



## Lab 6

### Objectives:

- Working with arrays and application on nested loops
- User defined Functions
- Strings manipulation

1. (Matrix manipulation): Write a program to subtract two matrix A and B.

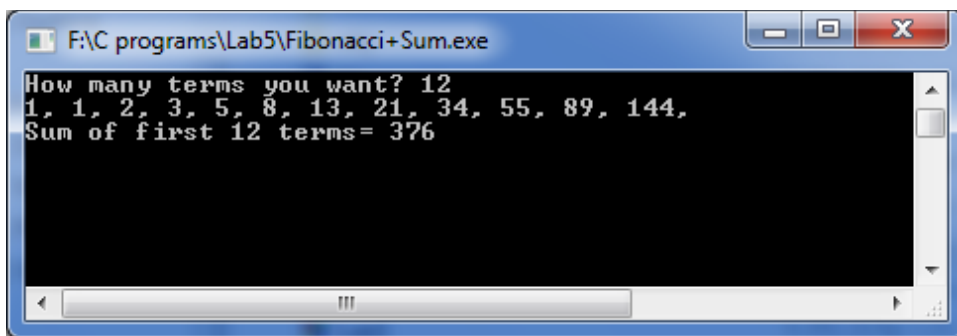
```
#include <stdio.h>
#define row  4
#define col  3
#define v1   25
#define v2   12
#define v3    6
#define v4    3
int main()
{
    int A[row][col] = {{v1,v1,v1},{v2,v2},{v3},{v4}};
    int B[row][col] = {{v4},{v3,v3},{v2},{0}};
    int C[row][col] = {0};
    for (int i = 0; i<row; i++){
        for (int j = 0; j<col; j++){
            C[i][j] = A[i][j] - B[i][j];
        }
    }
    printf("Matrix A:\n");
    for (int i = 0; i<row; i++){
        for (int j=0; j<col; printf("%2d ",A[i][j]),j++);
        printf("\n");
    }
    printf("\nMatrix B:\n");
    for (int i = 0; i<row; i++){
        for (int j=0; j<col; printf("%2d ",B[i][j]),j++);
        printf("\n");
    }
    printf("\nMatrix A-B:\n");
    for (int i = 0; i<row; i++){
        for (int j=0; j<col; printf("%2d ",C[i][j]),j++);
        printf("\n");
    }
    return 0;
}
```

The screenshot shows a Windows command prompt window titled "F:\C program...". It displays the output of the C program. Matrix A is a 4x3 matrix with values 25, 25, 25 in the first row; 12, 12, 0 in the second row; 6, 0, 0 in the third row; and 3, 0, 0 in the fourth row. Matrix B is a 4x3 matrix with values 3, 0, 0 in the first row; 6, 6, 0 in the second row; 12, 0, 0 in the third row; and 0, 0, 0 in the fourth row. The result Matrix A-B is a 4x3 matrix with values 22, 25, 25 in the first row; 6, 6, 0 in the second row; -6, 0, 0 in the third row; and 3, 0, 0 in the fourth row. The window ends with the message "Process returned 0 (0x0)" and "Press any key to continue".



2. **(Fibonacci series)** Write a program that reads n, then generates the first N of Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, and finds the sum of the N terms. (Note:  $F_1 = 1$ ,  $F_2 = 1$ ,  $F_i = F_{i-1} + F_{i-2}$  for every  $i > 2$ ).

```
#include <stdio.h>
long Fibonacci(int);      /* function prototype */
int main()
{
    int term;
    long f,sum = 0;
    printf("How many terms you want? ");
    scanf("%d",&term);
    for(int i=1;i<=term;i++){
        f = Fibonacci(i);    /* function call */
        sum +=f;
        printf("%ld, ",f);
    }
    printf("\nSum of first %d terms= %ld\n",term, sum);
    return 0;
}
long Fibonacci(int n)      /* function definition */
{
    if (n <=2) { return (1);}
    else{return (Fibonacci(n-1) + Fibonacci(n-2));}    /* function call! Recursion */
}
```





### 3. (Duplicate Elimination)

Write a program to read N numbers in the range [10 , 100] and save them into an array. As each number is read, save it into the array only if it is not a duplicate of a number that already exists. The code should handle the “worst case” in which 50 different numbers are needed.

```
#include <stdio.h>
#define Size 50
char C_Duplicate(int List[], int item);    /* function prototype */
int main(){
    int Elements[Size] = {0}, n , x , count = 0;
    char Found;
    do {
        printf("How many numbers do you want (1 , %d )? ",Size);
        scanf("%d",&n);
    } while (n < 0 || n > Size);
    for (int i = 0 ; i < n ; i++) {
        do {
            printf("Enter Number in range (10,100): ");
            scanf("%d",&x);
        } while (x < 10 || x > 100);
        Found = C_Duplicate(Elements,x);    /* function call */
        if (Found == 'D') {printf("Duplicate\n");}
        else {Elements[count] = x; count++; }
    }
    printf("%d different numbers had been saved\n",count);
    for (int i = 0 ; i < count ; i++) { printf("%5d\n",Elements[i]); }
    return 0;
}
char C_Duplicate(int List[Size], int item) /* function definition */
{
    for (int i = 0; i < Size ; i++) {
        if (List[i] == item) { return ('D'); }
    }
    return (0);
}
```

```
"F:\C programs\Lec-Functions\Duplicate.exe"
How many numbers do you want <1 , 50 >? 8
Enter Number in range <10,100>: 60
Enter Number in range <10,100>: 25
Enter Number in range <10,100>: 90
Enter Number in range <10,100>: 40
Enter Number in range <10,100>: 25
Duplicate
Enter Number in range <10,100>: 70
Enter Number in range <10,100>: 90
Duplicate
Enter Number in range <10,100>: 30
6 different numbers had been saved
 60
 25
 90
 40
 70
 30

Process returned 0 (0x0)   execution time : 37.931 s
Press any key to continue.
```

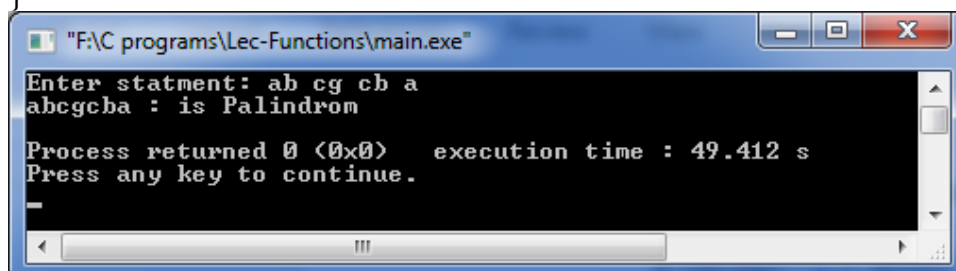


4. **(Palindrome Checking):** Write a program to read in a string into a character array of a length of at most 100. It is better to use `gets(string)` as it reads the entire input line but `scanf` stops reading as soon as it detects a white space. The program should check if the input is a palindrome or not. [A palindrome is a string that reads the same from right to left as well as from left to right].

[Note: `strlen(string)` is a function in `<string.h>` library to return the length of string.]

```
#include <stdio.h>
#include <string.h>
char Palindrome(char str[]);
int main() {
    char str[100], target[100], check;
    int s_len, s_pos, t_pos;
    printf("Enter statment: ");
    gets(str);
    // Remove spaces
    s_pos = 0; t_pos = 0; s_len = strlen(str);
    while(s_pos <= s_len){
        if (str[s_pos] != ' '){
            target[t_pos] = str[s_pos];
            t_pos++;
        }
        s_pos++;
    }
    // End of Remove spaces
    check = Palindrome(target);
    if (check == 'Y') { printf("%s : is Palindrom\n", target); }
    else{ printf("%s : is NOT Palindrom\n", target); }
    return 0;
}

char Palindrome(char str[]) {
    int start = 0; int end = strlen(str) - 1;
    while (end > start) {
        if (str[start++] != str[end--]) { return 'N'; }
    }
    return 'Y';
}
```





5. [Self-Reading example] Functions and Arrays: Write a program to read a key value and return its index in different lists. For simplicity, assume we have 3 lists initialized by the values ([20,25,30,...,65] , [40,45,50,...,85] , [30,35,40,...,75]. The program output should be as the sample shown below.

```
#include <stdio.h>
int Search_index(int Elements[] , int element);
int main()
{
    int list1 [10],list2 [10],list3 [10];
    int s1, s2, s3 ,id;
    for (int i=0;i<10;++i){ list1[i]= 20 + i * 5; printf("%d, ",list1[i]); }
    printf("\n");
    for (int i=0;i<10;++i){ list2[i]= 40 + i * 5; printf("%d, ",list2[i]); }
    printf("\n");
    for (int i=0;i<10;++i){ list3[i]= 30 + i * 5; printf("%d, ",list3[i]); }
    printf("\n Enter Element id: ");
    scanf("%d",&id);
    /* Search lists for required id */
    s1 = Search_index(list1, id);
    s2 = Search_index(list2, id);
    s3 = Search_index(list3, id);
    /* Print the results */
    if (s1!= 10) { printf("List #1: %d\n",s1); }
    else{ printf(" List #1: Not Found\n"); }
    if (s2 != 10){ printf("List #2: %d \n",s2); }
    else{ printf(" List #2: Not Found\n"); }
    if (s3 != 10){ printf("List #3: %d\n",s3); }
    else{ printf(" List #3: Not Found\n"); }
    return 0;
}

int Search_index(int N[10], int x) {
    int f=0;
    while (N[f]!=x && f<10){ f++; }
    return (f);
}
```

```
"F:\C programs\Lec-Functions\Search-index.exe"
20, 25, 30, 35, 40, 45, 50, 55, 60, 65,
40, 45, 50, 55, 60, 65, 70, 75, 80, 85,
30, 35, 40, 45, 50, 55, 60, 65, 70, 75,
Enter Element id: 35
List #1: 3
List #2: Not Found
List #3: 1

Process returned 0 (0x0)   execution time : 3.702 s
Press any key to continue.
```



Assignment:

1. Write a program to compute the multiplication of matrices A and B. The program should read the values of A and B and print out the result. The dimensions of A and B are as follows: A[4][3] , B[3][2].
2. Write a program to read a matrix of 4 rows and 3 columns, then it returns the value and location of the maximum element and the minimum element.
3. Rewrite the Palindrome checking program (example 4) but use only one array for reading the input and then remove the spaces from it without using any additional arrays! [the size of the array should not exceed the length of input string]