

Lab Assignment 13

Himanshu Gupta

Pull with Git?

1. Write a C function to sort the numbers of an array using Insertion sort.

```
#include <stdio.h>
#include <stdlib.h>

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

void insert_sort(int *arr, const size_t arrSize)
{
    int iter = 1, secIter = iter - 1;
    int key;
    for (iter = 0; iter < arrSize; iter++)
    {
        key = arr[iter];
        for (secIter = iter - 1; secIter >= 0; secIter--)
        {
            if (arr[secIter] >= arr[secIter + 1])
                arr[secIter + 1] = arr[secIter];
            else
                break;
        }
        arr[secIter + 1] = key;
    }
}

int main()
{
    const size_t inbuffSize = 100; //for the input buffer
    int iter = 0;
    const size_t inputArraySize = 5; //size of the array to be sorted
    int inputArray[inputArraySize];
    printf("Enter Array: \n");
    for (iter = 0; iter < inputArraySize; iter++)
    {
        char inbuff[inbuffSize]; //tmp buff to store input
        fgets(inbuff, inbuffSize - 2, stdin); //taking input in buff
        inputArray[iter] = strtol(inbuff, (char **)NULL, 10); //conv. and storing input
    }
    //array is now ready to be sorted
    insert_sort(inputArray, inputArraySize);
    //array is now sorted
    printf("Sorted Array: ");
    for (iter = 0; iter < inputArraySize; iter++)
    {
        printf("%d ", inputArray[iter]);
    }
}
```

```

    }
    printf("\n");
}

```

\$ bin/1.o

Input:

Enter Array: 3

1

2

5

4

Output:

Sorted Array: 1 2 3 4 5

```

#include <stdio.h>

```

```

int fibonacci(int n, int *arr)
{
    if (n == 2 || n == 1)
        arr[n - 1] = 1;
    else if (arr[n - 2] != 0 && arr[n - 3] != 0)
        arr[n-1] = arr[n - 2] + arr[n - 3];
    else
        arr[n - 1] = fibonacci(n - 1, arr) + fibonacci(n - 2, arr);
    return arr[n-1];
}

```

```

int main()
{
    const size_t toPrintSize = 7;
    int toPrint[toPrintSize];
    for (int i = 0; i < toPrintSize; i++)
        toPrint[i] = 0;
    fibonacci(toPrintSize, toPrint);
    for (int i = 0; i < toPrintSize; i++)
    {
        printf("%d ", toPrint[i]);
    }
}

```

\$ bin/2.o

Input:

None.

Output:

1 1 2 3 5 8 13

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

typedef union
{
    int integer;
    float floatingPoint;
    char string[100];
} Union;

typedef struct
{
    int integer;
    float floatingPoint;
    char string[100];
} Struct;

void input(Struct* S , Union* U)
{
    const size_t inputbuffSize = 100;
    char inputbuff[inputbuffSize];
    printf("Input Struct: \n");
    printf("\tEnter Integer: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    S->integer = strtol(inputbuff , (char**)NULL , 10);
    printf("\tEnter Floating-Point Number: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    S->floatingPoint = strtod(inputbuff , (char**)NULL);
    printf("\tEnter String: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    strcpy(S->string , inputbuff);
    printf("Input Union: \n");
    printf("\tEnter Integer: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    U->integer = strtol(inputbuff , (char**)NULL , 10);
    printf("\tEnter Floating-Point Number: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    U->floatingPoint = strtod(inputbuff , (char**)NULL);
    printf("\tEnter String: ");
    fgets(inputbuff , inputbuffSize - 2, stdin);
    strcpy(U->string , inputbuff);
    return;
}

int main()
{
    Struct Struct1;
    Union Union1;
    printf("Struct Stores all the variables separately, while union only stores one.\n");
    printf("Size of Struct: %d bytes,\nSize of Union: %d bytes\n" , sizeof(Struct1) , sizeof(Union1));
    printf("Give input: \n");
    input(&Struct1 , &Union1);
    printf("\nOutput:\n");
    printf("Struct:\n");
    printf("\tInteger: %d\n" , Struct1.integer);
    printf("\tFloating-Point Number: %f\n" , Struct1.floatingPoint);
    printf("\tString: %s" , Struct1.string);
    printf("Union:\n");
    printf("\tInteger: %d\n" , Union1.integer);
    printf("\tFloating-Point Number: %f\n" , Union1.floatingPoint);
    printf("\tString: %s" , Union1.string);
    return 0;
}

```

```
$ bin/3.o
```

Input:

Give input: Input Struct:

Enter Integer: 12
Enter Floating-Point Number: 13.43
Enter String: Hello, World

Input Union:

Enter Integer: 12
Enter Floating-Point Number: 13.43
Enter String: Hello, World

Output:

Struct:

Integer: 12
Floating-Point Number: 13.430000
String: Hello, World

Union:

Integer: 1819043144
Floating-Point Number: 1143139122437582505939828736.000000
String: Hello, World

```
#include <stdio.h>
#include <stdlib.h>

//Buffered
int copy(FILE *to, FILE *from)
{
    const size_t buffSize = 1024;
    char* buffer = (char*)malloc(sizeof(char)*buffSize);
    char ch;
    int count = 0;
    int iterations = 0;
    while (1)
    {
        if (count >= buffSize)
        {
            fwrite((void *)buffer, 1024, 1, to);
            count = 0;
        }
        buffer[count] = getc(from);
        if (buffer[count] == EOF)
            break;
        count++;
    }
    fwrite((void *)buffer, count, 1, to);
    printf("Total %d bytes read and written" , iterations*buffSize+count);
}

//Unbuffered
/*
int copy(FILE *to, FILE *from)
{
    char ch = getc(from);
```

```

int count = 0;
while (ch != EOF)
{
    putc(ch , to);
    count++;
    ch = getc(from);
}
printf("%d Bytes Written" , count);
}
*/

int main()
{
    FILE *from = fopen("quotes_from_stormlight.txt", "r");
    FILE *to = fopen("copy.txt", "w");
    if(from == (FILE*)NULL || to == (FILE*)NULL)
    {
        printf("Cannot Open File for reading or writing");
        return 0;
    }
    copy(to, from);
    fclose(to);
    fclose(from);
    return 0;
}

```

\$ bin/4.o

Input:

None.

Output: Total 746 bytes read and written

\$ cat quotes_from_stormlight.txt

“Somebody has to start. Somebody has to step forward and do what is right, because it is right. If nobody starts, then others cannot follow.” -Lirin, tWoK, Chapter 3

“Two blind men waited at the end of an era, contemplating beauty. They sat atop the world’s highest cliff, overlooking the land and seeing nothing.” -The Messenger, WoR, Chapter 45

“And so, does the destination matter? Or is it the path we take? I declare that no accomplishment has substance nearly as great as the road used to achieve it. We are not creatures of destinations. It is the journey that shapes us. Our callused feet, our backs strong from carrying the weight of our travels, our eyes open with the fresh delight of experiences lived.” -Dalinar, tWoK, Chapter 48

\$ cat copy.txt

“Somebody has to start. Somebody has to step forward and do what is right, because it is right. If nobody starts, then others cannot follow.” -Lirin, tWoK, Chapter 3

“Two blind men waited at the end of an era, contemplating beauty. They sat atop the world’s highest cliff, overlooking the land and seeing nothing.” -The Messenger, WoR, Chapter 45

“And so, does the destination matter? Or is it the path we take? I declare that no accomplishment has substance nearly as great as the road used to achieve it. We are not creatures of destinations. It is the journey that shapes us. Our callused feet, our backs strong from carrying the weight of our travels, our eyes open with the fresh delight of experiences lived.” -Dalinar, tWoK, Chapter 48

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc , char** argv)
{
    int iter = 0;
    int sum = 0;
    for(iter = 1; iter < argc; iter++)
    {
        sum += strtol(argv[iter] , (char**)NULL , 10);
    }
    printf("Sum is %d." , sum);
}
```

\$ bin/5.o 1 2 3 100

Input:

None.

Output:

Sum is 106.