CIS 241 Winter 2018
Homework 1

Due: Monday, 2/26/2018, by 10:00 pm

1. Given a C source code file "test.c", please give the command to generate an executable named "test". Assume the compiler gcc is used to compile the program. (5 points)

Answer: gcc -o test test.c

2. Please complete the following equations in the context of C programming language. (5 points)

    a. 3/2=?
    b. 5/2.0=?

answer: a. 1; b. 2.5

3. Please list five keywords reserved in C and briefly explain their meanings. (5 points)

Answer: for, while, switch, void, auto, do, long, case, break, union, continue float, etc.

4. Which header file is required to use the functions of "printf()" and "scanf()"? (5 points)

Answer: stdio.h

5. Please indicate whether the following statements with a question mark are allowed or not allowed in ANSI C programs. If not allowed, please explain why (5 points).

    a. int a[10];
        a++; ? //answer: not allowed because an array name is a constant pointer, cannot be changed.

    b. int a[10];
        int *p = a;
        p++; ? //answer: allowed. p is a pointer, can be assigned with any values.

    c. #define N 10
        int a[N]; ? //answer: allowed. N is a constant.

6. What are the outputs when the following code gets executed? Please study the code and understand why those values get printed. (5 points)

```
char *formant = "\n%d\n%d\n%d\n%d\n";
int a[] = {3, 6, 8};
int *p = a;
printf(format, *(p+1), *p + 7, 3 * **&p + 1, 2 * *(p + 1) – 2);
```

answer:
6

10
10
10

7. What get printed when the following code is executed (5 points)? Please study the code to understand the outputs.

```
char str[] = "system", *p;
p = str;
printf("\n%c %s %s %d %d\n", *p, str, p + 1, strlen(str), sizeof(str));
```

answer:
s system ystem 6 7 (size of str is its length plus one)

8. What are the differences between variable n and static variable m in the function below in terms of life time and initialization (5 points)?

```
void foo ( )
{
    int n = 3;
    static int m = 6;
    … …
}
```

answer: n will be initialized whenever foo is called, m is initialized only once when the function foo gets called for the first time. The life time of n is the life time of function foo; while the life time of m is the life time of the whole program that function foo belongs to.

9. What is a union and when would you want to use it in a C program (5 points)?

Answer: A union is a user defined data type that defines a set of alternative values that may be stored in a shared portion of memory. A union is used to save some storage when only one of a set of variables may be used during a program execution.

10. What is the purpose of using the built-in function *free()* in a C program? What types of pointers can be passed in to the function? (5 points)

Answer: it is used to release memory/return allocated memory to the system. Pointers that are returned by malloc(), calloc() and remalloc() functions can be passed in to the function.

11. (15 points) Answer the following questions based on the following code.

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

void swap(int *a, int *b)
{
    int *tmp = a;
    a = b;
    b = tmp;
}

int main()
{
    int a = 6, b = 8;
    printf("Before swap: a = %d, b = %d\n", a, b);
    swap(&a, &b);
    printf("After  swap: a = %d, b = %d\n", a, b);

    return 0;
}
```

1) What gets printed out when the code is executed?

Answer:
Before swap: a = 6, b = 8
After swap: a = 6, b = 8

2) The swap function is intended to swap the two passed in variables. Does it work as expected? If not, why?

Answer:
No. This is because two local variables (a and b) (pointers) get swapped only. This effect won't reflect to the calling environment.

3) How can you modify the swap function to make it work as expected?

Answer:

```c
void swap(int *a, int *b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```

12. (15 points) Given the definition of a structure below, complete the following function definitions

```
typedef struct {
   int num;
   char code[10];
}ITEM;
```

```
//set member num of item to 0 and code to an empty string
void initialize (ITEM * item)
{


answer:
item->num = 0;
strcpy(item->code, "");


}

//print out the data in item on the screen
void display (ITEM item)
{

answer:
printf("num = %d, code = %s\n", item.num, item.code);



}

//store s in code of item and its length in num
void set (ITEM *item, char s[]);
{

answer:
strcpy(item->code, s);
item->num = strlen(s);


}
```

13. (10 points) What are the differences between stack and heap in terms of memory management, more specifically, in terms of memory allocation and release?

    Answer:
    Memory is automatically allocated at compile time in stack and release automatically when a function returns. Memory is allocated dynamically at run time in heap and must be release manually by using the function free().

14. (10 points) Describe an example where using pointer variables are faster with less memory need than using arrays.

Answer:
A good example is ragged character arrays. For example:

```
char a[2][33] = {"abc", "A Baby Crying Daddy Eating Fast."};
```

char *p[2] = {“abc”, “A Baby Crying Daddy Eating Fast.”};

memory use
a: 2 * 33 = 66 bytes
p: 4 + 33 = 37 bytes, counting the memory for two pointer variables (p[0] and p[1]): 8 + 8 = 16. Total is 37 + 16 = 53.
Pointer uses less memory.
To access a[i][j]: The compiler uses a storage mapping function (i * j + j) to access a[i][j]. each access requires one multiplication and one addition.
However, the compiler does not generate code for storage mapping function to access p[i][j], which means that p does its work faster than n.

15.