

# FIRST PROJECT OBJECTIVE

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

```
int iterativesum(int, int);
int recursivesum(int, int);
```

```
int main()
```

```
{ /* Project's First Subject/Question. */
```

```
iterativesum(num, cycle){
```

```
recursivesum(num, cycle){
```

As the picture refers to the code was formed out of three major parts.

1. *Interface (Main Function).*
2. *Iterative Function.*
3. *Recursive Function.*

# 1-INTERFACE

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

int iterativesum(int, int);
int recursivesum(int, int);

int main()
{
    /* Project's First Subject/Question. */
    int num;
    int cycle;
    int counter;
    int forrecursive;
    printf("Please set two integer for summation \n \n first integer : ");
    scanf("%d", &num);
    printf("\n Second integer : ");
    scanf("%d", &cycle);
    printf("\n Iterative result is : %d\n ", iterativesum(num, cycle));
    printf("\n \n");
    printf(" Recursive result is : %d\n\n", recursivesum(num, cycle));
}
```

1. To begin the process requests two integers.
2. These two integers were saved to the memory.
3. Then called out required functions for the result.

## Output:

```
Please set two integer for summation
first integer : 5
Second integer : 2
```

As you can see, everything is clear for an user-friendly interface.

# 2-ITERATIVE FUNCTION

```
iterativesum(num, cycle){  
    int i1;  
    int i2;  
    int num1;  
    num1=num-1;  
    for(i2=1; i2<cycle; i2++){  
        for(i1=1; i1<=num1; i1++){  
            num=num+i1;  
        }  
        num1=num-1;  
    }  
    num1=num1+1;  
    return num1;  
}
```

i2  
has  
using  
for  
cycle  
loop  
and  
i1  
has  
using  
for  
summation  
Loop

## Output:

```
Please set two integer for summation  
first integer : 5  
Second integer : 2  
Iterative result is : 15
```

As you can see,

*it's works without any issue.*

# 3-RECURSIVE FUNCTION

```
46 int recursivesum(num, cycle){
47     if(cycle==0){
48         return num;
49     }
50     else if(cycle==1){
51         return recursive(num);
52     }
53     num=recursive(num);
54
55     return recursivesum(num, cycle-1);
56 }
57 int recursive(num){
58     if(num==1){
59         return 1;
60     }
61
62     return num + recursive(num-1);
63 }
```

I use **if** and **else if** command for control

mechanism.

And define some counters for loops.

# SECOND PROJECT OBJECTIVE

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int control_f(char[]);
6  char control_l(char[]);
7
8  int main()
9  {
10
11     char sentence[100];
12     printf("%25cXXXXXXX SENTENCE CHECKER XXXXXX\n");
13
14     while (1){
15
16         return 0;
17     }
18
19     int control_f(char sentence[]){
20
21     char control_l(char sentence[]){
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
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

As the picture refers to the code was formed out of three major parts.

1. *Interface .*
2. *Fully or Not.*
3. *Most Repeated Letter.*

# 1-INTERFACE

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int control_f(char[];
6  char control_1(char[];
7
8  int main()
9  {
10
11     char sentence[100];
12     printf("%25cXXXXXXX SENTENCE CHECKER XXXXXXX\n");
13
14     while (1){
15         printf("\n \nPlease enter a sentence : ");
16         gets(sentence);
17
18
19         if(control_f(strlwr(sentence))==1){
20             printf("\n"); puts(sentence);
21             printf("is a fully sentence\n");
22         }
23         else{
24             printf("\n ----> "); puts(sentence);
25             printf(" is not a fully sentence\n");
26         }
27         printf("\nmost repeated letter is : %c\n",control_1(sentence));
28     }
29     return 0;
30 }
```

1. To begin the process requests a sentence.

2. Sentence is save as array using “gets” command .

3. Then called out required functions for determine whether the sentence is fully and find the most repeated letter.

## Output:

```
XXXXXXX SENTENCE CHECKER XXXXXXX
Please enter a sentence : is this sentence fully? and what is the most repeated letter in this sentence?
```

# 2-FULLY OR NOT

```
31
32 int control_f(char sentence[]){
33
34     char alphabet[]={'a','b','c','d',
35                      'e','f','g','h',
36                      'i','j','k','l',
37                      'm','n','o','p',
38                      'q','r','s','t',
39                      'u','v','w','x',
40                      'y','z'};
41     int a=0, b=0;
42     int c=0;
43     while(sentence[a]!='\0'){
44         for(b=0; b<26; b++){
45             if(sentence[a]==alphabet[b]){
46                 alphabet[b]='-';
47                 c++;
48             }
49         }
50         a++;
51     }
52     if(c==26){
53         return 1;
54     }
55     return 0;
56 }
```

First I define the alphabet with an array.

Then define some counters for calculations and control mechanisms.

I use **for** and **if** functions for check alphabet .

## Output:

```
XXXXXXXX SENTENCE CHECKER XXXXXXXX

Please enter a sentence : is this sentence fully? and what is the most repeated letter in this sentence?
----> is this sentence fully? and what is the most repeated letter in this sentence?
is not a fully sentence

most repeated letter is : e

Please enter a sentence : The quick brown fox jumps over the lazy dog
the quick brown fox jumps over the lazy dog
is a fully sentence

most repeated letter is : o

Please enter a sentence : _
```

# 3-MOST REPEATED LETTER

First I define alphabet with an array.

Than define some counters and arrays for calculations and control mechanisms.

I use **for** and **if** functions for check all letters.

## Output:

```
XXXXXXXX SENTENCE CHECKER XXXXXXXX

Please enter a sentence : is this sentence fully? and what is the most repeated letter in this sentence?
----> is this sentence fully? and what is the most repeated letter in this sentence?
is not a fully sentence

most repeated letter is : e
```

*As you can see, it's works without any issue.*



# THIRD PROJECT OBJECTIVE

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <conio.h>
5  int main()
6  {
7      int matrix_1[100][100], matrix_2[100][100];
8      int addition[100][100], transpose, multiplication[100][100];
9      int secim, column, column2, row, row2;
10     int c, r, k;
11     srand(time(NULL));
12     printf(" Choose one of them and enter the number of choosen one:\n");
13     printf(" 1.) Transpose\n 2.) Addition\n 3.) Multiplication\n");
14     scanf("%d", &secim);
15
16     if(secim==1){                                     //Transpose.
17
18     }
19     else if(secim==2){                                 //Addition.
20
21     }
22     else if (secim==3){                               //Multiplication.
23
24     }
25     getch();
26     return 0;
27 }
```

As the picture refers to the code was formed out of four major parts.

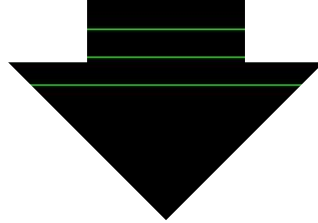
1. *Interface .*
2. *Transpose.*
3. *Addition.*
4. *Multiplication.*

## 1-INTERFACE

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <conio.h>
5  int main()
6  {
7      int matrix_1[100][100], matrix_2[100][100];
8      int addition[100][100], transpose, multiplication[100][100];
9      int secim, column, column2, row, row2;
10     int c, r, k;
11     srand(time(NULL));
12     printf(" Choose one of them and enter the number of choosen one:\n");
13     printf(" 1.) Transpose\n 2.) Addition\n 3.) Multiplication\n");
14     scanf("%d", &secim);
15

```



1. To begin the process requests a choosing operation.
2. Then called out required questions for the function.

## Output:

```

Choose one of them and enter the number of choosen one:
1.) Transpose
2.) Addition
3.) Multiplication
1

```

As you can see

everything is clear for an *user-friendly* interface.

# 2-TRANPOSE

```
int c, r, c_2;
srand(time(NULL));
printf("Matrix ile hangi islemi yapmak istiyorsun?\n");
printf(" Choose one of them and enter the number of choosen one:\n");
printf(" 1.) Transpose\n 2.) Addition\n 3.) Multiplication\n");
scanf("%d", &secim);

if(secim==1) { //Transpose.
    printf("\n\nPlease enter dimensions what you want");
    printf("\nrow: "); scanf("%d",&row);
    printf("\ncolumn: "); scanf("%d",&column);

    for(c=0; c<column; c++){ //Random matrix olusturma islemi.
        for(r=0; r<row; r++){
            matrix_1[r][c] = rand() % 100 ;
        }
    }

    printf("\n matrix1 : \n"); //Matrisi gosteren kod.

    for(c=0; c<column; c++){
        for(r=0; r<row; r++){
            printf("%d\t",matrix_1[r][c]);
            printf("\n\n");
        }
        // i = r // j = c //

        printf("\n Transpose : \n") ; //Matrisin Transpozasi.
    for (r = 0; r < row; r++)
    {
        for (c = 0; c<column; c++) //sonucu gosteren kod.
        {
            printf("%4d", matrix_1[r][c]);
        }
        printf("\n\n\n");
    }
}
else
else
return 0;
```

*if(secim==1) shows that, if user choose first function-transpose- use then this algorithm.*

**Printf and scanf functions** these are requests to user for matrix's dimensions.

And save the data from user about row and column.

**First and Second FOR Loops** Are using for generate random matrix values with dimensions from user.

**Last two FOR Loops** Are using for printing generated matrix on screen.

## 2.1-TRANSPOSE OUTPUT

```
2.) Addition
3.) Multiplication
1

Please enter dimensions what you want
row: 5

column: 3

matrix1 :
21      15      37      86      0
80      42      10      89      39
70      68      34      13      39

Transpose :
21      80      70
15      42      68
37      10      34
86      89      13
0       39      39
```

1. Choosing transpose function with write "1".
2. Entering matrix's dimensions.
3. Program prints "matrix1".
4. Program prints "matrix1"s "Transpose".

# 3-ADDITION

```
49 }
50 else if(secim==2){ //Addition.
51     printf("\n\nPlease enter first matrix's dimensions what you want");
52     printf("\nrow: "); scanf("%d",&row);
53     printf("\ncolumn: "); scanf("%d",&column);
54
55     for(r=0; r<row; r++){ //Random 1.matrix olusturma islemi.
56         for(c=0; c<column; c++){
57             matrix_1[r][c] = rand() % 100 ;
58         }
59     }
60     for(r=0; r<row; r++){ //Random 1.matrix olusturma islemi.
61         for(c=0; c<column; c++){
62             matrix_2[r][c] = rand() % 100 ;
63         }
64     }
65
66     printf("\n matrix1 : \n"); //1. Matrisi gosteren kod.
67     for(c=0; c<column; c++){
68         for(r=0; r<row; r++){
69             printf("%d\t",matrix_1[r][c]);
70             printf("\n\n");
71         }
72     }
73     printf("\n matrix2 : \n"); //2. Matrisi gosteren kod.
74     for(c=0; c<column; c++){
75         for(r=0; r<row; r++){
76             printf("%d\t",matrix_2[r][c]);
77             printf("\n\n");
78         }
79     }
80     printf("\n ADDITION OF TWO MATRIX : \n"); //Toplama isleminin yapilmasi.
81     for(c=0; c<column; c++){
82         for(r=0; r<row; r++){
83             addition[r][c]=matrix_1[r][c]+matrix_2[r][c];
84         }
85     }
86     for(c=0; c<column; c++){ //Sonucun gosterilmesi.
87         for(r=0; r<row; r++){
88             printf("%d\t",addition[r][c]);
89             printf("\n\n");
90         }
91     }
```

**1.if(secim==2)** shows that, if user choose second function-Addition- then use this algorithm.

**2.printf and scanf functions** these are requests to user for matrices's dimensions.

And save the data from user about rows and columns.

**3.Four FOR Loops** Are using for generate random matrices values with dimensions from user.

**4.Last two FOR Loops** Are using for printing generated matrices on screen.

**5.Next two FOR Loops** Are using for calculate the Addition

# 3.1-ADDITION OUTPUT

```
Choose one of them and enter the number of choosen one
1.) Transpose
2.) Addition
3.) Multiplication
2

Please enter first matrix's dimensions what you want
row: 4
column: 3

matrix1 :
83    98    30    65
52     15    15    71
11     74    87    19

matrix2 :
37     69    12    75
17     33     9    33
97     53    44    60

ADDITION OF TWO MATRIX :
120    167    42    140
69     48    24    104
108    127    131    79
```

1. Choosing Addition Function with write '2'.

2. Entering matrices's dimensions.

3. Program prints "matrix1" and "matrix2".

4. Program prints "matrix1" + "matrix2".

# 4-MULTIPLICATION

```
1  }
2  else if (secim==3){ //Multiplication.
3
4  printf("\n\nPlease enter first matrix's dimensions what you want"); //ilk matris boyutlarinin alinmasi.
5      printf("\nrow: ");      scanf("%d",&row);
6      printf("\ncolumn: ");    scanf("%d",&column);
7
8      for(r=0; r<row; r++){
9          for(c=0; c<column; c++){
10             matrix_1[r][c] = rand() % 100 ;
11         }
12     }
13     printf("\n\nPlease enter second matrix's dimensions what you want");//ikinci matris boyutlarinin alinmasi.
14     printf("\nrow: ");      scanf("%d",&row2);
15     printf("\ncolumn: ");    scanf("%d",&column2);
16
17     for(r=0; r<row2; r++){
18         for(c=0; c<column2; c++){
19             matrix_2[r][c] = rand() % 100 ;
20         }
21     }
22     printf("\n matrix1 : \n"); //ilk matrisi random olusturma.
23
24     for(c=0; c<column; c++){
25         for(r=0; r<row; r++){
26             printf("%d\t",matrix_1[r][c]);
27         }
28         printf("\n\n");
29     }
30     printf("\n matrix2 : \n"); //ikinci matrisi random olusturma.
31     for(c=0; c<column2; c++){
32         for(r=0; r<row2; r++){
33             printf("%d\t",matrix_2[r][c]);
34         }
35         printf("\n\n");
36     }
37 }
38 return 0;
```

**1.if(secim==3)** shows that, if user choose second function-Multiplication- then use this algorithm.

**2.printf and scanf functions** these are requests to user for matrices's dimensions.

And save the data from user about rows and columns.

**3.Four FOR Loops** Are using for generate random matrices values with dimensions from user.

**4.Last two FOR Loops** Are using for printing generated matrices on screen.

**5.Next two FOR Loops** Are using for calculate the Multiplication.



## 4.1-MULTIPLICATION OUTPUT

```
Choose one of them and enter the number of choosen one:
1.) Transpose
2.) Addition
3.) Multiplication
3

Please enter first matrix's dimensions what you want
row: 3
column: 3

Please enter second matrix's dimensions what you want
row: 3
column: 3

matrix1 :
1      8      6
9      0      4
4      6      9

matrix2 :
2      8      6
7      5      4
9      1      9

Carpma islemi basliyor:
112    54    92
36     4     36
131    71    129
```

1.Choosing Multiplication Function with write '3'.

2.Entering matrices's dimensions.

3.Program prints "matrix1"and "matrix2".

5. Program prints "matrix1" \* "matrix2".