SE185: Problem Solving in Software Engineering
Midterm 2: Thursday (11-04-21)

Last Name: First Name: Lab Section:

1. True/False Questions (10 x 1p each = 10p)

(a) The name of a character array is also a pointer which points to the first element of the array. TRUE / FALSE

(b) The calloc() function reserve memory and initializes to 0 TRUE / FALSE

(c) The following statements will have compilation error TRUE / FALSE

```
int x[3] = {2, 3, 9};
int *my_ptr;
my_ptr = x[2];
```

(d) The following printf statement will print 6. TRUE / FALSE

```
int a[5];
a[-2]=3;
printf("%d", a[-2]);
```

(e) char name[15] = "COVID19"; TRUE / FALSE
&name[2] is equivalent to name+2

(f) Typedef double usrVariable; TRUE / FALSE
usrVariable results; is equivalent to double results;

(g) The following statements will give compilation error: TRUE / FALSE

```
int a[100];
a[100]=6;
```

(h) The following statements will give compilation error: TRUE / FALSE

```
int x[3] = {9, 10, 11};
int *my_ptr = x;
printf("%d", *my_ptr);
```

(i) The following printf statement prints the address stored inside the pointer variable x. TRUE / FALSE

```
int* x;
printf("%p", &x);
```

(j) The following statement will give a compilation error: TRUE / FALSE

```
char alphabet [4] = {{N},{A},{M},{E}};
```

2. Pointers (2 x 6p each = 12p)

Given the following block of C code. Write down the contents of **array b** after the following codes are executed. The results are not cumulative (i.e., part (b) is independent of the part (a), etc.)

(a)

```
int a[5] = {2, 4, 6, 12, 24};
int b[5] = {16, 10, 9, 12, 7};
int *p, *q, *r;
p = &b[1];
q = ++p;
(*q)++;
```

Contents of array b = ……

(b)

```
int a[5] = {2, 4, 6, 12, 24};
int b[5] = {16, 10, 9, 12, 7};
int *p, *q, *r;
p = &a[1];
q = &b[1];
while (*q %4!=3){
     *q +=*p;
    ++q;
}
```

Contents of array b = ……

3. Struct with function (5 x 4p each = 20p)

Without altering the program structure, complete the following program that will ask the user to **enter the radius and height of a cylinder**. Then the program **calls a user-defined function** to calculate the **cylinder volume and surface area**. Finally, it prints the cylinder volume and the surface area.

```
#include <stdio.h>
#include <stdlib.h>
#define pi 3.1416
struct cylinder{
   double radius;
   double height;
};
double cylinderProperties(/* (a)  fix me: parameters*/){
  x[0]=(2*pi*(d2.radius)*(d2.radius) + 2*pi*(d2.radius)*(d2.height));
  x[1]=(pi*(d2.radius)*(d2.radius)*(d2.height));
}
void main(){
  /* (b)  fix me: declare a struct variable */
  double x[2];
  printf("Enter radius of the cylinder: ");
  scanf(/* (c)  fix me: get radius input*/);
  printf("Enter height of the cylinder: ");
  scanf(/* (d)  fix me: get height input*/);
  /* (e)  fix me: call the function*/
  printf("Volume=%0.2lf\nSurface Area=%0.2lf\n", x[0], x[1]);
}
```

Enter your answer below:

(a) ----------

(b) ----------

(c) ----------

(d) ----------

(e) ----------

4. Pointers with function (10p)

What is the output of the printf statement in the following program?

```
#include <stdio.h>
#include <stdlib.h>
        int fun(int *ptr, int *z);
void main() {
        int i;
        int *ptr = NULL;
        int y = 0;
        int *z = NULL;
        ptr = malloc(10*sizeof(int));
        z = malloc(sizeof(int));
        for(i=0;i<5;i++){
                *(ptr + i)= (i * i);
        }
        y = fun(ptr,z);
        printf("%d", y);
        free(ptr);
        free(z);
}
        int fun(int *ptr, int *z){
                int x = 0;
                while(*z != 9){
                        *z = *(ptr+x);
                        x++;
                }
                return x;
        }
```

Enter your answer below:

Output of the printf statement: ----------

5. Pointers (10 x 2p each = 20p)

Given the program below, replace the **[pointerSymbol]** in the 10 instances with one of the following terms: '**&**' , '*****' , or '**none**' . (The last **[pointerSymbol10]** is equivalent to '**none**').

```
#include<stdio.h>
#include<stdlib.h>
void main(){
    int i, num;
    float [pointerSymbol1]data;
    printf("Enter total number of elements (1 to 100):");
    scanf("%d", [pointerSymbol2]num);
    data = (float[pointerSymbol3])calloc(num, sizeof(float));
    if([pointerSymbol4]data == NULL){
        printf("Error!!! memory not allocated.");
        exit(0);
    }
    printf("\n");
    for(i = 0; i<num; ++i){
        printf("Enter Number %d: ", i+1);
        scanf("%f", [pointerSymbol5](data + i));
    }
    for(i = 1; i<num; ++i){
        if([pointerSymbol6]data<[pointerSymbol7](data + i)){
            [pointerSymbol8]data = [pointerSymbol9](data + i);
        }
    }
  printf("%0.2f", [pointerSymbol10]data);
}
```

Enter your answer below:

• pointerSymbol1 : • pointerSymbol2 : • pointerSymbol3 : • pointerSymbol4 : • pointerSymbol5 :	• pointerSymbol6 : • pointerSymbol7 : • pointerSymbol8 : • pointerSymbol9 : • pointerSymbol10 :

6. Struct with function and array (7 x 4p each = 28p)

Without altering the program structure, complete the following program that ask the user to enter 135 student names and their midterm exam score. Then the program prints each student name with their exam score. It will then call a function to calculate the average exam score and print the average score.

```
#include <stdio.h>
//----struct definition----
struct studentRecord {
        char name[20];
        double examScore[135];
};
//----function definition----
void grade (/*(a) fix me: */){
        int i;
        double sum = 0;
        for(i=0;i<135;i++){
                sum =  /*(b) fix me: */;

        }
        printf("\nExam average: %lf", sum/3);
}
void main(){
        int i;
        //----struct declaration----
        /* (c) fix me: struct variable that can store info for 135 students*/;
        //----get students name and exam score----
        for(i=0;i<135;i++){
                printf("\nEnter student%d  name: ",i+1);
                scanf("%s", /* (d) fix me: store name of each students*/);
                printf("\nEnter student%d  exam score: ",i+1);
                scanf("%lf",  /* (e) fix me: store each students exam score*/);
        }
        //----print student names = exam score----
        for(i=0;i<135;i++){
                printf("\n%s = %lf", /* (f) fix me: print student name and exam score*/);
        }
        //----call function to calculate exam average and print----
        grade(/*(g) fix me: */);
}
```

Enter your answer below:

(a) ----------

(b) ----------

(c) ----------

(d) ----------

(e) ----------

(f) ----------

(g) ----------

7. Bonus question: Struct (15p)

Write a complete C program that allows a user to keep track of their friend's birthdays. First, define a friend struct with three data members: the first name of the friend, the month of the friend's birthday, and the day of the friend's birthday. Then, write a main function that declares an array of 12 of these structs. The program should then enter a loop with the following behavior:

(a) Ask the user if he/she would like to enter a friend's birthday. You may choose how the user should indicate yes or no.

(b) If the user's input indicates no, the program exits. It does not need to print the contents of the structs before exiting.

(c) If the user's input indicates yes, the program prompt the user to enter information (i.e., name, month, day) for the friend. Store this information in the next struct in your array (i.e., the first friend entered goes in the first element of the array, the second friend entered goes in the second element of the array, etc.).

(d) Repeat the loop.

Question	Max	Score
1. True/False	10	
2. Pointers	12	
3. Struct with function	20	
4. Pointers with function	10	
5. Pointers	20	
6. Struct with function and array	28	
7. Bonus question: Struct	15	
TOTAL:	115	