Homework 3

Array & String

Example 1 of Multidimensional Array In C

Write a C program to find sum of two matrix of order 2*2 using multidimensional arrays where, elements of matrix are entered by user.

```c
#include<stdio.h>
int main() {

	float matrix_1[2][2] = { };
	float matrix_2[2][2] = { };
	float SumMatrix[2][2] = { };

	printf("Enter the elements of 1st matrix\n");
	for (int i = 0; i < 2; i++) {
		for (int j = 0; j < 2; j++) {
			printf("Enter matrix_1[%d][%d]:", i, j);
			fflush(stdout);
			scanf("%f", &matrix_1[i][j]);
		}
	}


	printf("\nEnter the elements of 2nd matrix\n");
	for (int i = 0; i < 2; i++) {
		for (int j = 0; j < 2; j++) {
			printf("Enter matrix_2[%d][%d]:", i, j);
			fflush(stdout);
			scanf("%f", &matrix_2[i][j]);
		}
	}


	int j = 0;
	for (int i = 0; i < 2; i++) {
		for (j = 0; j < 2; j++) {

			SumMatrix[i][j] = matrix_2[i][j] + matrix_1[i][j];
		}
	}


	printf("\nSum of Matrix:\n");
	for (int i = 0; i < 2; i++) {
		for (j = 0; j < 1; j++) {

			printf("%0.2f    %0.2f\n", SumMatrix[i][j], SumMatrix[i][j + 1]);
		}
	}

	return 0;
}
```

matrix_1[0][0]	2	0x61fef4
matrix_1[0][1]	0.5	0x61FEF8
matrix_1[1][0]	-1.1	0x61FEFC
matrix_1[1][1]	2	0x61FF00

matrix_2[0][0]	0.2	0x61fee4
matrix_2[0][1]	0	0x61FEE8
matrix_2[1][0]	0.23	0x61FEEC
matrix_2[1][1]	23	0x61FEF0

test cases

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah
Enter the elements of 1st matrix
Enter matrix_1[0][0]:2
Enter matrix_1[0][1]:.5
Enter matrix_1[1][0]:-1.1
Enter matrix_1[1][1]:2

Enter the elements of 2nd matrix
Enter matrix_2[0][0]:.2
Enter matrix_2[0][1]:0
Enter matrix_2[1][0]:.23
Enter matrix_2[1][1]:23

Sum of Matrix:
2.20    0.50
-0.87    25.00
```

SumMatrix[0][0]	2.20	0x61fed4
SumMatrix[0][1]	0.50	0x61FED8
SumMatrix[1][0]	-0.87	0x61FEDC
SumMatrix[1][1]	25.0	0x61FEE0

Example 2 C Program to Calculate Average Using Arrays

This program takes N number of elements from user (where, n is specified by user), stores data in an array and calculates the average of those numbers.

```c
#include<stdio.h>
int main() {
	int N;
	float sum=0;
	printf("Enter the numbers of data:");
	fflush(stdout);
	scanf("%d",&N);

	float arr[N];
	for(int i=0;i<N;i++){
		printf("Enter number:");
		fflush(stdout);
		scanf("%f",&arr[i]);
		sum+=arr[i];
	}

	printf("sum = %0.2f\n",sum);
	printf("Average = %0.2f",(sum/N));
	return 0;
}
```

test cases

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Gh
Enter the numbers of data:6
Enter number:45.3
Enter number:45.6
Enter number:33
Enter number:20.34
Enter number:-45.6
Enter number:67.5
sum = 166.14
Average = 27.69
```

إذا لم يعطي الكومبيلر أي خطأ عند استخدام مصفوفة ذات حجم متغير (Variable Length Array - VLA) كما في الكود الأصلي، فهذا يعني أنك تستخدم كومبيلر يدعم C99 أو نسخة أحدث حيث تُقبل VLAs.

في C99 ونسخ أحدث من لغة C، يمكنك استخدام حجم مصفوفة يعتمد على قيمة مدخلة من قبل المستخدم (حجم متغير) كما فعلت في الكود الأصلي. وبالتالي، الكود الذي قدمته سيعمل بشكل صحيح في هذه النسخ من لغة C.

إليك توضيح إضافي حول هذا الموضوع:

VLAs في C99 وأحدث:

في C99، أصبحت مصفوفات ذات حجم متغير (VLAs) قانونية. لذا، الكود الذي قمت بكتابته يمكن أن يُترجم ويعمل بشكل صحيح إذا كان الكومبيلر يدعم C99 أو نسخة أحدث.

لا تحتاج إلى استخدام `malloc` في هذه الحالة إذا كنت تعمل ضمن مواصفات C99 أو أحدث.

VLAs في C89/C90:

إذا كنت تعمل ضمن معايير C89/C90، فلا يمكنك استخدام VLAs. في هذه الحالة، ستحتاج إلى استخدام `malloc` كما أوضحت في الكود المعدل.

تنفيذ الكود:

في الكود الأصلي الذي قدمته، يتم تخصيص الذاكرة للمصفوفة `arr` باستخدام حجم تم تحديده من قبل المستخدم، وهذا مدعوم في C99 والأحدث.

تأكد من أنك تستخدم الكومبيلر المناسب الذي يدعم C99 أو أحدث إذا كنت تعتمد على هذا النمط.

إذا كان الكود يعمل بشكل صحيح في بيئتك الحالية، فهذا يعني أن الكومبيلر الذي تستخدمه يدعم C99 أو أحدث.

Example 3 C Program to Find Transpose of a Matrix

This program asks user to enter a matrix (size of matrix is specified by user) and this program finds the transpose of that matrix and displays it.

```c
#include <stdio.h>

int main() {
    int rows;
    int columns;

    printf("Enter rows and columns of matrix: ");
    fflush(stdout);
    scanf("%d %d", &rows, &columns);

    int matrix[rows][columns];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            printf("Enter element matrix[%d][%d]: ", i, j);
            fflush(stdout);
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("\nEntered Matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }

    int matrix_T[columns][rows];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            matrix_T[j][i] = matrix[i][j];
        }
    }


    printf("\nTranspose of Matrix:\n");
    for (int i = 0; i < columns; i++) {
        for (int j = 0; j < rows; j++) {
            printf("%d\t", matrix_T[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

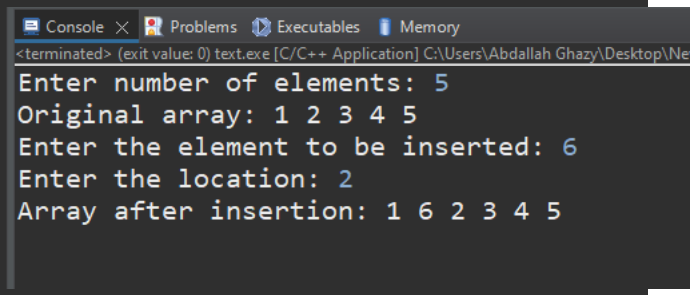
test cases

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Deskt
Enter rows and columns of matrix: 5 5
Enter element matrix[0][0]: 1
Enter element matrix[0][1]: 2
Enter element matrix[0][2]: 3
Enter element matrix[0][3]: 4
Enter element matrix[0][4]: 5
Enter element matrix[1][0]: 6
Enter element matrix[1][1]: 7
Enter element matrix[1][2]: 8
Enter element matrix[1][3]: 9
Enter element matrix[1][4]: 10
Enter element matrix[2][0]: 11
Enter element matrix[2][1]: 12
Enter element matrix[2][2]: 13
Enter element matrix[2][3]: 14
Enter element matrix[2][4]: 15
Enter element matrix[3][0]: 16
Enter element matrix[3][1]: 17
Enter element matrix[3][2]: 18
Enter element matrix[3][3]: 19
Enter element matrix[3][4]: 20
Enter element matrix[4][0]: 0
Enter element matrix[4][1]: 0
Enter element matrix[4][2]: 0
Enter element matrix[4][3]: 0
Enter element matrix[4][4]: 0

Entered Matrix:
1       2       3       4       5
6       7       8       9       10
11      12      13      14      15
16      17      18      19      20
0       0       0       0       0

Transpose of Matrix:
1       6       11      16      0
2       7       12      17      0
3       8       13      18      0
4       9       14      19      0
5       10      15      20      0
```

Example 4 C Program to Insert an element in an Array

```c
#include <stdio.h>

int main() {
    int N;
    int element;
    int location;

    printf("Enter number of elements: ");
    fflush(stdout);
    scanf("%d", &N);

    int arr1[N];
    for (int i = 0; i < N; i++) {
        arr1[i] = i + 1;
    }

    printf("Original array: ");
    for (int i = 0; i < N; i++) {
        printf("%d ", arr1[i]);
    }
    printf("\n");

    printf("Enter the element to be inserted: ");
    fflush(stdout);
    scanf("%d", &element);

    printf("Enter the location: ");
    fflush(stdout);
    scanf("%d", &location);

    if (location < 1 || location > N + 1) {
        printf("Invalid location!\n");
        return 1;
    }

    int arr2[N + 1];
    for (int i = 0, j = 0; i < N + 1; i++) {
        if (i == location - 1) {
            arr2[i] = element;
        } else {
            arr2[i] = arr1[j++];
        }
    }

    printf("Array after insertion: ");
    for (int i = 0; i < N + 1;  i++) {
        printf("%d ", arr2[i]);
    }
    printf("\n");

    return 0;
}
```

test cases



Example 5 C Program to Search an element in Array

```c
#include <stdio.h>

int main() {
	int NElements;
	int SElements;
	int flag = 0;
	int location = 0;

	printf("Enter no of elements :");
	fflush(stdout);
	scanf("%d", &NElements);

	int arr[NElements];
	for (int i = 0, j = 1; i < NElements; i++, j++) {
		arr[i] = (j * 10) + j;
		printf("%d\t", arr[i]);
	}

	printf("\nEnter the elements to be searched :");
	fflush(stdout);
	scanf("%d", &SElements);

	for (int i = 0; i < NElements; i++) {
		if (SElements == arr[i]) {
			flag = 1;
			location = i + 1;

			break;
		}
	}

	if (flag == 1) {
		printf("Number found at the location = %d", location);
	} else {
		printf("\nNumber %d is not available", SElements);
	}

	return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder (2)\
Enter no of elements :6
11	22	33	44	55	66
Enter the elements to be searched :77

Number 77 is not available
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New
Enter no of elements :5
11	22	33	44	55
Enter the elements to be searched :44
Number found at the location = 4
```

test cases

Example 6 C Program to Find the Frequency of Characters in a String

This program asks user to enter a string and a character and this program checks how many times that character is repeated in the string entered by user.

```c
#include <stdio.h>

int main() {
	char string[100];
	char character;
	char counter = 0;
	printf("Enter a string: ");
	fflush(stdout);
	gets(string);
	printf("Enter a character to find frequency: ");
	fflush(stdout);
	scanf("%c", &character);

	for (int i = 0; i < sizeof(string); i++) {
		if (character == string[i]) {
			counter++;
		}
	}

	if (counter) {
		printf("Frequency of %c = %d", character, counter);
	} else {
		printf("The character is missing.");
	}
	return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\
Enter a string: This website is awesome.
Enter a character to find frequency: 4
The character is missing.
```

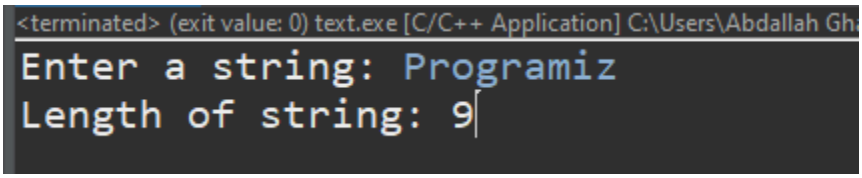
test cases

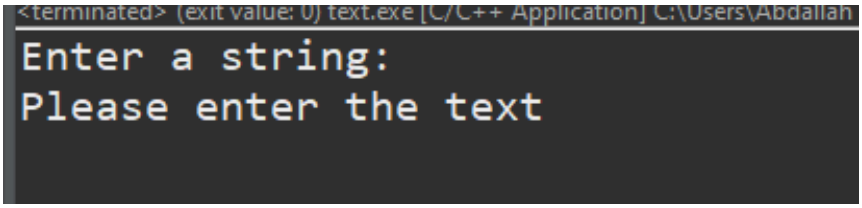
```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\Ne
Enter a string: This website is awesome.
Enter a character to find frequency: e
Frequency of e = 4
```

Example 7 C Program to Find the Length of a String

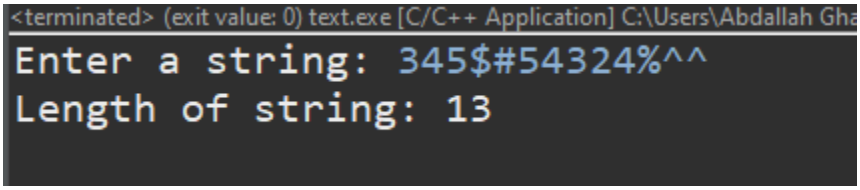
You can use standard library function strlen() to find the length of a string but, this program computes the length of a string manually without using strlen() function.

```c
#include <stdio.h>

int main() {
	char string[100];
	char counter = 0;

	printf("Enter a string: ");
	fflush(stdout);
	gets(string);

	for (int i = 0; string[i] != 0; i++) {
		counter++;
	}

	if (counter) {
		printf("Length of string: %d", counter);
	} else {
		printf("Please enter the text");
	}
	return 0;
}
```





test cases



EX3: C Program to Reverse String Without Using Library Function

You can only use library function strlen(), To find the length of the string

```c
#include <stdio.h>

int main() {
	char string[100];
	char counter = 0;

	printf("Enter a string: ");
	fflush(stdout);
	gets(string);

	for (int i = 0; string[i] != 0; i++) {
		counter++;
	}

	char Rstring[counter];
	for (int i = counter - 1, j = 0; i >= 0; i--, j++) {
		Rstring[j] = string[i];
	}
	Rstring[counter] = 0;

	if (counter) {
		printf("Reverse string is : %s", Rstring);
	} else {
		printf("Please enter the text");
	}
	return 0;
}
```

```
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abda
Enter a string: Pritesh
Reverse string is : hsetirP
```

```
Console  Problems  Executables  Memory
<terminated> (exit value: 0) text.exe [C/C++ Application] C:\Users\Abdallah Ghazy\Desktop\New folder
Enter a string: Please enter the text
Reverse string is : txet eht retne esaelP
```

test cases