

Midterm Part 2

1. You are given a simple sorting program in C. Convert this program to use F#. The data is in an array which is a mutable structure. You must use F# constructs.

C Program:

```
int data[] = {6, 92, 3, 7, 6, 77, 13, 9, 11, 46, 19, 24, 30, 55, 4};

void swap(int i, int j, int *arr) {
    int tmp = arr[i];
    arr[i] = arr[j];
    arr[j] = tmp;
}

int main() {
    int len = sizeof(data) / sizeof(int);

    // Print unsorted data
    for (int i = 0; i < len; i++)
        printf("%d ", data[i]);

    printf("\n");
    for (int i = len - 1; i >= 0; i--)
        for (int j = 1; j <= i; j++)
            if (data[j-1] > data[j])
                swap((j-1), j, data)

    // Print sorted data
    for (int i = 0; i < len; i++)
        printf("%d ", data[i]);

    return 0;
}
```

F# Program:

```
open System

[<EntryPoint>]
let main argv =
    let data = [|6; 92; 3; 7; 6; 77; 13; 9; 11; 46; 19; 24; 30; 55; 4|]

    printfn "Unsorted array: %A" data

    let swap (i: int) (j: int) (arr: 'a[]) =
        let temp = arr.[i]
        arr.[i] <- arr.[j]
        arr.[j] <- temp

    let sort (arr: 'a[]) =
        for i = arr.Length - 1 downto 0 do
            for j = 1 to i do
                if arr.[j-1] > arr.[j] then
                    swap (j-1) j arr

    sort data

    printfn "Sorted array: %A" data
    0 // return an integer exit code
```

Output:

```
Unsorted array: [|6; 92; 3; 7; 6; 77; 13; 9; 11; 46; 19; 24; 30; 55; 4|]
Sorted array: [|3; 4; 6; 6; 7; 9; 11; 13; 19; 24; 30; 46; 55; 77; 92|]
```

2. Write a program in F# that counts the vowels in a string. The program has a function countVowels with the following signature

```
val countVowels : (string -> int * int * int * int * int)
```

The input to the function is a string and the output is a 5-tuple. The elements of the tuple are counters to keep count of the number of vowels, one for each vowel in the string. The counters are initialized to 0s.

The List module provides functions to convert from/to both sequences and arrays. The function List.ofSeq converts the Sequence to a list.

Example:

```
printfn "%A" (List.ofSeq "The brown fox")
```

produces

```
['T'; 'h'; 'e'; ' '; 'b'; 'r'; 'o'; 'w'; 'n'; ' '; 'f'; 'o'; 'x']
```

The function countVowels uses List.fold to count the number of vowels, as shown.

```
List.fold Func (0, 0, 0, 0, 0) (List.ofSeq str)
```

```
val Func : (int*int*int*int*int->char-> int*int*int*int*int)
```

The function to fold checks the list of characters, matches them with the vowels and update the counters accordingly.

Run this test:

```
countVowels "A bird in hand is worth two in the bush"
```

Output:

```
(2, 1, 4, 2, 1)
```

Complete code:

```
open System

[<EntryPoint>]
let main argv =
    let countChar (tuple: int*int*int*int*int) (x: char): (int*int*int*int*int) =
        match tuple with
        | (a, b, c, d, e) ->
            match x with
            | 'a' | 'A' -> (a+1, b, c, d, e)
            | 'e' | 'E' -> (a, b+1, c, d, e)
            | 'i' | 'I' -> (a, b, c+1, d, e)
            | 'o' | 'O' -> (a, b, c, d+1, e)
            | 'u' | 'U' -> (a, b, c, d, e+1)
            | _ -> (a, b, c, d, e)

    let countVowels (str: string): (int*int*int*int*int) =
        List.fold (countChar) (0, 0, 0, 0, 0) (List.ofSeq str)

    let output = countVowels "A bird in hand is worth two in the bush"

    printfn "%A" output
    0 // return an integer exit code
```

Output:

```
(2, 1, 4, 2, 1)
```