Problem 1

What will be the output of the following program? Explain how you reached your solution by filling in comments with the missing values that result after each statement is run.

```c
int main() {
    int a[5] = {0, 1, 2, 3, 4};
    int x, w, y;
    x = a[2]--;
    // x = _, a = { _, _, _, _, _ }
    w = --a[2];
    // w = _, a = { _, _, _, _, _ }
    y = a[++x];
    // x = _, y = _, a = { _, _, _, _, _ }
    printf("%d, %d, %d", x, w, y);
    return 0;
}
```

Problem 2

Identify an error in the following C code fragment (assume that this code is embedded in the otherwise correct program). Suggest a correction.

```c
int x[8], i;
for (i = 0; i <= 9; ++i)
    x[i] = i;
```

Problem 3

Given the following two dimensional array declaration: `double m[2][4];`

1. Write a code fragment that fills the entire array with some values (does not matter what they are) - use the array notation
2. Write a code fragment that fills the entire array with some values (does not matter what they are) - use the pointer notation

Problem 4

Write a declaration for a function that given two integers computes and returns the larger. The two integers passed to the function should not be changed.

Problem 5

Write a definition for a function that given a character, converts it to lower case (if the character passed to this function is already lower case it should stay unchanged). This function does not return any value.

Problem 6

Write a definition for a function that given two strings computes their total length. You cannot use the `strlen()` function. The function should take only the two strings as its parameters.
Problem 7

Write a definition for a function that given an array of integers sorts the numbers in the array from smallest to largest. Use mergesort algorithm. Hint: what are the parameters that such function needs?

Problem 8

Write a definition for a structure that represents a student. A student should have name, gpa value and a list of course taken in a given semester.

Problem 9

Write a binary search function that searches for a word in a sorted array of words.

Problem 10

Assume we are executing the following program on a 64-bit machine:

1. What is the output?

2. If we change p’s declaration to be double *p; and use the type casting with malloc to be (double*), what is the output?

3. In both cases, how many bytes are freed by calling free(…)?

```c
int main() {
    int *p;
    p = (int *) malloc(32);
    printf("%d\n", sizeof(p));
    free(p);
    return 0;
}
```

Problem 11

What is printed by the program shown below? What is the name of the sequence of numbers that are printed? (Try to answer this question without running the code.)

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct node* pointer;

struct node {
    int data;
    pointer next;
};

int main(void) {
    pointer x, y, t;
    x = malloc(sizeof(struct node));
    y = malloc(sizeof(struct node));
    x->next = y;
    x->data = 1;
    y->next = x;
    y->data = 1;
    t = x;
    int counter = 0;
```
for (counter = 0; counter < 10; counter++) {
    t->data = x->data + y->data;
    printf("%d\n", t->data);
    t = t->next;
}
return 0;

Problem 12

A node for a binary search tree is defined as follows:

```c
struct bst_node {
    char * data;
    struct bst_node * right;
    struct bst_node * left;
}
```

1. Write a function that given a pointer to the root (possibly equal to NULL for an empty tree) and a string, adds a new node in the appropriate position of the BST. (That node should contain the string as its data.)

2. Write a function that removes the node storing the smallest value.

Problem 13  (This one is not easy)

Consider the following C program:

```c
#include <stdlib.h>
int main(int argc, char** argv) {
    float x[3][4];
    float *a = &(x[3][0]);
    float *b = a;
    b--;
    float *c = &(x[2][4]);
    float *d = &(c[1]);
    int e = x[3] - x[1];
    int f = &(x[3]) - &(x[1]);
    int g = &(x[3][0]) - &(x[1][0]);
    float * h = x[3] - 2;
    return 0;
}
```

Assume that this program runs on an architecture where pointers are 8 bytes large, and the sizes of char and float are 1 and 4 bytes, respectively. Further assume that the value of pointer variable x is the address 100 (that is 0x64 in hexadecimal notation). What are the values of variables a through h just before the return statement?

Warning 1: this program will generate some warning if you compile it - these are not errors and for the purpose of this exercise should be ignored.

Warning 2: this is a hard problem that tests your understanding of pointers. Explain your answers to help us understand how you arrived at them.