

EED 1010 ALGORITHMS&PROGRAMMING

NAME&SURNAME: Enes Erten

STUDENT NUMBER: 2018502036

LAB#6

LAB SECTION: 4(laboratory), 1(theory)

DATE: 02.04.2020

Task 1 : An enumeration, introduced by the keyword `enum`, is a set of integer enumeration constants represented by identifiers. Values in an enum start with 0, unless specified otherwise, and are incremented by 1. Write a C program that contains two enumeration variables, named months and seasons. The program displays months in a sequence and also the season to which current month belongs. The example output screen is shown below:

```
No      Month      Season
1       January    Winter
2       February   Winter
3       March      Spring
4       April      Spring
5       May        Spring
6       June       Summer
7       July       Summer
8       August     Summer
9       September  Autumn
10      October    Autumn
11      November   Autumn
12      December   Winter
Press any key to continue . . .
```

Figure 1.1: Output of Laboratory study task1

The code:

```
#include<stdio.h> //include standart input output

enum months{JAN=1,FEB,MAR,APR,MAY,JUNE,JUL,AUG,SEPT,OCT,NOV,DEC};

//declare enum months jan to increment 1 for each member until dec=12 enum

seasons{WINTER=1,SPRING,SUMMER,AUTUMN};

//declare enum seasons winter to increment by 1 to autumn=4

main()
{
    enum months i;

    char m[][DEC]={"January","February","March","Aril","May","June","July","August","September","October","November","December"};

    //declare 2-D array months january to december

    char s[][DEC]={"Winter","Spring","Summer","Autumn"};

    //declare 2-D array seasons winter autumn


    printf("No Month Season\n");

    for(int i=JAN;i<=DEC;i++) //for loop, loops JAN to DEC

    printf("%-3d %-13s %-11s\n",i,m[i-1],s[i%12/3]);

    //display number month season

}
```



```
C:\Users\Enes\OneDrive\Documents\lt.exe

No Month Season
1 January Winter
2 February Winter
3 March Spring
4 Aril Spring
5 May Spring
6 June Summer
7 July Summer
8 August Summer
9 September Autumn
10 October Autumn
11 November Autumn
12 December Winter

-----
Process exited after 0.04266 seconds with return value 0
Press any key to continue . . .
```

The output of the code

Task 2: Write a program that reverses the order of the bits in an unsigned integer value. The program should input the value from the user and call function reverse Bits to print the bits in reverse order. Print the value in bits both before and after the bits are reversed to confirm that the bits are reversed properly.



```

please give an Integer:2
2 = 00000000 00000000 00000000 00000010
reverse of this integer:
1073741824 = 01000000 00000000 00000000 00000000
=

```

Figure 1.2: Output of Laboratory study task2.

The code:

```

#include<stdio.h> //include standart input output

#include<stdlib.h> //include standart library


void db(unsigned a); //function prototype void
dbr(unsigned a); //function prototype unsigned
power(unsigned a,int c); //function prototype


void db(unsigned a) //function definition
{

    unsigned m=1<<31,i; //create 1 and left to the 31 bit


    printf("%u ",a); //display the number a


    for(i=1;i<=32;i++) //for loop,loops 32 times
    {
        if(a&m) //if a&m true
            printf("1"); //display 1
        else //else
            printf("0"); //display 0
        a=a<<1; //assign to the number one bit left


        if(i%8==0) //if i is 8
            printf(" "); //display space


    } //end of for loop
    printf("\n"); //skip new line
} //end of the function

```

```

void dbr(unsigned a) //function definition
{
    unsigned m=1,i,sum=0; //unsigned variables
    printf(" "); //display space

    for(i=1;i<=32;i++) //for loop,loops 32 times
    {
        if(a&m) //if a&m
        {
            printf("1");
            //display 1
            sum+=power(2,32-i); //sum to the calculate the reverse bits value
        }
        //the sum is reverse bitd so it have sum up with 2^(32-i)

        else //else
        {
            printf("0"); //display 0
        }

        m=m<<1; //shift 1 bit to the left

        if(i%8==0) //if i%8 is 0
            printf(" "); //display space
    }

    printf(" %u",sum); //display the value of sum
} //end of the function

```

```

unsigned power(unsigned a,int c) //function definition
{
    unsigned temp=1; //declaration of unsigned variable
    int i; //deklarition of integer variable

```

```

        for(i=1;i<=c;i++) //for loop,loops c times

            temp*=a; //temp=temp*a to calculate power of 2


        return temp; //return the temp value 2^c
    }


main()
{
    unsigned num; //declaration of unsigned variable


    printf("please give an integer>>>");

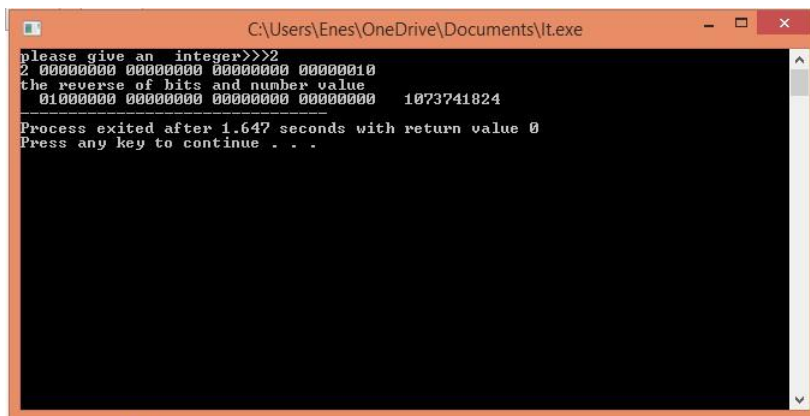
    scanf("%u",&num);    //get a unsigned value


    db(2); //send the num to function db

    printf("the reverse of bits and number value\n");

    dbr(2); //send the num to function dbr
}

```



```

C:\Users\Enes\OneDrive\Documents\lt.exe
please give an integer>>>2
2 00000000 00000000 00000000 00000010
the reverse of bits and number value
01000000 00000000 00000000 00000000 1073741824
=====
Process exited after 1.647 seconds with return value 0
Press any key to continue . . .

```

The output of the code