(i) (5 pts.)

Consider the following C program.

```c
#include <stdio.h>

int f(int x, char a[]) {  
  x = x + 2;
  a[4] = '\0';
  return x;
}

int main(void) {
  int y = 0;
  char s[] = "message."
  int z = 0;

  printf("one: y = %d, s = %s, z = %d\n", y, s, z);
  z = f(y, s);
  printf("two: y = %d, s = %s, z = %d\n", y, s, z);

  return 0;
}
```

Show the output of this program exactly as it appears on standard output.
Write your solution to Question (i) on this page. Please clearly indicate your solution and show all work on this page.
Consider the following C program.

```c
#include <stdio.h>

int main(void) {
    char c1, c2 = 'P';
    char *p1, *p2 = &c2;

    *p2 = c2 + 1;
    c1 = *p2 + 1;
    p1 = &c1;

    printf("c1 = %c, c2 = %c, *p1 = %c, *p2 = %c\n",
           c1, c2, *p1, *p2);

    return 0;
}
```

Suppose that each character occupies one byte of memory. Suppose also that the value assigned to `c1` is stored in (hexadecimal) address F8C and the value assigned to `c2` is stored in address F8D.

PART 1: Show the output of this program exactly as it appears on standard output.

PART 2:  
  a) What value is represented by `&c1`?  
  b) What value is represented by `&c2`?  
  c) What value is assigned to `p1`?  
  d) What value is assigned to `p2`?
Write your solution to Question (ii) on this page. Please clearly indicate your solution and show all work on this page.
(iii) (5 pts.)

Write a program that carries out the following sequence of steps.

1. Prompts the user to enter a positive integer from standard input.
2. Reads the value entered into a variable \( x \).
3. Prints an error message to standard error and exits the program if \( x \) is not positive-valued.
4. Dynamically allocates an array \( A \) of floating point numbers that has \( x \) elements.
5. Prompts the user to enter \( x \) floating point values from standard input.
6. Reads the values entered into successive elements of the array \( A \).
7. Computes the maximum value over all elements of \( A \).
8. Prints this maximum value to standard output.

Develop a complete C code implementation of this program. No error handling is required in your program beyond what is described in the specifications above.
Write your solution to Question (iii) on this page. Please clearly indicate your solution and show all work on this page.
PART 1: Define a structure template called `date`, which contains three integer members that represent the month, day, and year, respectively of a calendar date.

PART 2: Write a function that takes as an argument a pointer to such a date structure; prompts the user to enter a month, day, and year (each in integer form) from standard input; and then reads the entered values into corresponding members of the structure that is pointed to by the argument. A prototype of this function is as follows.

```c
void read_date(struct date *x);
```

PART 3: Write a function that takes as an argument a pointer to a date structure, and displays to standard output the specified date in the form mm/dd/yy, where mm, dd, and yy represent the numeric (integer) month, day, and year, respectively. A prototype of this function is as follows.

```c
void display_date(struct date *x);
```

As an example of the required functionality, consider the following program:

```c
int main(void) {
    struct date d1 = {0, 0, 0};
    struct date d2 = {0, 0, 0};

    read_date(&d1);
    display_date(&d1);
    read_date(&d2);
    display_date(&d2);

    return 0;
}
```

Consider also the following sample input to this program:

```
4 12 08
7 31 10
```

Then the corresponding output, and shown on standard output, would be as follows.

```
Enter month, day, and year
4/12/8
Enter month, day, and year
7/31/10
```

NOTE: No error handling is required in the functions `read_date` and `display_date`. 
Write your solution to Question (iv) on this page. Please clearly indicate your solution and show all work on this page.
Problem 1:

one: \( y = 0, \ s = \text{message}, \ z = 0 \)

two: \( y = 0, \ s = \text{mess}, \ z = 2 \)

Problem 2:

c1 = R, c2 = Q, *p1 = R, *p2 = Q

F8C
F8D
F8C
F8D
Problem 3:

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

int main(void) {
    int x = 0;
    float *A = NULL;
    int i = 0;
    float max = 0;

    printf("Enter a positive integer.\n");
    scanf("%d", &x);
    if (x <= 0) {
        fprintf(stderr, "Positive value expected.\n");
        exit(1);
    }

    A = malloc(x * sizeof(float));

    printf("Enter %d positive integers.\n", x);
    for (i = 0; i < x; i++) {
        scanf("%f", &A[i]);
    }

    max = A[0];
    for (i = 1; i < x; i++) {
        if (A[i] > max) {
            max = A[i];
        }
    }

    printf("The maximum value is %f\n", max);

    return 0;
}
```
Problem 4:

#include <stdio.h>

struct date {
    int month;
    int day;
    int year;
};

void read_date(struct date *x) {
    printf("Enter month, day, and year\n");
    scanf("%d%d%d", &(x->month), &(x->day), &(x->year));
}

void display_date(struct date *x) {
    printf("%d/%d/%d\n", x->month, x->day, x->year);
}