

Project 1:

//Toni Hunter 187009925

//ASSIGNMENT 4 PROJECT 1

//

// Take two strings from user (no larger than 20 characters), pass the strings as pointers
// to the concatenate function.

//

#include <stdio.h>

void Catfun(char*, char*, char*); //function prototype, character pointer is argument type

int main(void)

{

char str1[21] = {0}; //character named str1, 11 makes space for \0

char str2[21] = {0};

char fin[41] = {0};

printf("Please enter your first string: \n"); //user input

scanf("%20s", str1);

printf("Please enter your second string: \n");

scanf("%20s", str2);

Catfun(str1, str2, fin);

printf("Your concatenated string is: %s\n", fin);

}

//DEREFRENCING *: to retrieve the value from the memory address that is pointed by the pointer

//character pointer named p1 is set to the address of str1

void Catfun(char*p1, char*p2, char*f3)

{

while(*p1 != '\0') //dereference increments through address of str1 to get to null char

{

(f3++)=(p1++);

}

//p1 is pointing to \0 at the end of this while loop

// once the end of address one is reached, values in p2's address are stored in p1's address

//this is done by continuing to increment through p1 to elongate it and incrementing through

//p2. p2's values are stored in p1.

while(*p2 != '\0')

{

(f3++)=(p2++);

}

*f3 = '\0'; //the while loop terminates at null of p2 so null char is added to the end of p1

}

Project 2:

```
//Toni Hunter 187009925
//Assignment 4 Project 2
//
// Card shuffling and dealing modification -- create high performance shuffling algorithm that
// avoids indefinite postponement(). Modify shuffle function to loop row by row and column by column
// through the array.
//
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// prototypes
//void shuffle(int d[4][13]); // modified shuffle
void printArray(int d[4][13]);

int main(void){
int ndeck[4][13] = { //ROW||, COLUMN--
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13},
    {14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26},
    {27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39},
    {40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52}};

    puts("Below is the UNSHUFFLED deck: \n");
    printArray(ndeck);
    shuffle(ndeck);
    puts(" ");
    puts("\nBelow the deck has been SHUFFLED 3 times: \n");
    printArray(ndeck);
    shuffle(ndeck);
    printf("\n");
    printArray(ndeck);

}

void printArray(int d[4][13])
{
    int k = 0;
    printf("   0  1  2  3  4  5  6  7  8  9 10 11 12\n");
    printf("   -----\n");
    // loop through rows
    for (int i = 0; i <= 3; i++) {
        printf("%d|", k++);

        // output column values
```

```
    for (int j = 0; j <= 12; j++) {  
        printf("%4d", d[i][j]);  
    }  
  
    printf("\n"); // start new line of output  
}  
}
```

```
// shuffle cards in deck  
void shuffle(int d[4][13])  
{  
    int row, col;  
  
    for (int i = 0; i < 4; i++)  
    {  
        for (int j = 0; j < 13; j++)  
        {  
            int e = d[i][j]; //takes a element  
            row = rand() % 4; //gets random row  
            col = rand() % 13; // random column  
            d[i][j] = d[row][col]; //moves the element  
            d[row][col] = e; //sets the new number equal to element  
        }  
    }  
}
```