

# SE 322 Final Exam

Olcay Taner YILDIZ

## I. QUESTION (15 POINTS)

Write a function that will take the transpose of a matrix  $m$  and returns it.

```
double** transpose(double** m, int numRows,
                  int numCols)
```

*numRows* and *numCols* indicate the number of rows and number of columns of the original matrix respectively.

## II. QUESTION (15 POINTS)

Write the function `strcat` that will concatenates two given strings and returns the new concatenated string.

```
char* strcat(char* string1, char* string2)
```

## III. QUESTION (15 POINTS)

Suppose you have access to a binary file containing a database of phonebook entries. Every entry is stored and processed using the following structure:

```
struct entry {
    char name[32];
    int telephone;
    char address[128];
};
typedef struct entry Entry;
typedef Entry* Entryptr;
```

Implement a function that opens a database at the given **path** and retrieves the first **n** entries having **txt** on its name field. The signature of your function will be:

```
Entryptr searchEntries(char *path, int n,
                      char* txt)
```

You may use the following standard I/O calls in your implementation.

```
FILE* fopen(const char* path,
            const char* type)
size_t fread(void* ptr, size_t size,
             size_t nobj, FILE* fp)
```

## IV. QUESTION (10 POINTS)

`FILE* fp` is a line-buffered stream. Its standard I/O buffer size is exactly 8 bytes. For each of the following operations, on the corresponding row of the table, indicate the contents of the buffer after the operation is carried out successfully. If the buffer is empty after an operation, then fill-in all cells with 'X'. Operations will be carried out in the order given below. This means, each one of your answers affects the next one.

```
fp = fopen(...);
```

```
fputs("abc", fp);
fputs("de\\", fp);
fputs(" gh", fp);
fputs("\nijk", fp);
fputs("lmn\nop", fp);
```

## V. QUESTION (15 POINTS)

- (3 points) Define a structure `struct point` representing a point on the Cartesian plane. Every point has an  $x$ -coordinate and a  $y$ -coordinate.
- (3 points) Define a structure `struct rect` representing a rectangle on the Cartesian plane. Every rectangle has two points: top-left corner, and bottom-right corner.
- (9 points) Write a function that calculates the area of a given `struct rect`. Your function will check if the rectangle is malformed. If so, return -1 as the result. A `struct rect` is malformed if its top-left corner is not strictly above and to the left of its bottom-right corner.

## VI. QUESTION (15 POINTS)

Write a function that searches the file names in the directory named **dir\_name** recursively which start with character **ch**. The signature of your function will be:

```
void print_contents(char* dir_name, char ch)
```

You may use the following functions.

```
int stat(const char *pathname,
         struct stat *restrict buf)
DIR* opendir(const char *pathname)
struct dirent* readdir(DIR *dp)
int closedir(DIR *dp)
```

## VII. QUESTION (15 POINTS)

Write C codes with `fork()` to get the following process trees:

