

20BDS0405- Bimal Parajuli

Advanced C Programming
(CSE2010)

Assignment- 1 From Moodle

Question 1

Write a program to store a given list [Number of inputs is even, max 24 inputs] in the following two-dimensional matrix (5*5) – in the same way as seen using pointers,

Given list:

Example 1:

1 2 3 4 5 6 [Even Number]

Two-Dimensional Matrix:

1	2	3	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	4	5	6

Example 2:

Values: 1 2 3 4 5 6 7 8 9 10 11 12

Two-Dimensional Matrix:

1	2	3	4	5
6	0	0	0	0
0	0	0	0	0
0	0	0	0	7
8	9	10	11	12

Code:

```
#include <stdio.h>
void display(int *);

int main()
{
    int n, temp, temp1;
    printf("Enter the number of integers you want to input.\n(should be even positive integer):");
    while (1)
    {
        scanf("%d", &n);                                //get number of entries in the matrix(should be a positive even number).
        if (n > 0 && n % 2 == 0)
        {
            break;
        }
        else
        {
            printf("Re-enter a positive even integer.");
        }
    }

    int myarray[25];
    int *ptr = myarray;
    printf("Now enter the %d integers.\n", n);
    for (int i = 0; i < 25; i++)
    {
        if (i < (n >> 1))                                //Used right shift operator to divide by 2.
        {
            scanf("%d", (ptr + i));                    //upto n/2 numbers, take input from user and insert into matrix.
        }

        if ((i >= (n >> 1)) && (i < (25 - (n >> 1))))
        {
            *(ptr + i) = 0;                            //from n/2 to last n/2 numbers, put directly zero.
        }

        if (i >= (25 - (n >> 1)))
        {

```

```

        scanf("%d", (ptr + i)); //in last n/2 numbers, take unput from
user and insert into matrix.
    }
}

display(myarray);
return 0;
}

void display(int *p)
{
    int counter = 0;
    printf("\nFinally, The matrix is:\n");
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            int x = p[counter];
//making the matrix output properly aligned in straight columns.
            if (x > 999)
            {
                printf("%d ", x);
            }
            else if (x > 99)
            {
                printf("%d ", x);
            }
            else if (x > 9)
            {
                printf("%d ", x);
            }
            else
            {
                printf("%d ", x);
            }
            counter++;
        }
        printf("\n");
    }
}

```

Question 2:

Write a program to print the nearest integer value of corresponding float value given by the user, without using any library function.

Code:

```
#include<stdio.h>
int main()
{
    float x;
    int y;

    scanf("%f", &x);           //get float input from user.
    y = (int) (x + 0.5);        //convert into nearest int by adding 0.5 and removing the trailing decimal part.
    printf("%d", y);           //display the resultant integer.
    return 0;
}
```

Question 3:

Try to print the following pattern using loops:

0

1 0

0 1 0

1 0 1 0

0 1 0 1 0

....

For n number of lines.

Eg. For n=1,

0

For n=3,

0

1 0

0 1 0

Code:

```
#include<stdio.h>
int main()
{
    int choice, counter=1;
    scanf("%d", &choice);

    for(int i=1; i<=choice; i++)
    {
        for (int j=1; j<=i; j++)
        {
            if( (counter+i) & 1 )           //checking if its odd by bit manipulation.
            {
                printf("%d ", 1);
            }
            else
            {
                printf("%d ", 0);
            }
            counter++;
        }
        counter=1;
        printf("\n");
    }
    return 0;
}
```

Question 4:

Write a program to print the output (sum) of the series, $1 + x + 2x + 3x\dots$ and $1 + x/2 + 2x/3 + 3x/4\dots$ (Use double for output) till n numbers, where n is coefficient of x.

Code:

```
//Write a program to print the output (sum) of the series,  
//1 + x + 2x + 3x... and  
//1 + x/2 + 2x/3 + 3x/4..  
//(Use double for output) till n numbers, where n is coefficient of x.  
  
#include<stdio.h>  
int main()  
{  
    int n, x;  
    scanf ("%d %d", &n, &x);  
  
    double sum1=1, sum2=1;  
    for (int i=0; i < n; i++)  
    {  
        sum1 += ( double )( i * x ) ;  
    }  
    printf("%lf\n", sum1);  
  
    for(int i=0; i < n; i++)  
    {  
        sum2 += ( (double)(i * x) / (i+1) );  
    }  
    printf("%lf\n", sum2);  
    return 0;  
}
```


Question 5:

Write a program using function to indicate whether a given number is divisible by 5 or 6, if divisible by 5 multiply it to 6 or vice versa and return the value, indicate the divisibility also (which must be calculated in the function). In case the number is divisible by both then multiply the number by 10, then return it, and assign its divisibility as 6.

Code:

```
//Write a program using function to indicate whether a given number is divisible
by 5 or 6,
//if divisible by 5 multiply it to 6 or vice versa and return the value,
//indicate the divisibility also (which must be calculated in the function).
//In case the number is divisible by both then multiply the number by 10, then re
turn it, and assign its divisibility as 6.

#include <stdio.h>

int func(int a, int *b)
{
    if (a % 5 == 0 && a % 6 == 0)
    {
        *b = 6;
        return a * 10;
    }
    else if (a % 5 == 0)
    {
        *b = 5;
        return a * 6;
    }
    else if (a % 6 == 0)
    {
        *b = 6;
        return a * 5;
    }
    return 0;
}

int main()
{
    int x, val, divisiblity = 1;
    //printf("Enter a number: ");
    scanf("%d", &x);
    val = func(x, &divisiblity);

    printf("%d  %d",val, divisiblity );
    // printf("Divisibility: %d and Result of multiplication: %d", val, divisibli
ty);
    return 0;
}
```

Question 6:

Write a program to get four integer variables as input and do the check whether each of them is divisible by 2 and 3 or it is divisible by 5. If it is true and if more than one value is satisfying the condition then select the largest of the numbers which satisfy the condition and pass that value by reference to a function refer, multiply it by 10 and print the variable value in the main.

Code:

```
// Write a program to get four integer variables as input
// and do the check whether each of them is divisible by 2 and 3 or it is divisible by 5.
// If it is true and if more than one value is satisfying the condition,
// then select the largest of the numbers which satisfy the condition
// and pass that value by reference to a function refer,
// multiply it by 10 and print the variable value in the main.

#include <stdio.h>

void mul10(int *max)
{
    *max = *max * 10;
}

int divisible(int);

int main()
{
    int a, b, c, d, max = -1;
    // printf("Enter four numbers: a b c d : ");
    scanf("%d %d %d %d", &a, &b, &c, &d);

    if ((a % 2 == 0 && a % 3 == 0) || a % 5 == 0)
    {
        if (a > max)
```

```
    {
        max = a;
    }
}
if ((b % 2 == 0 && b % 3 == 0) || b % 5 == 0)
{
    if (b > max)
    {
        max = b;
    }
}
if ((c % 2 == 0 && c % 3 == 0) || c % 5 == 0)
{
    if (c > max)
    {
        max = c;
    }
}
if ((d % 2 == 0 && d % 3 == 0) || d % 5 == 0)
{
    if (d > max)
    {
        max = d;
    }
}

if (max == -1)
{

}
else
{
    mul10(&max);
    printf("%d", max);
}
return 0;
}
```

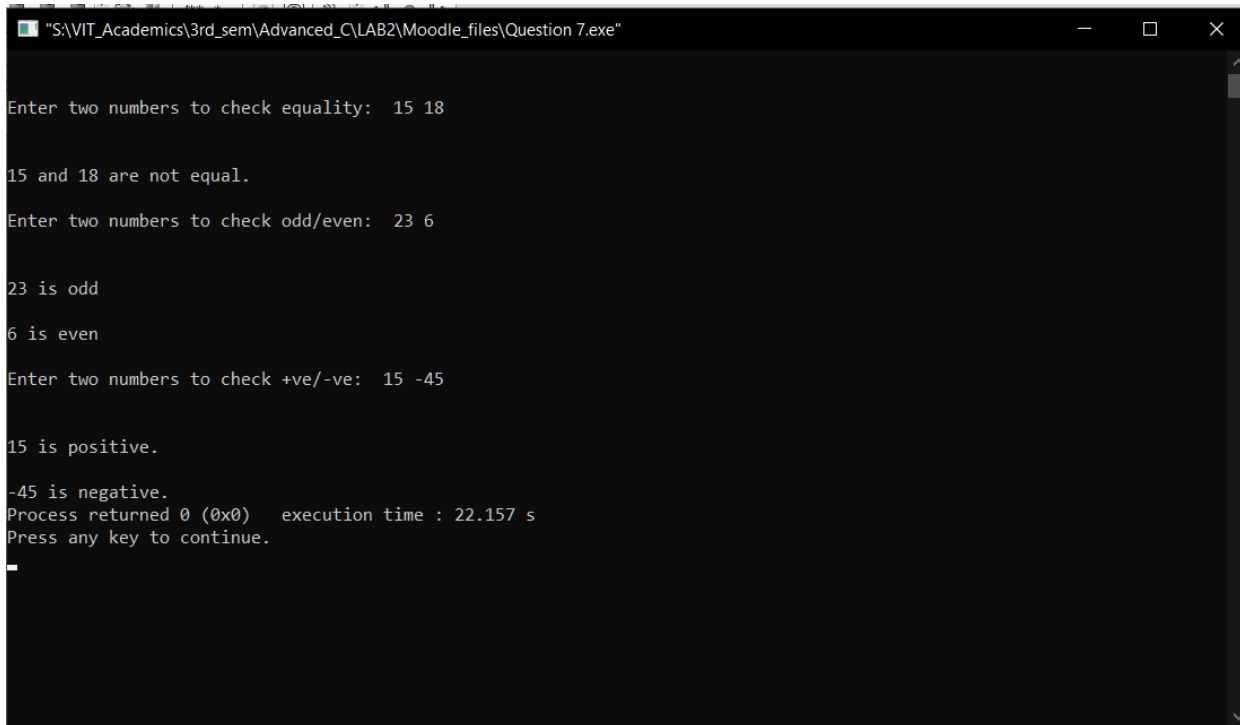
Question 7:

Use only bitwise operators and control statements for the following:

- a. Check whether two numbers are equal*
- b. Check whether a number is odd or even*
- c. Check whether a number is positive or negative (note: the first bit is zero if it is positive)*

Codes:

```
1 // Use only bitwise operators and control statements for the following:
2
3 // a. Check whether two numbers are equal
4
5 // b. Check whether a number is odd or even
6
7 // c. Check whether a number is positive or negative (note: the first bit is zero if it is positive)
8
9 #include<stdio.h>
10 void isequal(int x, int y) ;
11 void iseven(int x);
12 void ispositive(int x);
13 int main()
14 {
15     int a, b;
16
17     printf("\n\nEnter two numbers to check equality: ");
18     scanf("%d %d", &a, &b);
19     isequal(a, b);
20
21     printf("\n\nEnter two numbers to check odd/even: ");
22     scanf("%d %d", &a, &b);
23     iseven(a);
24     iseven(b);
25
26     printf("\n\nEnter two numbers to check +ve/-ve: ");
27     scanf("%d %d", &a, &b);
28     ispositive(a);
29     ispositive(b);
30
31     return 0;
32 }
33
34 //function to check by bit manipulation if two numbers are equal by comparing two numbers bit by bit.
35 //if any corresponding bit are unequal, the numbers are unequal. Else equal
36 //bitwise XOR. If all bits are not same, they're not equal. Else equal
37 void isequal(int x, int y)
38 {
39     if(x ^ y)
40     {
41         printf("\n\n%d and %d are not equal.", x, y);
42     }
43     else
44     {
45         printf("\n\n%d and %d are equal.", x, y);
46     }
47     return;
48 }
49
50 //function to check and display if a number is even or odd by bit manipulation.
51 //if last bit is 1, its odd. Else its even.
52 void iseven(int x)
53 {
54     if(x&1)
55     {
56         printf("\n\n%d is odd", x);
57         return;
58     }
59     printf("\n\n%d is even", x);
60     return;
61 }
62
63 //function to check and display if a number is positive or negative.
64 //If the first bit is 1, its negative. Else its positive.
65 void ispositive(int x)
66 {
67     if(x & (1 << 31))
68     {
69         printf("\n\n%d is negative.", x);
70         return;
71     }
72     printf("\n\n%d is positive.", x);
73     return;
74 }
75 }
```

Test case 1:

```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 7.exe"

Enter two numbers to check equality: 15 18

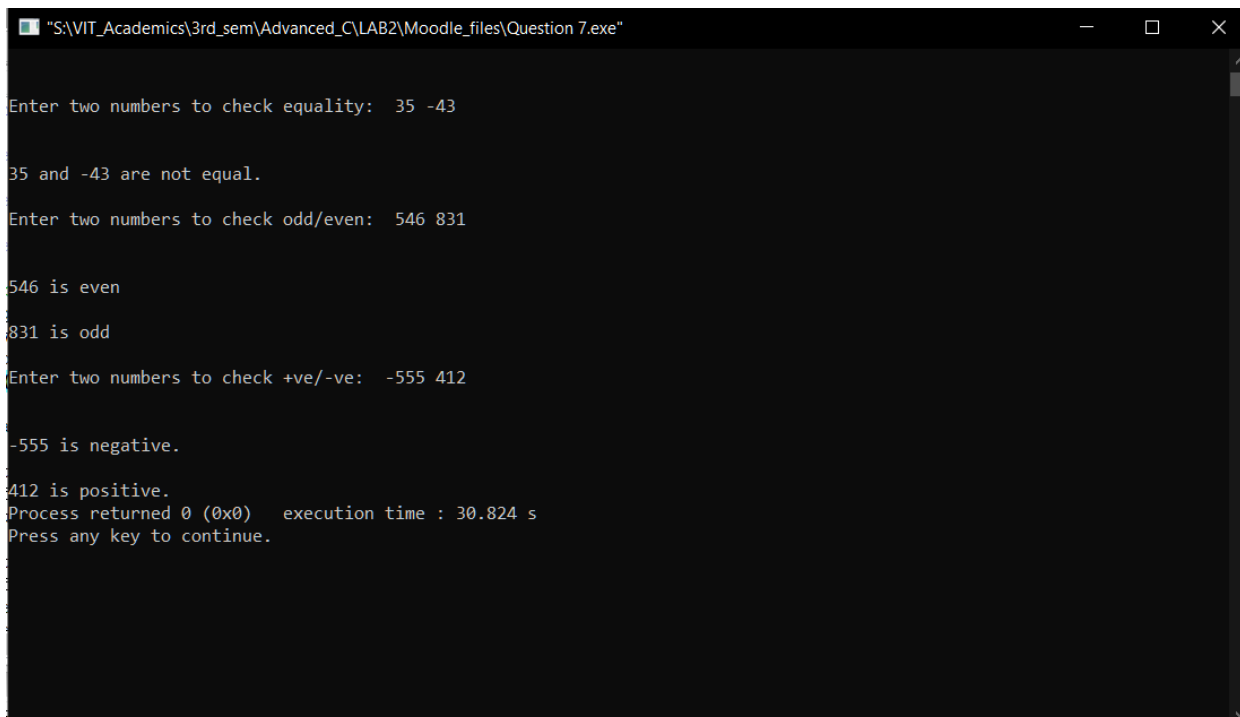
15 and 18 are not equal.

Enter two numbers to check odd/even: 23 6

23 is odd
6 is even

Enter two numbers to check +ve/-ve: 15 -45

15 is positive.
-45 is negative.
Process returned 0 (0x0)   execution time : 22.157 s
Press any key to continue.
_
```

Test case 2:

```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 7.exe"

Enter two numbers to check equality: 35 -43

35 and -43 are not equal.

Enter two numbers to check odd/even: 546 831

546 is even
831 is odd

Enter two numbers to check +ve/-ve: -555 412

-555 is negative.
412 is positive.
Process returned 0 (0x0)   execution time : 30.824 s
Press any key to continue.
```

Question 8:

*Write a program to convert and display a decimal number in number with base 26, with its digit being represented as alphabets from a – z (0-25). Maximum size of input is $26*26 - 1$.*

Eg.

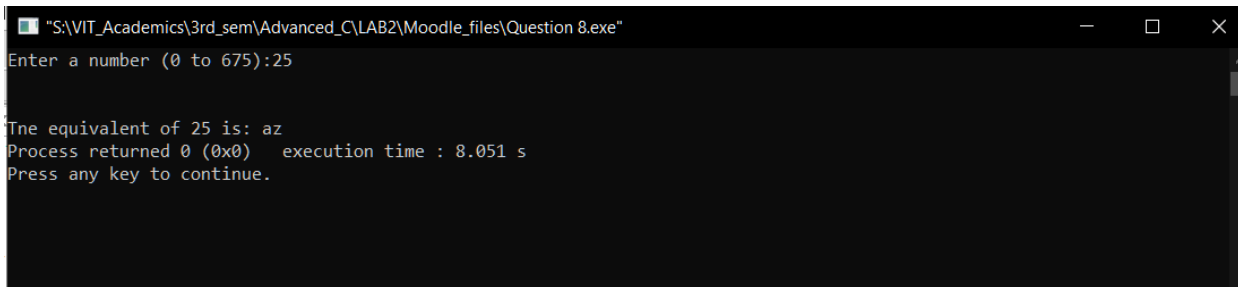
Inp = 25, out= az;

inp = 26, out = ba;

inp=52, out=ca.

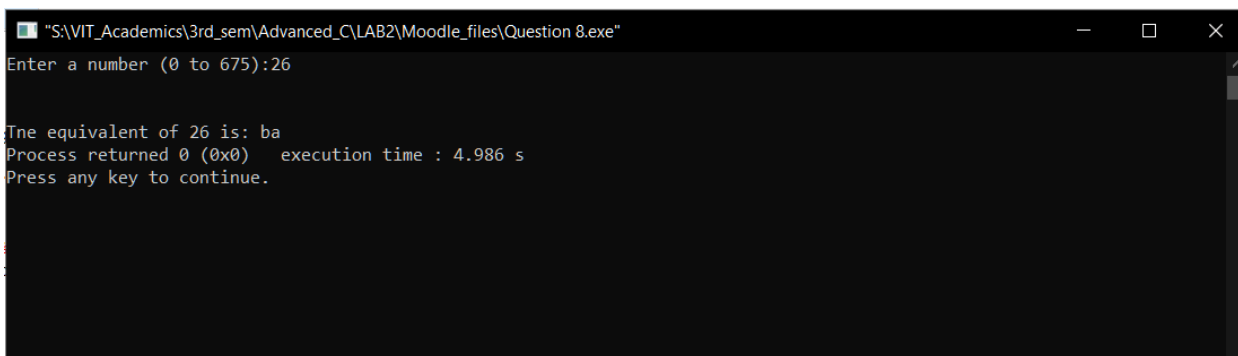
Codes:

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int x, first_dig, second_dig;
6      printf("Enter a number (0 to 675):");
7      scanf("%d", &x);
8      printf("\n\nThe equivalent of %d is: ", x);
9      |
10     second_dig = x % 26;
11     second_dig = second_dig + 'a';
12
13     x = x / 26;
14
15     first_dig = x ;
16     first_dig = first_dig + 'a';
17
18     printf("%c%c", first_dig, second_dig);
19
20     return 0;
21 }
```

Test cases:

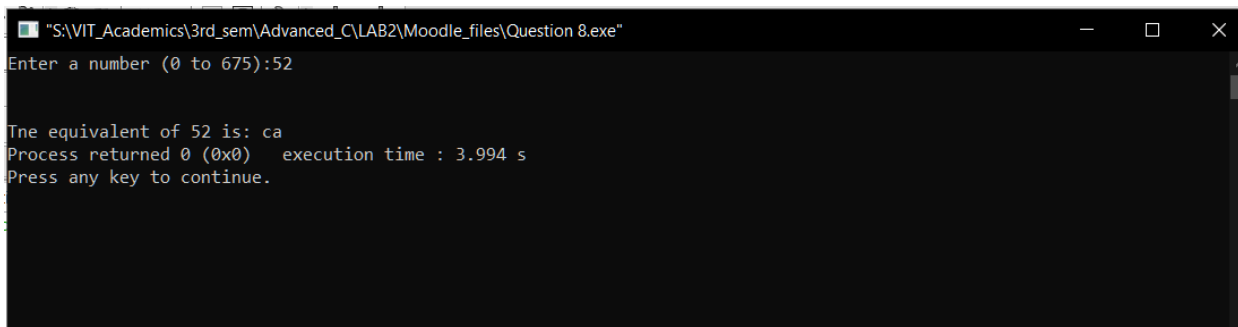
```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 8.exe"
Enter a number (0 to 675):25

The equivalent of 25 is: az
Process returned 0 (0x0) execution time : 8.051 s
Press any key to continue.
```



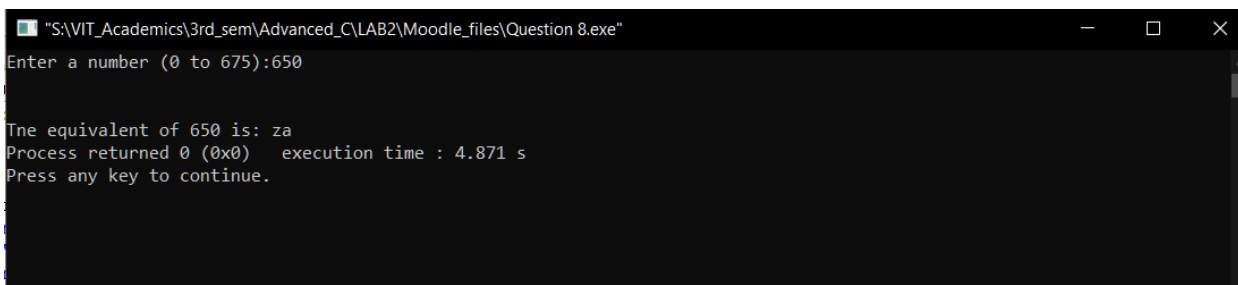
```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 8.exe"
Enter a number (0 to 675):26

The equivalent of 26 is: ba
Process returned 0 (0x0) execution time : 4.986 s
Press any key to continue.
```



```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 8.exe"
Enter a number (0 to 675):52

The equivalent of 52 is: ca
Process returned 0 (0x0) execution time : 3.994 s
Press any key to continue.
```



```
"S:\VIT_Academics\3rd_sem\Advanced_C\LAB2\Moodle_files\Question 8.exe"
Enter a number (0 to 675):650

The equivalent of 650 is: za
Process returned 0 (0x0) execution time : 4.871 s
Press any key to continue.
```

