

SE 322 Bütünleme

Olca Taner YILDIZ

I. QUESTION (15 POINTS)

Consider the following algorithm to generate a sequence of numbers. Start with an integer n . If n is even, divide by 2. If n is odd, multiply by 3 and add 1. Repeat this process with the new value of n , terminating when $n = 1$. For example, the following sequence of numbers will be generated for $n = 12$.

12 6 3 10 5 16 8 4 2 1

Write the method

```
int* create_sequence(int N)
```

which generates the sequence and returns it as an array for the number N .

II. QUESTION (20 POINTS)

Write the function

```
double atof(char *source)
```

which converts the string in `source` to a floating point.

III. QUESTION (20 POINTS)

Write a function that will multiply two matrices m and n and returns it.

```
double** multiply(double** m, double** n,
int numRows, int numCols, int numCols2)
```

`numRows` and `numCols` indicate the number of rows and number of columns of the matrix m respectively. `numCols` and `numCols2` indicate the number of rows and number of columns of the matrix n respectively.

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("abcd", fp);
fputs("def", fp);
fputs("g", fp);
fputs("\nhij", fp);
fputs("lmn\nop", fp);
```

V. QUESTION (20 POINTS)

Given the following C program, draw the process tree.

```
void main(void)
{
  int pid, pid1, pid2, pid3;
  int pid4, pid5, pid6;
  pid = fork();
  if (pid == 0){
    pid2 = fork();
    if (pid2 > 0){
      pid4 = fork();
      if (pid4 > 0)
        pid5 = fork();
    }
  }
  else{
    pid1 = fork();
    if (pid1 == 0){
      pid3 = fork();
      if (pid3 > 0)
        pid6 = fork();
    }
  }
}
```

VI. 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 starting with **s** on its telephone field. The signature of your function will be:

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

You may use file I/O calls in your implementation.