

Pull with git

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

const unsigned int STRINGS_NUMBER = 5;
const size_t MAX_STRING_SIZE = 30;

char* largest_string(char(*)[MAX_STRING_SIZE]);

int main()
{
    int i = 0;
    char stringArray[STRINGS_NUMBER][MAX_STRING_SIZE];
    printf("WARNING: MAX BUFFER SIZE IS SET TO %d. ANY STRING CONTAINING MORE THAN
%d CHARACTERS WILL CAUSE AN OVERFLOW.\n" , MAX_STRING_SIZE , MAX_STRING_SIZE - 1);
    printf("Input %d newline terminated strings: \n" , STRINGS_NUMBER);
    for(i = 0 ; i < STRINGS_NUMBER; i++)
        scanf("%s" , *(stringArray + i));    //fgets would be safer and better.
    printf(largest_string(stringArray));
}

char* largest_string(char strArr[][MAX_STRING_SIZE])
{
    char* tmp = strArr[0];
    int i = 0;
    for(i = 0 ; i < STRINGS_NUMBER; i++)
    {
        tmp = strlen(tmp) < strlen(strArr[i]) ? strArr[i] : tmp;
    }
    return tmp;
}

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

const unsigned int STRINGS_NUMBER = 5;
const size_t STRING_MAX_SIZE = 30;

char *input(void);
int string_size(char *);
char *longest_string(char *stringArray[]);
void largest_str(char strings[STRINGS_NUMBER][STRING_MAX_SIZE]);

int main()
{
    int i = 0;
    char strings[STRINGS_NUMBER][STRING_MAX_SIZE];
    char *stringArray[STRINGS_NUMBER];
    // for (i = 0; i < STRINGS_NUMBER; i++)
    {
        // stringArray[i] = input();
    }
    largest_str(strings);
    stringArray[i] = (char*)NULL;
    printf(longest_string(stringArray));
}
```

```

char *input(void)
{
    char* buff = (char*)malloc(30);
    printf("Input String (Warning Max string length is 29 characters): ");
    scanf("%s", buff);
    return buff;
}

char *longest_string(char *stringArray[])
{
    int i = 0;
    char *largest = stringArray[0];
    for (i = 1; stringArray[i] != (char*)NULL; i++)
    {
        if (strlen(largest) < strlen(stringArray[i]))
            largest = stringArray[i];
    }
    return largest;
}

void largest_str(char strings[STRINGS_NUMBER][STRING_MAX_SIZE])
{
    int i = 0;
    for(i = 0; i < STRINGS_NUMBER; i++)
        scanf("%s" , strings[i]);
}

#include <stdio.h>

void val_swap(int , int );
void ref_swap(int* , int*);

int main()
{
    int a , b;
    scanf("%d %d" , &a , &b);
    printf("Not Swapped: %d %d \n" , a , b);
    val_swap(a , b);
    ref_swap(&a , &b);
    printf("Not Swapped: %d %d \n" , a , b);
}

void val_swap(int num1, int num2)
{
    printf("Val Swap: %d %d \n" , num2 , num1);
}

void ref_swap(int* ptr1 , int* ptr2)
{
    int tmp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = tmp;
}

#include <stdio.h>
int digit_sum(int );

int main()
{
    int a;
    scanf("%d" , &a);
    printf("Sum of digits %d" , digit_sum(a));
}

```

```

}

int digit_sum(int num)
{
    int result = 0;
    while(num != 0)
    {
        result += num%10;
        num /= 10;
    }
    return result;
}
#include <stdio.h>
void leap_check(int year);

int main()
{
    int in_year;
    scanf("%d" , &in_year);
    leap_check(in_year);
}

void leap_check(int year)
{
    if(year%4 == 0 && year%100 != 0 || year%400 == 0)
    {
        printf("Leap Year");
        return;
    }
    printf("Not Leap Year");
}
#include <stdio.h>
const int arr1_x = 2 , arr1_y = 2 , arr2_x = 2 , arr2_y = 2;
const int result_x = arr1_x , result_y = arr2_y;

void multiply(int arr_a[arr1_x][arr1_y] , int arr_b[arr2_x][arr2_y] , int
result[result_x][result_y])
{
    int iter1 , iter3, iter2;
    int temp = 0;
    for(iter1 = 0; iter1 < result_x; iter1++)
    {
        for(iter2 = 0; iter2 <= result_y; iter2++)
        {
            temp = 0;
            for(iter3 = 0; iter3 < arr1_y; iter3++)
                temp += (arr_a[iter1][iter3] * arr_b[iter3][iter2]);
            result[iter1][iter2] = temp;
        }
    }
}

int main()
{
    if(arr1_y != arr2_x)
    {
        printf("Dimensions of Matrices not suitable for multiplication.");
        return 0;
    }
    int arr1[arr1_x][arr1_y], arr2[arr2_x][arr2_y] , result[result_x][result_y];
    int iter1 = 0, iter2 = 0;

```

```

printf("Input First Matrix.\n");
for(iter1 = 0; iter1 < arr1_x; iter1++)
{
    printf("*****d Row*****\n" , iter1 + 1);
    for(iter2 = 0; iter2 < arr1_y; iter2++)
    {
        printf("Enter Element No. %d: " , iter2 + 1);
        scanf("%d" , &arr1[iter1][iter2]);
    }
}
printf("Input Second Matrix.\n");
for(iter1 = 0; iter1 < arr2_x; iter1++)
{
    printf("*****d Row*****\n" , iter1 + 1);
    for(iter2 = 0; iter2 < arr2_y; iter2++)
    {
        printf("Enter Element No. %d: " , iter2 + 1);
        scanf("%d" , &arr2[iter1][iter2]);
    }
}
multiply(arr1 , arr2 , result);

for(iter1 = 0; iter1 < result_x; iter1++)
{
    for(iter2 = 0; iter2 < result_y; iter2++)
    {
        printf("%d\t" , result[iter1][iter2]);
    }
    printf("\n");
}
}

```