1121 Computer Programming I Final Writing Exam

Total scores: 100 pts. (The minimum requirement is 60 pts.)

Name: ____________ Student ID: _____________ Dept.: ___________

1. Pointer and Array [20%] (4 points for each)

An array of integers is stored starting at address 0x5678.

```
int arr[] = {74, 66, 14, 87, 94};
/* the address of the first element in the array is 0x5678*/
```

What is the value produced by each of the following expressions? The value will either be an address like **`0x1A2B`** or an integer value inside the array, like 5 or 7 etc. (Note that the space of an integer is 4 byte, and the address is represented in a hexadecimal format)

(a) `arr + 1 = ?`
(b) `arr[2] = ?`
(c) `&arr[2] = ?`
(d) `*arr = ?`
(e) `*(arr + 3) = ?`

Ans:

2. File [5%] (1 points for each)

Please fill in the blanks with the options provided to complete this passage.

(A) file descriptor
(B) operating system
(C) `FILE` structure
(D) open file table
(E) file control block (FCB)

Opening a file returns a pointer to a (1)__________ (defined in <stdio.h>) that contains information used to process the file. This structure includes a (2) __________ —an index into an operating-system array called the (3) __________ Each array element contains a (4) __________ that the (5) __________ uses to administer a particular file.

3. Linked List [20%] (10 points for each)

In this question, you are asked to finish some methods of a link list. The definition of the link list is defined as follows:

```
struct listNode {
    char data; /* the data of the linked list node */
    struct listNode *nextPtr; /* the pointer to the next node */
}
typedef struct listNode ListNode;
ListNode* sPtr /* the pointer to the header of linked list */
```

(a) In your opinion, please explain what a linked list is. (i.e. how it works, implement or what else you know about it)

Ans:

(b) Consider that the definition of linked list is changed to be as follows. Please list a benefit and a drawback of the modification.

```
struct listNode {
    char data; /* the data of the linked list node */
    struct listNode *nextPtr; /* the pointer to the next node */
    struct listNode *prePtr; /* the pointer to the previous node */
}
typedef struct listNode ListNode;
ListNode* sPtr /* the pointer to the header of linked list */
```

4. C struct [25%] (4 points for each)

(a) Please define a structure named `Employee` with the following fields: `name` (string), `id` (integer) and `department` (string).

(b) Write a function that takes an array of `Employee` structs and size of the array and prints the name and department of each employee.

(c) Describe how you would use `struct` in C code to create a linked list. Provide an example struct definition for a linked list node of `Employee` and explain how the nodes are linked together.

(d) Expand the `Employee` struct to include a nested struct named `EmploymentDetails`, which contains the following fields: `hireDate` (string) and `salary`(int).

(e) Please use `typedef` to define a type named `Employee`, as the alias of the above structure `Employee`.

Ans:

5. Bit field [10%]

In our hw09, we were tasked with declaring a struct for a monster's data, adhering to the following constraints:

- N ≤ 10000
- len(Monster name) < 100
- 0 ≤ Monster attack < 10000
- 0 ≤ Monster recovery<10000
- Monster type ∈ {"Water", "Fire", "Earth", "Light", "Dark"}

Now, with our new understanding of bit fields, I need help modifying the struct declaration to include these. Additionally, please use the typedef feature to name the struct "Monster".

Here's a hint to get you started: Since the monster's name is a string, bit fields will not be used for it. You should declare the name within the struct using a character array."

6. String [10%] (2 points for each)

Select the string function that matches the correct description.

(A) `size_t strcspn(const char *s1, const char *s2);`
(B) `size_t strspn(const char *s1, const char *s2);`
(C) `char *strpbrk(const char *s1, const char *s2);`
(D) `char *strstr(const char *s1, const char *s2);`
(E) `char *strtok(char *s1, const char *s2);`

(   ) Determines and returns the length of the initial segment of string s1 consisting only of characters contained in string s2.
(   ) A sequence of calls to this function breaks string s1 into tokens separated by characters contained in string s2. Tokens are logical pieces, such as words in a line of text. The first call uses s1 as the first argument. Subsequent calls to continue tokenizing the same string require NULL as the first argument. Each call returns a pointer to the current token. If there are no more tokens, this function, returns NULL.
(   ) Locates the first occurrence in string s1 of string s2. If the string is found, this function returns a pointer to the string in s1. Otherwise, it returns NULL.
(   ) Determines and returns the length of the initial segment of string s1 consisting of characters not contained in string s2.
(   ) Locates the first occurrence in string s1 of any character in string s2. If a character from s2 is found, this function returns a pointer to that character in s1. Otherwise, it returns NULL.

7. General [10%]

What do you think about this course, including instructor, the lab class, and the TAs? Please write down your suggestions, comments, or anything you would like to say.