 to insert an element into the list.\n"
           "   2 to delete an element from the list.\n"
           "   3 to end.\n" );
}
```

```
/* Return 1 if the list is empty, 0 otherwise */
int isEmpty( ListNodePtr sPtr )
{
   return sPtr == NULL;
}

/* Insert a new value into the list in sorted order */
void insert( ListNodePtr *sPtr, char value )
{
   ListNodePtr newPtr;      /* pointer to new node */
   ListNodePtr previousPtr; /* pointer to previous node in
list */
   ListNodePtr currentPtr;  /* pointer to current node in list
*/

   newPtr = malloc( sizeof( ListNode ) ); /* create node */

   if ( newPtr != NULL ) { /* is space available */
      /* write this line */ /* place value in node */
      newPtr->nextPtr = NULL; /* node does not link to another
node */

      previousPtr = NULL;
      currentPtr = *sPtr;

      /* loop to find the correct location in the list */
      while ( currentPtr != NULL && value > currentPtr->data )
{
         /* write this line */          /* walk to ...   */
         currentPtr = currentPtr->nextPtr;  /* ... next node
*/
      } /* end while */

      /* insert new node at beginning of list */
      if ( previousPtr == NULL ) {
         newPtr->nextPtr = *sPtr;
         /* write this line */
      } /* end if */
```

```
      else { /* insert new node between previousPtr and
currentPtr */
         previousPtr->nextPtr = newPtr;
         /* write this line */
      } /* end else */

   } /* end if */
   else {
      printf( "%c not inserted. No memory available.\n",
value );
   } /* end else */

} /* end function insert */
```

10. (10%) There are 10 customers. Each customer has a different last name, first name, customer number, and phone number. Write a program which will list the customers in the order of last name, first name, customer number, or phone number based on the user input. Your program must:

    A. Define and use a structure which has at least four members as follows

```
struct customer {
    char lastName[15];
    char firstName[15];
    int customerNumber;
    char phoneNumber[11];
};
```

    B. Your main program needs to call at least the following functions written by yourself:
       (a) a function takes user input
       (b) a function sorts customers based on the user input (you can use any sorting algorithm)
       (c) a function prints the sorting result
    C. The main program will repeat until a predefined string is entered. You need to tell the user how to end the program.
    D. The output should look like this when the sorting is based on customer number:

Last Name	First Name	Customer Number	Phone Number
Smith	John	30001	212-582-3344

Goodman	Jack	30008	716-256-7777
Jordan	Marry	72567	408-222-7777
.....			