CS 125 **Data Structures and Problem Solving** Test 1
Name: Maximum Points: 65

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem.

1. [8 pt] Write a function `fe` that takes as input a character string and evaluates to true if there is a file corresponding to the character string in the current directory. It returns a false if there is no such file. Also write a little driver function that will just call this function and print the relevant message depending on the outcome of the function. The driver function will ask the user to enter the file name (or character string), call the function `fe` with the string (after manipulating the string, if necessary), and print the message.

2. [5 pt] What does the following program do?

```
#include <stdio.h>

int mystery ( unsigned );

main()
{
    unsigned x;

    printf ( "Enter an integer: " );
    scanf ( "%u", &x );
    printf ( "The result is %d\n", mystery ( x ) );

    return ( 0 );
}

int mystery ( unsiged bits )
{
    unsigned i,
             mask = 1 << 15,
             total = 0;

    for ( i = 1; i <=16; i++, bits <<= 1 )
        if ( ( bits & mask ) == mask )
            ++total;

    return ( total % 2 ? 0 : 1 );
}
```

3. [16 pt] Given the following structure definitions and variable declarations,

```
struct customer
{
    char  lastname[15];
    char  firstname[15];
    int customer_num;
    struct
    {
        char phone_num[11];
        char address[50];
        char city[15];
        char state[3];
        char zip[6];
    } personal;
} customer_rec, *customer_ptr;

customer_ptr = customer_rec;
```

write a separate expression that can be used to access the structure members in each of the following parts.

(a) Last name of customer record.

(b) Last name of the structure pointed to by `customer_ptr`

(c) First name of customer record.

(d) First name of the structure pointed to by `customer_ptr`

(e) Customer number of customer record.

(f) Customer number of the structure pointed to by `customer_ptr`

(g) Phone number in customer record

(h) Phone number in the structure pointed to by `customer_ptr`

(i) Address in customer record

(j) Address in the structure pointed to by `customer_ptr`

(k) City in customer record

(l) City in the structure pointed to by `customer_ptr`

(m) State in customer record

(n) State in the structure pointed to by `customer_ptr`

(o) Zipcode in customer record

(p) Zipcode in the structure pointed to by `customer_ptr`

4. [10 pt] Write a statement to allocate memory for 100 records of the structure data type defined in the previous question. How will you access the ith record within the allocated memory. Write a statement to print first five letters in the address field of the ith record, but print the entire field if the field contains less than five characters.

5. [4 pt] In the `makefile` below, identify the targets and the prerequisites.

```
CC = gcc
CFLAGS = -g
INCLUDEDIR = .

enum: enum.o
${CC} ${CFLAGS} -o $@ enum.o

enum.o: enum.c types.h
${CC} ${CFLAGS} -I${INCLUDEDIR} -c enum.c
```

6. [8 pt] Explain why software might need to be modified

   (a) in the design phase

   (b) in the coding phase

   (c) in the testing phase

   (d) in the maintenance phase

7. [5 pt] What is the result of the execution of following program?

```
#include <stdio.h>

main()
{
    char s[] = "Harry",
         t[] = "Hong"
    if ( s > t )
        printf ( "%s comes before %s\n", s, t );
    else
        printf ( "%s comes before %s\n", t, s );
}
```

8. [5 pt] In the following program, the HIGH_SPEED flag works, but the DIRECT_CONNECT flag does not. Why?

```
#include <stdio.h>

#define HIGH_SPEED      ( 1 << 7 )      /* modem is running fast */
#define DIRECT_CONNECT  ( 1 << 8 )      /* Hardwired connection  */

char flags = 0;                         /* Start with nothing    */

main()
{
    flags |= HIGH_SPEED;                /* modem is running fast         */
    flags |= DIRECT_CONNECT;            /* because of hardwire connection */

    if ( flags & HIGH_SPEED )
         printf ( "High speed set\n" );
    if ( flags & DIRECT_CONNECT )
         printf ( "Direct connection established\n" );
}
```

9. [4pt] What is wrong with the following initialization?

```
int b[10] = { 0 }, i;
for ( i = 0; i <= 10; b[i++] = 1 );
```