Advanced C Lab assignment 5

Name: Sankalp Mukim

Registration Number: 20BDS0128

Slot: L27+L28

Table of Contents

[Write a program to read a file and display its contents along with line numbers before each line. 2](#_Toc101560574)

[Code 2](#_Toc101560575)

[Output 3](#_Toc101560576)

[File 3](#_Toc101560577)

[Write a program to copy the contents of one file to another, while doing so replace all lowercase characters with their equivalent uppercase characters 3](#_Toc101560578)

[Code 3](#_Toc101560579)

[Output 4](#_Toc101560580)

[Files 4](#_Toc101560581)

[Write a program to encrypt/decrypt a file using a substitution cipher: in this each character read from the source file is substituted by a corresponding predetermined character and this character is written to the target file. For example, if character 'A' is read from the source file, and if we have decided that every 'A' is to be substituted by '!', then a '!' would be written to the target file in place of every 'A'. 5](#_Toc101560582)

[Code 5](#_Toc101560583)

[Output 6](#_Toc101560584)

[Files 6](#_Toc101560585)

[Given a text file, write a program to create another text file deleting the words 'a', 'the', 'an' and replacing each one of them with a blank space. 7](#_Toc101560586)

[Code 7](#_Toc101560587)

[Output 8](#_Toc101560588)

[Write a program to carry out the following: A. Read a text file 'Input.Txt'. B. Print each word in reverse order 8](#_Toc101560589)

[Code 8](#_Toc101560590)

[Output 9](#_Toc101560591)

[Files 9](#_Toc101560592)

[Write down macro definitions for the following 9](#_Toc101560593)

[1. To find the arithmetic mean of two numbers. 9](#_Toc101560594)

[Code 9](#_Toc101560595)

[Output 9](#_Toc101560596)

[2. To find the absolute value of a number. 9](#_Toc101560597)

[Code 9](#_Toc101560598)

[Output 10](#_Toc101560599)

[3. To convert an uppercase alphabet to lowercase. 10](#_Toc101560600)

[Code 10](#_Toc101560601)

[Output 10](#_Toc101560602)

[4. To obtain the bigger of two numbers. 10](#_Toc101560603)

[Code 10](#_Toc101560604)

[Output 11](#_Toc101560605)

[Write down macro definitions for the following. 1. To test whether a character entered is a small case letter or not. 2. To test whether a character entered is an upper case letter or not. 3. To test whether a character is an alphabet or not. Make use of the macros you defined in (1) and (2) above. 11](#_Toc101560606)

[Code 11](#_Toc101560607)

[Output 11](#_Toc101560608)

[Create an enumerated data type logical with TRUE and FALSE values. Write a program to check whether the entered number is prime or not prime. If the number is prime display 0 otherwise 1. Use enumerated datatype 12](#_Toc101560609)

[Code 12](#_Toc101560610)

[Output 12](#_Toc101560611)

[Create an enumerated datatype for 12 months and display the values in the integer. 13](#_Toc101560612)

[Code 13](#_Toc101560613)

[Output 14](#_Toc101560614)

[Create a user-defined datatype from a structure. The structure should contain the variables such as name, regno, cgpa, and age of students. Use array of structures. 14](#_Toc101560615)

[Code 14](#_Toc101560616)

[Output 15](#_Toc101560617)

# Write a program to read a file and display its contents along with line numbers before each line.

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

***int*** main()

{

    FILE \*fp;

***char*** ch;

***int*** i = 1;

    fp = fopen(**"**q1.txt**"**, **"**r**"**);

    if (fp == NULL)

    {

        printf(**"**File not found\n**"**);

        return 0;

    }

    while ((ch = fgetc(fp)) != EOF)

    {

        printf(**"**%c**"**, ch);

        i++;

    }

    printf(**"**\n**"**);

    fclose(fp);

    return 0;

}

## Output

Text

Description automatically generated

## File

Graphical user interface, text, website

Description automatically generated

# Write a program to copy the contents of one file to another, while doing so replace all lowercase characters with their equivalent uppercase characters

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

#include **<**ctype.h**>**

***int*** main()

{

    FILE \*rptr, \*wptr;

    rptr = fopen(**"**q2\_read.txt**"**, **"**r**"**);

    wptr = fopen(**"**q2\_write.txt**"**, **"**w**"**);

    if (rptr == NULL || wptr == NULL)

    {

        printf(**"**File not found\n**"**);

        return 0;

    }

    printf(**"**The contents of the file are being read and copied to a new file.\n**"**);

***char*** ch;

    while ((ch = fgetc(rptr)) != EOF)

    {

        if (islower(ch))

        {

            ch = toupper(ch);

        }

        fputc(ch, wptr);

    }

    printf(**"**The contents of the file have been succesfully copied to a new file.\n**"**);

    fclose(rptr);

    fclose(wptr);

}

## Output

A screenshot of a computer

Description automatically generated with medium confidence

## Files

Graphical user interface, text, website

Description automatically generated

Graphical user interface, website

Description automatically generated with medium confidence

# Write a program to encrypt/decrypt a file using a substitution cipher: in this each character read from the source file is substituted by a corresponding predetermined character and this character is written to the target file. For example, if character 'A' is read from the source file, and if we have decided that every 'A' is to be substituted by '!', then a '!' would be written to the target file in place of every 'A'.

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

***int*** main()

{

    FILE \*fp;

    FILE \*pp;

    fp = fopen(**"**q3\_read.txt**"**, **"**r**"**);

    pp = fopen(**"**q3\_write.txt**"**, **"**w**"**);

    printf(**"**->Reading the text from the file.\n**"**);

***char*** ch, sh;

    while ((ch = fgetc(fp)) != EOF)

    {

        if ((ch != **'**\n**'**) && (ch != **'** **'**))

        {

            sh = ch + 10;

            fputc(sh, pp);

        }

        else

        {

            fputc(ch, pp);

        }

    }

    printf(**"**->The text has been succesfully copied to the new file.\n**"**);

    fclose(fp);

    fclose(pp);

**//** decrypt

    printf(**"**->The decrypted text is as follows:\n**"**);

    printf(**"**\"\"\"\n**"**);

    pp = fopen(**"**q3\_write.txt**"**, **"**r**"**);

    while ((ch = fgetc(pp)) != EOF)

    {

        sh = ch - 10;

        if ((ch != **'**\n**'**) && (ch != **'** **'**))

        {

            printf(**"**%c**"**, sh);

        }

        else

        {

            printf(**"**%c**"**, ch);

        }

    }

    printf(**"**\n\"\"\"**"**);

    printf(**"**\n**"**);

}

## Output

Text

Description automatically generated

## Files

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

# Given a text file, write a program to create another text file deleting the words 'a', 'the', 'an' and replacing each one of them with a blank space.

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

***int*** main()

{

    FILE \*rptr, \*wptr;

    rptr = fopen(**"**q4\_read.txt**"**, **"**r**"**);

    wptr = fopen(**"**q4\_write.txt**"**, **"**w**"**);

    if (rptr == NULL || wptr == NULL)

    {

        printf(**"**File not found\n**"**);

        return 0;

    }

    printf(**"**The contents of the file are being read and copied to a new file.\n**"**);

***char*** str[20];

    while (fscanf(rptr, **"**%s**"**, str) != EOF)

    {

        if (strcmp(str, **"**a**"**) == 0 || strcmp(str, **"**the**"**) == 0 || strcmp(str, **"**an**"**) == 0)

        {

            fputs(**"** **"**, wptr);

        }

        else

        {

            fputs(str, wptr);

            fputs(**"** **"**, wptr);

        }

    }

    printf(**"**The contents of the file have been succesfully copied to a new file.\n**"**);

    fclose(rptr);

    fclose(wptr);

}

## Output

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

# Write a program to carry out the following: A. Read a text file 'Input.Txt'. B. Print each word in reverse order

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

***int*** main()

{

    FILE \*fp;

    fp = fopen(**"**Input.txt**"**, **"**r**"**);

***char*** str[20];

    while (fscanf(fp, **"**%s**"**, str) != EOF)

    {

***int*** i = strlen(str) - 1;

        while (i >= 0)

        {

            printf(**"**%c**"**, str[i]);

            i--;

        }

        printf(**"** **"**);

    }

    printf(**"**\n**"**);

    fclose(fp);

}

## Output

Text

Description automatically generated

## Files

Graphical user interface, text, website

Description automatically generated

# Write down macro definitions for the following

## To find the arithmetic mean of two numbers.

### Code

#include **<**stdio.h**>**

#include **<**math.h**>**

#include **<**string.h**>**

#define a 10

#define b 20

***int*** main()

{

#ifdef a

#ifdef b

    printf(**"**The arithmetic mean of %d and %d is %d\n**"**, (a + b) / 2);

#endif

#endif

}

### Output

Text

Description automatically generated

## To find the absolute value of a number.

### Code

#include **<**stdio.h**>**

#include **<**math.h**>**

#include **<**string.h**>**

#define a -12

***int*** main()

{

#if (a >= 0)

    printf(**"**The absolute value is: %d**"**, a);

#else

    printf(**"**The absolute value is: %d**"**, a \* -1);

#endif

}

### Output

Text

Description automatically generated

## 3. To convert an uppercase alphabet to lowercase.

### Code

#include **<**stdio.h**>**

#include **<**math.h**>**

#include **<**string.h**>**

#define a -12

***int*** main()

{

#if (a >= 0)

    printf(**"**The absolute value is: %d**"**, a);

#else

    printf(**"**The absolute value is: %d**"**, a \* -1);

#endif

}

### Output

Text

Description automatically generated

## 4. To obtain the bigger of two numbers.

### Code

#include **<**stdio.h**>**

#include **<**math.h**>**

#include **<**string.h**>**

#define ch **'**A**'**

***int*** main()

{

#ifdef ch

***int*** r;

    r = ch + 32;

    printf(**"**The lowercase alphabet is: %c**"**, r);

#endif

}

### Output

Graphical user interface, text

Description automatically generated

# Write down macro definitions for the following. 1. To test whether a character entered is a small case letter or not. 2. To test whether a character entered is an upper case letter or not. 3. To test whether a character is an alphabet or not. Make use of the macros you defined in (1) and (2) above.

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

#define IS\_SMALL\_LETTER(c) ((c) >= **'**a**'** && (c) <= **'**z**'**)

#define IS\_UPPER\_LETTER(c) ((c) >= **'**A**'** && (c) <= **'**Z**'**)

#define IS\_ALPHABET(c) (IS\_SMALL\_LETTER(c) || IS\_UPPER\_LETTER(c))

***int*** main()

{

***char*** c;

    printf(**"**Enter a character: **"**);

    scanf(**"**%c**"**, &c);

    if (IS\_ALPHABET(c))

    {

        if (IS\_SMALL\_LETTER(c))

            printf(**"**%c is a small letter.\n**"**, c);

        else

            printf(**"**%c is an upper letter.\n**"**, c);

    }

    else

        printf(**"**%c is not an alphabet.\n**"**, c);

    return 0;

}

## Output

Text

Description automatically generated

# Create an enumerated data type logical with TRUE and FALSE values. Write a program to check whether the entered number is prime or not prime. If the number is prime display 0 otherwise 1. Use enumerated datatype

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

**enum** boolean

{

    FALSE,

    TRUE

};

**enum** boolean isPrime(***int*** n)

{

***int*** i;

    for (i = 2; i <= n / 2; i++)

    {

        if (n % i == 0)

            return FALSE;

    }

    return TRUE;

}

***int*** main()

{

***int*** n;

    printf(**"**Enter a number: **"**);

    scanf(**"**%d**"**, &n);

    if (isPrime(n) == TRUE)

        printf(**"**%d is a prime number.\n**"**, n);

    else

        printf(**"**%d is not a prime number.\n**"**, n);

    return 0;

}

## Output

Text

Description automatically generated

# Create an enumerated datatype for 12 months and display the values in the integer.

## Code

#include **<**stdio.h**>**

**enum** months

{

    JANUARY = 1,

    FEBRUARY,

    MARCH,

    APRIL,

    MAY,

    JUNE,

    JULY,

    AUGUST,

    SEPTEMBER,

    OCTOBER,

    NOVEMBER,

    DECEMBER

};

***int*** main()

{

**//** print all months

    printf(**"**%d\n**"**, JANUARY);

    printf(**"**%d\n**"**, FEBRUARY);

    printf(**"**%d\n**"**, MARCH);

    printf(**"**%d\n**"**, APRIL);

    printf(**"**%d\n**"**, MAY);

    printf(**"**%d\n**"**, JUNE);

    printf(**"**%d\n**"**, JULY);

    printf(**"**%d\n**"**, AUGUST);

    printf(**"**%d\n**"**, SEPTEMBER);

    printf(**"**%d\n**"**, OCTOBER);

    printf(**"**%d\n**"**, NOVEMBER);

    printf(**"**%d\n**"**, DECEMBER);

    return 0;

}

## Output

Graphical user interface, text

Description automatically generated

# Create a user-defined datatype from a structure. The structure should contain the variables such as name, regno, cgpa, and age of students. Use array of structures.

## Code

#include **<**stdio.h**>**

#include **<**string.h**>**

***struct*** student

{

***char*** name[20];

***int*** regno;

***float*** cgpa;

***int*** age;

};

***int*** main()

{

***struct*** student s[3];

***int*** i;

    for (i = 0; i < 3; i++)

    {

        printf(**"**Enter name: **"**);

        scanf(**"**%s**"**, s[i].name);

        printf(**"**Enter regno: **"**);

        scanf(**"**%d**"**, &s[i].regno);

        printf(**"**Enter cgpa: **"**);

        scanf(**"**%f**"**, &s[i].cgpa);

        printf(**"**Enter age: **"**);

        scanf(**"**%d**"**, &s[i].age);

    }

    printf(**"**\n**"**);

    for (i = 0; i < 3; i++)

    {

        printf(**"**Name: %s\n**"**, s[i].name);

        printf(**"**Regno: %d\n**"**, s[i].regno);

        printf(**"**Cgpa: %f\n**"**, s[i].cgpa);

        printf(**"**Age: %d\n**"**, s[i].age);

    }

    return 0;

}

## Output

Text

Description automatically generated with medium confidence