

- Calculate average of each column in a 2D Array.

The screenshot shows a C program in a code editor and its execution in a terminal window. The program calculates the average of each column in a 2D array.

```
main.c X main.c X main.c X
3
4
5
6 int main() {
7     int rows, columns;
8     printf("Enter the number of rows and column: ");
9     scanf("%d %d", &rows, &columns);
10    int array[rows][columns];
11    printf("Enter elements of the array:\n");
12    for (int i = 0; i < rows; i++) {
13        for (int j = 0; j < columns; j++) {
14            scanf("%d", &array[i][j]);
15        }
16    }
17    int columnSum[columns];
18    for (int j = 0; j < columns; j++) {
19        columnSum[j] = 0;
20        for (int i = 0; i < rows; i++) {
21            columnSum[j] += array[i][j];
22        }
23    }
24    printf("Average of each column:\n");
25    for (int j = 0; j < columns; j++) {
26        double columnAverage = (double)columnSum[j] / rows;
27        printf("Column %d: %.1f\n", j + 1, columnAverage);
28    }
29
30    return 0;
31 }
32
33
```

Terminal Output:

```
"D:\ITI 2023_2024\C Programming\labs\lab_4\bin\Debug\lab_4.exe"
Enter the number of rows and column: 2 3
Enter elements of the array:
2
4
6
2
2
2
Average of each column:
Column 1: 2.000000
Column 2: 3.000000
Column 3: 4.000000

Process returned 0 (0x0)   execution time : 8.307 s
Press any key to continue.
```

- C Program to add two matrix and put the result in a third one, then print the result.

The screenshot shows a C program in a code editor and its execution in a terminal window. The program adds two matrices and prints the result.

```
main.c X main.c X main.c X main.c X
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int rows, columns;
6     printf("Enter the number of rows and columns for the matrices: ");
7     scanf("%d %d", &rows, &columns);
8     int matrix1[rows][columns], matrix2[rows][columns], result[rows][columns];
9     printf("Enter elements of the first matrix:\n");
10    for (int i = 0; i < rows; i++) {
11        for (int j = 0; j < columns; j++) {
12            scanf("%d", &matrix1[i][j]);
13        }
14    }
15    printf("Enter elements of the second matrix:\n");
16    for (int i = 0; i < rows; i++) {
17        for (int j = 0; j < columns; j++) {
18            scanf("%d", &matrix2[i][j]);
19        }
20    }
21    for (int i = 0; i < rows; i++) {
22        for (int j = 0; j < columns; j++) {
23            result[i][j] = matrix1[i][j] + matrix2[i][j];
24        }
25    }
26    for (int i = 0; i < rows; i++) {
27        for (int j = 0; j < columns; j++) {
28            printf("%d ", result[i][j]);
29        }
30        printf("\n");
31    }
32 }
33
```

Terminal Output:

```
"D:\ITI 2023_2024\C Programming\labs\lab_4Q2\bin\Debug\l...
Enter the number of rows and columns for the matrices: 2 3
Enter elements of the first matrix:
2
3
2
6
4
1
Enter elements of the second matrix:
1
2
3
4
2
1
3 5 5
10 6 2

Process returned 0 (0x0)   execution time : 16.089 s
Press any key to continue.
```

- Receive character by character and then place the string terminator upon pressing enter, then display the string.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      char array[100];
6      printf("Please enter a statement: ");
7      scanf("%99[^\n]", array);
8      printf("The statement is: %s\n", array);
9
10     return 0;
11 }
12

```

Execution output:

```

Please enter a statement: eng mina is big boss
The statement is: eng mina is big boss

Process returned 0 (0x0)   execution time : 9.223 s
Press any key to continue.

```

- C Program to take char and displays if it is "Normal or Extended key" along with its Ascii value.
- Highlight Menu: New Colored Menu "with Arrows"

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <windows.h>
4  #include <dos.h>
5  #include <dir.h>
6
7  #define right_key 77
8  #define left_key 75
9  #define right_key 77
10 #define up_key 72
11 #define down_key 80
12 #define enter_key 13
13
14
15 void SetColor(int ForgC) {
16     HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
17     WORD wColor;
18     CONSOLE_SCREEN_BUFFER_INFO csbi;
19
20     if (GetConsoleScreenBufferInfo(hStdOut, &csbi)) {
21         wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);
22         SetConsoleTextAttribute(hStdOut, wColor);
23     }
24 } //to change the color red or white
25
26 void gotoxy(int x, int y) {
27     COORD coord = {0, 0};
28     coord.X = x; // col
29     coord.Y = y; // row
30     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
31 } // to set row and column
32

```

Execution output:

```

New
Display
Exit

```

```

main.c X main.c X main.c X main.c X main.c X main.c X
33 int main() {
34     char character;
35     int position = 1; // my start position
36     int totalOptions = 3; // three items that i put in menu
37
38     while (1) {
39         for (int i = 1; i <= totalOptions; i++) { // to print options
40             gotoxy(10, 10 + i); // to put them one below other 10 col 11 row
41             if (i == position) { //check position to color text
42                 SetColor(4); // Red
43             } else {
44                 SetColor(15); // White
45             }
46
47             switch (i) { //display menu items
48                 case 1:
49                     printf("New \n \n ");
50                     break;
51                 case 2:
52                     printf("Display \n \n ");
53                     break;
54                 case 3:
55                     printf("Exit \n \n ");
56                     break;
57             }
58         }
59
60         character = getch(); // to know the extended characters up down and so on
61         if (character == -32) {
62             char character1 = getch();
63             switch (character1) {
64                 case up_key:
65                     if (position > 1) { // to go up we will decrement the position
66                         position--;
67                     }
68                     break;
69                 case down_key:
70                     if (position < totalOptions) { // to go down we will increment
71                         position++;
72                     }
73                     break;
74             }
75         } else if (character == enter_key) { // to go to another screen
76             system("cls");
77             switch (position) {
78                 case 1:
79                     gotoxy(5, 10);
80                     printf("New");
81                     break;
82                 case 2:
83                     gotoxy(5, 10);
84                     printf("Display");
85                     break;
86                 case 3:
87                     gotoxy(5, 10);
88                     printf("Exit");
89                     break;
90             }
91             printf("\n\n");
92         }
93     }
94 }

```

"D:\ITI 2023\_2024\C Programming\labs\lab\_4Q4\bin\Debug\lab\_4Q4.exe"

```

New
Display
Exit

```

```

main.c X main.c X main.c X main.c X main.c X main.c X
60 character = getch(); // to know the extended characters up down and so on
61 if (character == -32) {
62     char character1 = getch();
63     switch (character1) {
64         case up_key:
65             if (position > 1) { // to go up we will decrement the position
66                 position--;
67             }
68             break;
69         case down_key:
70             if (position < totalOptions) { // to go down we will increment
71                 position++;
72             }
73             break;
74     }
75 } else if (character == enter_key) { // to go to another screen
76     system("cls");
77     switch (position) {
78         case 1:
79             gotoxy(5, 10);
80             printf("New");
81             break;
82         case 2:
83             gotoxy(5, 10);
84             printf("Display");
85             break;
86         case 3:
87             gotoxy(5, 10);
88             printf("Exit");
89             break;
90     }
91     printf("\n\n");
92 }
93 }
94 }

```

"D:\ITI 2023\_2024\C Programming\labs\lab\_4Q4\bin\Debug\lab\_4Q4.exe"

```

Display
Process returned 0 (0x0)   execution time : 74.46 s
Press any key to continue.

```

- C Program to take firstName and lastName from user, then displays fullName.

```

main.c X main.c X main.c X main.c X main.c X main.c X
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char firstname[10];
7     char lastname[10];
8
9     printf("enter your first name: ");
10    scanf("%s", firstname);
11    printf("enter your last name: ");
12    scanf("%s", lastname);
13    printf("the full name is %s %s.\n", firstname, lastname);
14    return(0);
15 }
16

```

"D:\ITI 2023\_2024\C Programming\labs\lab\_4Q5\bin\Debug\lab\_4Q5.exe"

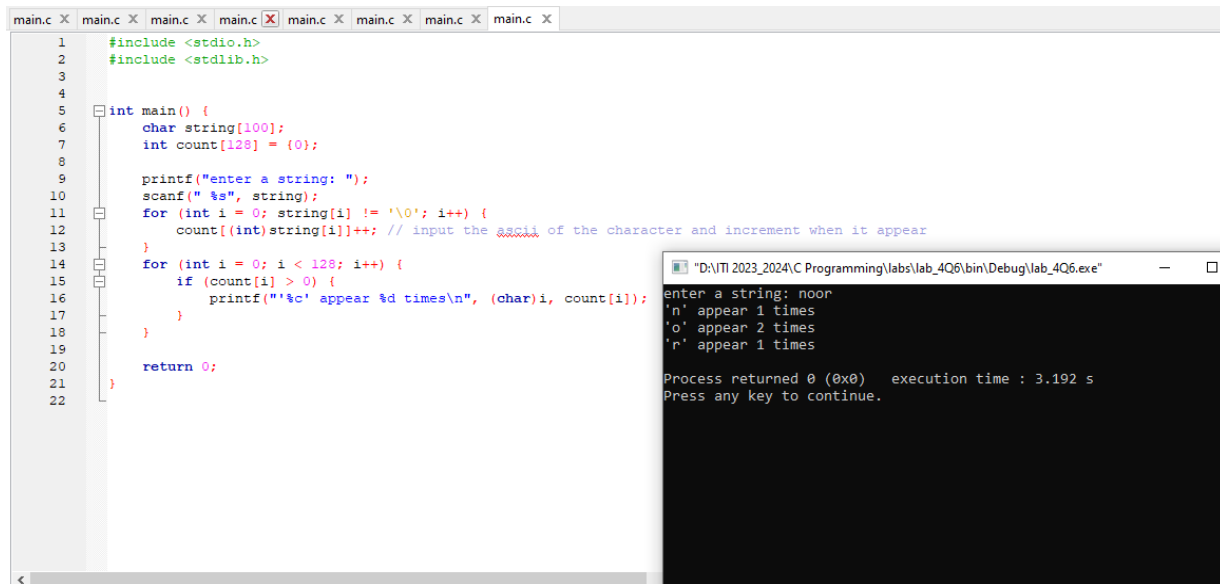
```

enter your first name: nouran
enter your last name: tarek
the full name is nouran tarek.

Process returned 0 (0x0)   execution time : 6.598 s
Press any key to continue.

```

### - C Program to Find the Frequency of Characters in a String



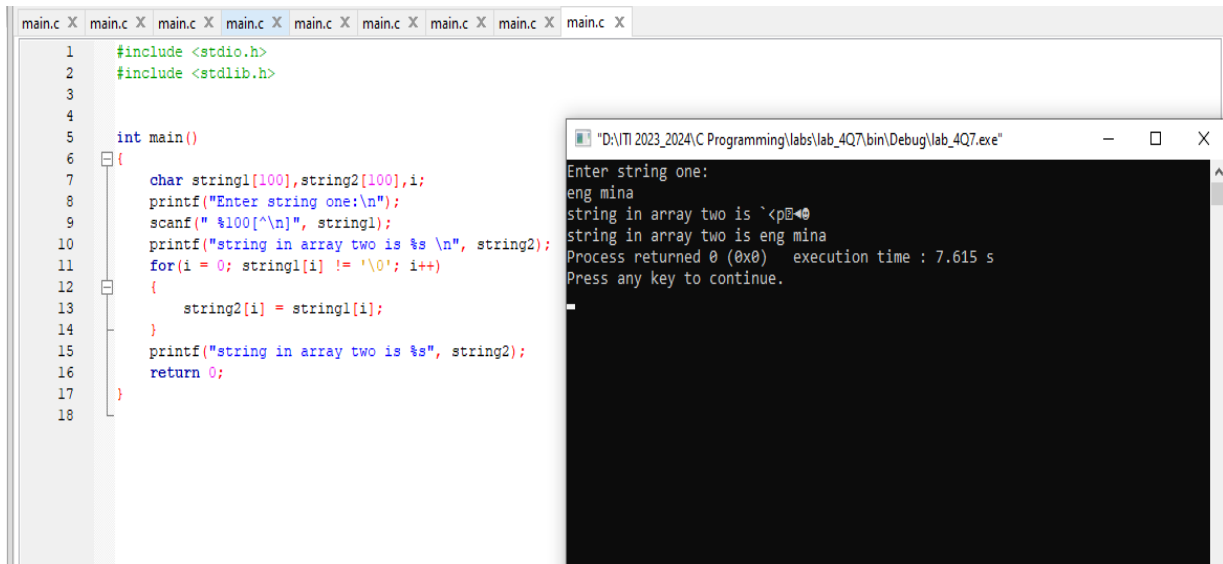
The screenshot shows a C program in a code editor and its execution in a terminal window. The program defines a character array 'string' of size 100 and an integer array 'count' of size 128, initialized to zero. It prompts the user to enter a string, reads it using 'scanf', and then iterates through each character. For each character, it increments the corresponding index in the 'count' array. Finally, it prints the frequency of each character that appears in the string.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main() {
6     char string[100];
7     int count[128] = {0};
8
9     printf("enter a string: ");
10    scanf("%s", string);
11    for (int i = 0; string[i] != '\0'; i++) {
12        count[(int)string[i]]++; // input the ASCII of the character and increment when it appear
13    }
14    for (int i = 0; i < 128; i++) {
15        if (count[i] > 0) {
16            printf("'c' appear %d times\n", (char)i, count[i]);
17        }
18    }
19    return 0;
20 }
21
22
```

Execution output:

```
enter a string: noor
'n' appear 1 times
'o' appear 2 times
'r' appear 1 times
Process returned 0 (0x0) execution time : 3.192 s
Press any key to continue.
```

### - C Program to Copy String Without Using strcpy()



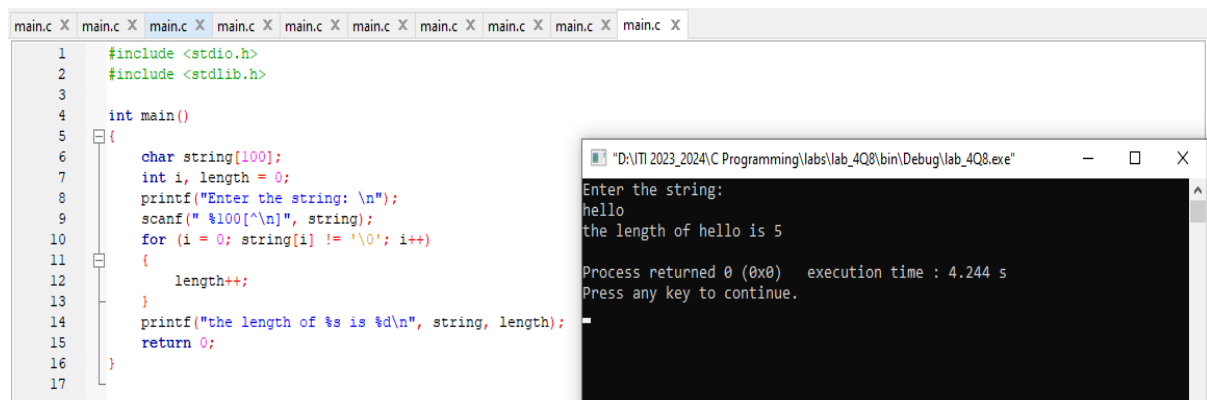
The screenshot shows a C program in a code editor and its execution in a terminal window. The program defines two character arrays, 'string1' and 'string2', both of size 100. It prompts the user to enter a string, reads it into 'string1', and then copies each character from 'string1' to 'string2' using a loop. Finally, it prints the contents of 'string2'.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main()
6 {
7     char string1[100], string2[100];
8     printf("Enter string one:\n");
9     scanf("%100[^\n]", string1);
10    printf("string in array two is %s\n", string2);
11    for (i = 0; string1[i] != '\0'; i++)
12    {
13        string2[i] = string1[i];
14    }
15    printf("string in array two is %s", string2);
16    return 0;
17 }
18
```

Execution output:

```
Enter string one:
eng mina
string in array two is
string in array two is eng mina
Process returned 0 (0x0) execution time : 7.615 s
Press any key to continue.
```

### - C Program to Find the Length of a String without Using strlen()



The screenshot shows a C program in a code editor and its execution in a terminal window. The program defines a character array 'string' of size 100 and an integer variable 'length' initialized to 0. It prompts the user to enter a string, reads it into 'string', and then iterates through each character until the null terminator is reached, incrementing 'length' for each character. Finally, it prints the length of the string.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main()
6 {
7     char string[100];
8     int i, length = 0;
9     printf("Enter the string: \n");
10    scanf("%100[^\n]", string);
11    for (i = 0; string[i] != '\0'; i++)
12    {
13        length++;
14    }
15    printf("the length of %s is %d\n", string, length);
16    return 0;
17 }
18
```

Execution output:

```
Enter the string:
hello
the length of hello is 5
Process returned 0 (0x0) execution time : 4.244 s
Press any key to continue.
```

### - C Program to Remove all Characters in a String Except Alphabet.

```
main.c X main.c X main.c X main.c X main.c X main.c X main.c X main.c X main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char string[100];
7      int i, length=0;
8      printf("enter s string ");
9      scanf("%100[^\n]", string);
10     for (i=0; string[i] != '\0'; i++) //calculate length
11     {
12         length++;
13     }
14     for (int i=0; i< length; i++)
15     {
16         if (!(string[i]>='a' && string[i]<='z') || (string[i]>='A' && string[i]<='Z')) //insure that it not contains only alphabetic
17         {
18             for (int j=i; j<length; j++)
19             {
20                 string[j]=string[j+1]; //shift letter to the left as if i remove the strange character
21             }
22             length--; //becz we remove the strange character
23             i--; //to re check that we get letter
24         }
25     }
26     printf("the alphabetic string is = %s\n", string);
27 }
28
```

"D:\ITI\_2023\_2024\C Programming\labs\lab\_4Q9\bin\Debug\lab\_4Q9.exe" — □ ×

enter s string hel\*\*lo  
the alphabetic string is = hello

Process returned 0 (0x0) execution time : 5.182 s  
Press any key to continue.

### - C Program to Covert all Characters in a String to Capital Case.

```
main.c X main.c X main.c X main.c X main.c X main.c X main.c X main.c X main.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      char string[100];
6      printf("Enter a string: ");
7      scanf("%100[^\n]", string);
8      for (int i = 0; string[i] != '\0'; i++) {
9          if (string[i] >= 'a' && string[i] <= 'z') {
10              string[i] = string[i] - ('a' - 'A'); //difference between any lower and upper is 32 so this to get uppercase
11          }
12      }
13      printf("the string in upper is : %s\n", string);
14
15      return 0;
16 }
17
```

"D:\ITI\_2023\_2024\C Programming\labs\lab\_4Q10\bin\Debug\lab\_4Q10.exe" — □ ×

Enter a string: eng mina  
the string in upper is : ENG MINA

Process returned 0 (0x0) execution time : 6.168 s  
Press any key to continue.