Midterm 2019, partial answers

Programming Fundamentals for Software and Computer (University of Calgary)

Partial Solution for Midterm Exam Fall 2019

(AR Diagram Solutions Are Not Included)

Important Note: In this exam, if applies, you should assume all required header files are included. Also if necessary you can assume the size of int is 4 bytes and size of pointer is 8 bytes **Section I -**

Multiple Choice Section – Select the best answer (15 marks)

1. Which one of the following variable declaration will be automatically set to zero by the compiler.
   a. Local variables
   b. Function arguments
   c. Static data members of a function
   d. Global variables
   e. (a) and (b) are both correct answers
   f. (c) and (d) are both correct answers
   g. None of the above are correct answers

2. What is the output of the following program

```
void fun() {
  int j = 60;
  static int m = 30;
  printf("%d%d", ++j, ++ m);
}
```

```
int main(){
  int j;
  for( j = 0; j < 2; j++)
     fun();
  return 0;
}
```

   a. 60306030
   b. 61316131
   c. 61316132
   d. 61316231
   e. None of the above

3. Consider the following code segment:

```
1. void fun() {
2.     char s[10] = "Mars";
3.     const char* p1 = s;
4.     char *const p2 = &s[2];
5.     *p2 = 'A';
6.     p1 = p1 + 1;
7.     p1 = p2;
8. }
```

Which one of the following statements is correct?

   a. There is compilation error on line 4
   b. There is compilation error on line 5
   c. There is compilation error on line 6
   d. There is compilation error on line 7
   e. None of the above

4. What is the output of this code segment:

```
char * st1 = "LECTURE";
st1 = st1 + 2;
printf("%s\n", &*(st1++));
```

   a. ECTURE
   b. LECTURE
   c. CTURE
   d. NULL
   e. None of the above

5. What is the output of this code segment if user input is 4.2

```
int i, j;
printf ("%d\n", scanf("%d%d", &i, &j));
```

   a. 4
   b. 2
   c. 1
   d. 0
   e. None of the above

6. What is the output of the following code segment:

```
char s1[100] = "ENG";
char s2[100] = "T";
printf("%s\n", strcat(s1, strcat(s2, "AB")));
```

   a. ABENGT
   b. TENGAB
   c. ABTENG
   d. ENGTAB
   e. None of the above.

7. What is the output of the following fprintf function call on the computer screen. Assume □, represents a space character on the screen :

```
double d = 9;
fprintf(stdout,"%-5.1f%+6.2f%4.1f\n", d, d, d);
```

   a. 9.0□□□□□□□□□□□□+9.00□□□□9.0
   b. 9.0□□□+9.00□9.0
   c. □□□□□9.0□□□□□□+9.00□□□□9.0
   d. None of the above

8. What is the output of the following code segment?

```
int numbers [7] = {11, 22, 33, 44, 55, 66, 17};
int *p = &numbers[7];
printf("%d\n", (int)(p - numbers));
```

   a. 0
   b. Garbage
   c. 7
   d. 6
   e. -7
   f. None of the above

9. Which one of the following lines of C code gives a compilation error?

```
1 const char* s = "1234567";
2 const char* p = s;
3 s = 3 + s;
4 s = 3 - s;
5 long y = p - s;
```

   a. Line 1
   b. Line 2
   c. Line 3
   d. Line 4
   e. Line 5
   f. None of the above

Use the following code segment to answer questions 10, and 11.

```
typedef struct Car{
    char make[6];
    int seats;
    double price;
    struct Car* next;
}Car;
```

10. What is the output of the following code?

```
Car c2 = {"Honda", 7, 3000, NULL};
Car c1 = {"Ford", 5, 20000, &c2};
c2.seats = c1.seats;
*(c1.next->make + 1) = *c1.make + 1 ;printf("%s", c2.make);
```

   a. Garbage
   b. HGnda
   c. Fonda
   d. HFond
   e. None of the above

11. Which line in the following code segment gives a compilation error?

```
1 Car c2 = {"Honda", 7, 3000, NULL};
2 Car c1 = {"Ford", 5, 20000, &c2};
3 *c1.make = *c2.make;
4 c1 = c2;
5 c2.make = c1.make;
```

   a. Line 3
   b. Line 4
   c. Line 5
   d. None of the above.

Use the following code segment to answer questions 12, 13.

```
typedef struct player_info_s{
    char name[20];
    int age;
}Player_info;

typedef struct player_s{
    Player_info p_info;
    double speed;
}Player;

int main (){
    Player flames [2] = {{{"Iginla", 33}, 1.00},{{"Tanguay", 30}, 0.8}};
    Player *p;
    p = flames;
    return 0;
}
```

12. With regards to the above code, which one of the following lines of code can be used to change "age" for the first index of the array "flames":
    a. (*p).p_info.age = 32;
    b. p ->p_info.age = 32;
    c. flames[0].p_info.age = 32;
    d. (*flames).p_info.age = 32;
    e. flames ->p_info.age = 32;
    f. All of the above

13. With regards to the above code, what would be the output of the following printf statement:

```
const char* pc = (*(flames + 1)).p_info.name + 3 ;
printf("%s",   pc );
```

    a. Garbage
    a. Iginla
    b. nla
    c. guay
    d. Tanguay
    e. None of the above

14. Consider the following small program:

```
void foo (int *x, int *y, int *z){
      *z = (*x) * (*y);
}
```

Which one of the following prototype of function foo are incorrect?
   a. void foo (int a[60], int b[70], int c[1000]);
   b. void foo (int a[], int b[], int c[]);
   c. void foo (int *a, int *b, int *c);
   d. All of the above
   e. None of the above

15. Consider the following struct and main function:

```
struct person{
   char name [100];
   char mid_init;
};

int main (){
   struct person p1;
   func (*p1.name, &p1.mid_init, &p1);
}
```

Which one of the followings is the correct function prototype for function "func":
   a. void func(char*, char*, struct person*);
   b. void func(char, char*, person*);
   c. void func(char, char*, struct person*);
   d. void func(char*, char*, person*);
   e. None of the above

Section II – Short Answer Questions (10 marks)

Part a. (1 marks.) What is the output from the following code fragment?

```
int a[5] = { 2 - 2, 1, 0, 1, -2
}, i;
  for (i = 0; i < 5; i++) {
    if (a[i])
     printf("Y");
   else
    printf("N");
}
```

Answer: NYNYY

Part b. (2 marks.) An executable file is made from the two rather bizarre files below. What is the output of the program?

File main.c	File stuff.h

`#include <stdio.h>`
`int main(void)`
`{`
`#include "stuff.h"`
`#include "stuff.h"`
`  return 0;`
`}`
	`printf("hello!\n");`
	`#ifndef GOODBYE`
	`#define GOODBYE "bye!"`
	`printf("GOODBYE %s\n", GOODBYE);`
	`#endif`

Answer:

```
hello!
GOODBYE bye!
hello!
```

Part c. (2 marks.) What is the output of the program if the size of an int is 4 bytes and the size of a pointer is 8 bytes?

Answer is:

```
main says 5
func says 2
```

What is the output of the program if the sizes of ints and pointers are both 4 bytes

Answer is:

```
main says 5
func says 1
```

```
#include <stdio.h>
void func(int y[5]);
int main(void) {
   int x[5] = {10, 8, 6, 4, 2};
   printf("main says %lu\n",
   sizeof(x)/sizeof(x[0]));
   func(x);
   return 0;
}
void func(int y[5]) {
   printf("func says %lu\n", sizeof(y)/sizeof(y[0]));
}
```

Part d. (2 marks.) What is the output from the following program? For full credit, you must show how you got your answer.

```
#include <stdio.h>
#define MAC1(x) x * x
#define MAC2(y, z) MAC1(y) - MAC1(z)
int main(void) {
 printf("%d\n", MAC2(3 + 1, 2 + 3));
 return 0;
}
```

Answer: 14

3 + 1 * 3 + 1 – 2 + 3 * 2 + 3
3 + 3 + 1 – 2 + 6 + 3 = 14

Part e. (3 marks.) What is the output from the following code fragment? For full credit, you must show how you got your answer.

```
char x[10] = {
 'A', 'B', 'C', '\0', '\0',
 '\0', '\0', '\0', '\0', '\0'
};
strcat(x, "UV");
x[8] = 'W';
printf("%lu %lu %lu %lu\n",
strlen(x), strlen(x+2), strlen(x+5), strlen(x+8));
```

Answer: 5 3 0 1

Array x will b holding: "ABCUV\0\0\0W\0
Therefore:
strlen(x) == 5
strlen(x + 2) == 3
strlen(x + 5) == 0
strlen(x + 8) == 1

Section III – AR diagrams – 15 marks

Part a. Draw a memory diagram for **point one** in the following C++ program (7 marks).

```
int counter;
char * build (const char *z, int n){
    char *p = malloc (n);
    char *s = p;
    while (*z){
        *p++ = *z++;
        counter++;
    }
    *p++ = '\0';
    //--------------------------------------POINT 1
    return s;
}
char * foo (const int *x, double *y, const char *z)
{
    y[0] = 100;
    *(y + 1) = 200;
    return build(z, *x);
}
int main (){
    const int size = 5;
    double a [size] = {11, 22, 33, 44, 55};
    const char *p = foo (&size , a + 3, "ABC");
    printf("%s", p);
}
```

Part b. Consider the following C program, and draw an AR diagram for point one (8 marks).

```
typedef struct Point_s { int x, y; } Point;
typedef struct Circle_s{
    int radius;
    Point* centre;
} Circle;

Point ref = {660, 710};

void doSomething(Circle *x, Circle *y) {
    *x = *y;
    // Point ONE
}

Circle* create_circles(int size){
    Circle* p = calloc(size, sizeof (Circle));
    assert(p != NULL);

    for(int i =0; i<size; i++) {
        p[i].radius = 100 + i;
        p[i].centre = malloc(sizeof(Point));
      assert(p[i].centre != NULL);
       *p[i].centre = ref;
    }
    return p;
}

int main(){
    Point def = {45};
    Circle *arr = create_circles(3);
    *(arr[1].centre) = def;
    doSomething (arr, arr + 2);
    return 0;
}
```

SECTION IV – Function Definitions (22 marks)

Part a (8 marks) – Write a definition for function copy based on given function interface comment.

```
void copy(const char* input_file, const char *output_file);
/* REQUIRES: input_file and output_file point to valid C-strings representing input
 * and output file names.
 * PROMISES: opens two text files (an input and an output file), and copies every
 * content of the input file into the output file with the exception that if any
 * character is an upper case letter will be converted to lower case before writing */
```

Note: You are not allowed to use library functions such as islower, isupper, tolower and toupper in this question

```
void copy(const char* input, const char *output);{
    FILE *fpin, *fpout;

    fpin = fopen(input, "rt");
    if(fpin == NULL) {
        fprintf(stderr, "input file cannot be openee");
        exit(1);
    }

    fpout = fopen(output, "wt");
    if(fpout == NULL) {
        fprintf(stderr, "output file cannot be openee");
        exit(1);
    }
    // reads the first character
    int c;
    c = fgetc(fpin);

    if(ferror(fpin)){
           fprintf(stderr, "read failed ....");
           exit(1);
    }
    //checks if read operation hits end of file
    while(c != EOF) {
        if(c >= 'A' && c <= 'Z' )
            c = c + ('a' - 'A');

        fputc(c, fpout);
        if(ferror(fpout)){
            fprintf(stderr, "write failed ....");
            exit(1);
        }
        // updates the value c by reading a new character
        c = fgetc(fpin);

        if(ferror(fpin)){
            fprintf(stderr, "read failed ....");
            exit(1);
        }
    }
    fclose(fin);
    fclose(fout);
}
```

Part b (6 marks) - Write a the definition of remove_outliers based on given function interface comment.

```
void remove_outliers(int* x, int n, int min, int max);
/* REQUIRES: x points to an integer array with n numbers
 * PROMISES: if any element of x holds a value less than min or greater than max will
 * be removed in place. It means the values to the right of the removed number will
 * be all shifted to the left.
   This line was added during the exam: Returns the number of elements of array x after
                                        removing the outliers.
 */
```

Here are two possible solutions:

```
int remove_outliers(int* a, int n, int min, int max) {
    int j = 0, j = 0;
    while (i < n){
        if(a[i] >= min && a[i] <= max) {
            a[j] = a[i];
            j++;
        }
        i++;
    }
    return j;
}
//--------------------------- SOLUTION TWO ---------------------------------
int remove_outliers1(int* a, int n, int min, int max) {
    int i, j, k=0;
    for(i = 0; i < n; i++){
        if(a[i] < min || a[i] > max){
            for (j = i; j < n-1; j++)
                a[j] = a[j+1];
            i--; // make sure not to skip next number if it is outlier
            k++;
        }
    }
    return i - k;
}
```

Part c. (8 marks) Write the definition of function same_letters based on given function interface comment.

```
int same_letters(const char *s, const char *t);
// REQUIRES: s and t point to beginnings of strings.
// PROMISES:
// Return value is 1 if the two strings contain the same sequence of letters in the
// same order; characters that are not letters are ignored. Otherwise, return value
// is 0.
// EXAMPLES:
// same_letters("ENEL", ".E..N...E....L...") == 1
// same_letters("A B C D", " ABC ") == 0, because there is no match for 'D' in the
// second string.
// same_letters("__A_B_c", ",,A,B,C,,") == 0, because 'c' is not the
// same letter as 'C'.
```

Hint #1: The library function

```
int isalpha(int c);
```

returns a non-zero value if c is the character code for a letter and a zero value otherwise.

Hint #2: Here's some **imprecise** pseudocode for an algorithm:

```
while (1) {
   loop through first string to find a letter or '\0'
   loop through second string to find a letter or '\0'
   if (characters do not match)
      break
   if (characters are both '\0')
      break
 }
```

```
int same_letters(const char *s, const char *t)
{
    int result = 1;
    int si = 0, ti = 0; // indexes into s and t
    while (1) {
        while (s[si] != '\0' && !isalpha(s[si]))
            si++;
        while (t[ti] != '\0' && !isalpha(t[ti]))
            ti++;
        // Change result to 0 if two letters don't match or we're
        // comparing a letter to '\0'.
        if (s[si] != t[ti]) {
            result = 0;
            break;
        }
        // result stays 1 if we're at the ends of both strings.
        if (s[si] == '\0' || t[ti] == '\0')
            break;
        // If we're here, two letters just matched, so we need to step
        // forward in both strings.
        si++;
        ti++;
    }
    return result;
}
```