Advanced C Lab assignment 5

Name: Sankalp Mukim

Registration Number: 20BDS0128

Slot: L27+L28

Table of Contents

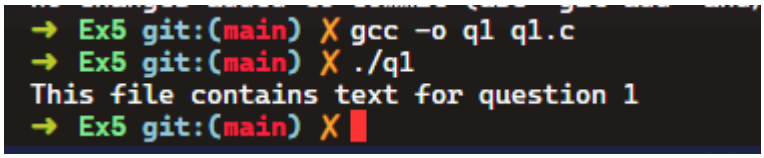
Write a program to read a file and display its contents along with line numbers before each line.

Code

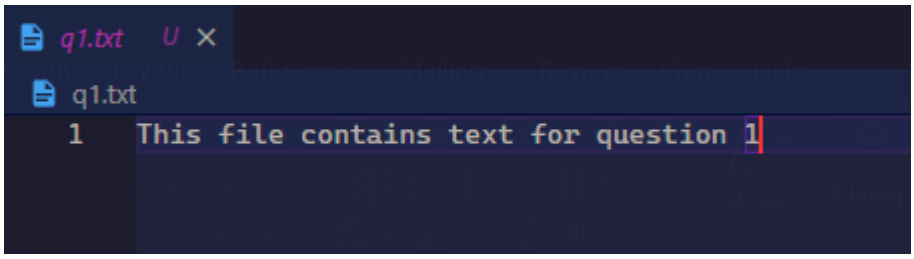
```c
#include <stdio.h>
#include <string.h>

int main()
{
    FILE *fp;
    char ch;
```

```c
    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



File



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

```c
#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)
    {
```

```c
        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

```
➜ Ex5 git:(main) X gcc q2.c -o q2
➜ Ex5 git:(main) X ./q2
The contents of the file are being read and copied to a new file.
The contents of the file have been succesfully copied to a new file.
➜ Ex5 git:(main) X
```

Files

q2_read.txt U

q2_read.txt

```
1   q2 read contents ALREADY_CAPITAL
```

q2_write.txt U

q2_write.txt

```
1   Q2 READ CONTENTS ALREADY_CAPITAL
```

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

```c
#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

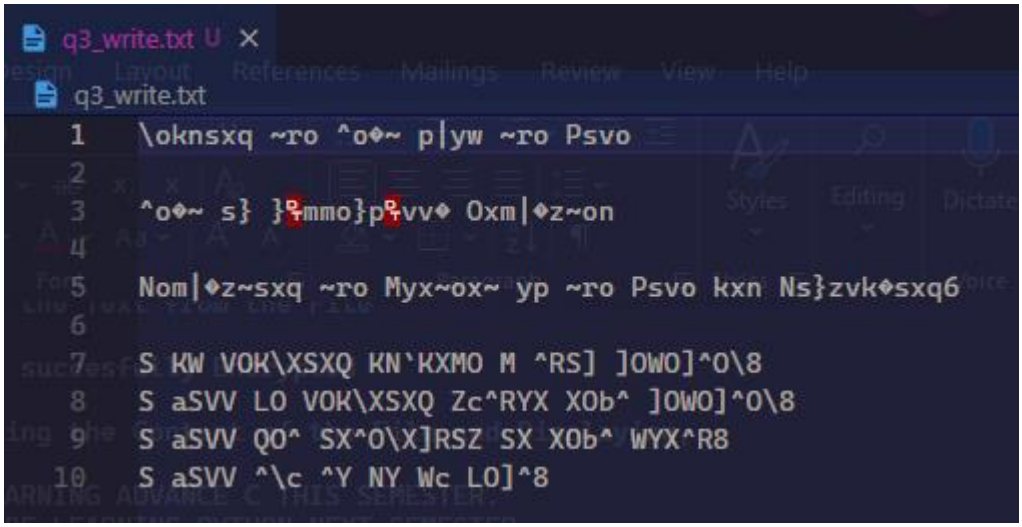
```
➜ Ex5 git:(main) ✗ gcc q3.c -o q3
➜ Ex5 git:(main) ✗ ./q3
->Reading the text from the file.
->The text has been succesfully copied to the new file.
->The decrypted text is as follows:
"""
Reading the Text from the File

Text is succesfully Encrypted

Decrypting the Content of the File and Displaying,

I AM LEARNING ADVANCE C THIS SEMESTER.
I WILL BE LEARNING PYTHON NEXT SEMESTER.
I WILL GET INTERNSHIP IN NEXT MONTH.
I WILL TRY TO DO MY BEST.
"""
➜ Ex5 git:(main) ✗
```

Files

q3_read.txt U

q3_read.txt

```
1  Reading the Text from the File
2
3  Text is succesfully Encrypted
4
5  Decrypting the Content of the File and Displaying,
6
7  I AM LEARNING ADVANCE C THIS SEMESTER.
8  I WILL BE LEARNING PYTHON NEXT SEMESTER.
9  I WILL GET INTERNSHIP IN NEXT MONTH.
10 I WILL TRY TO DO MY BEST.
```



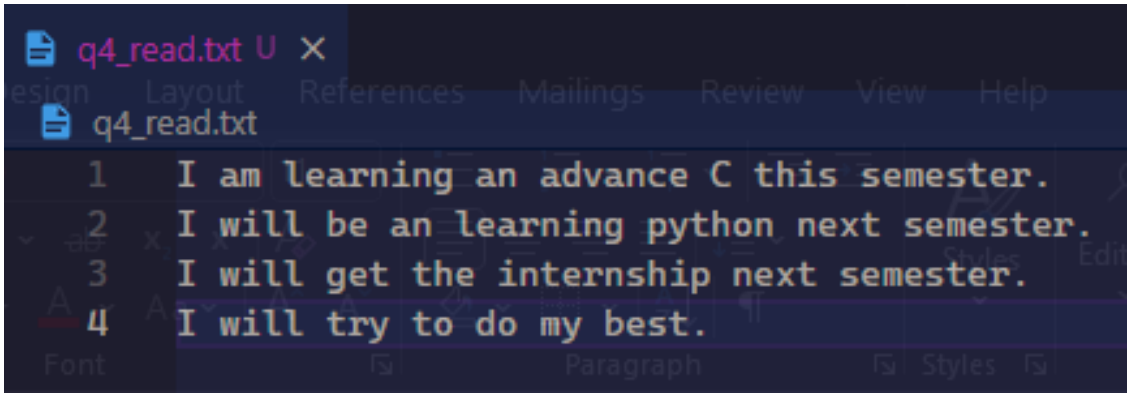
Given a text file, write a program to create another text file deleting the words 'a', 'the', 'an' and replacing each one of them with a blank space.

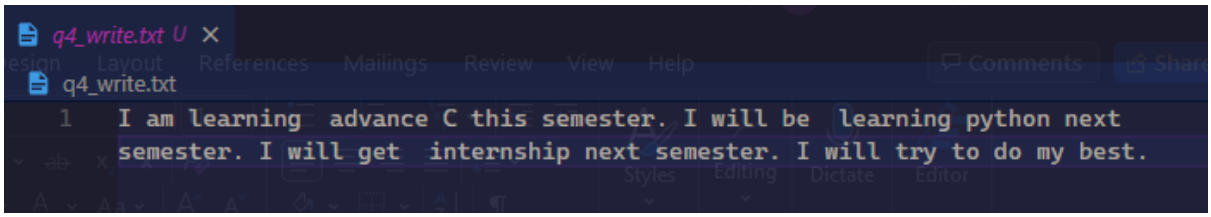
Code

```
#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





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

Code

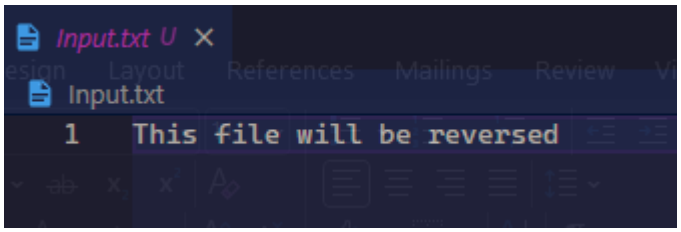
```
#include <stdio.h>
#include <string.h>

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

```
→ Ex5 git:(main) X gcc q5.c -o q5
→ Ex5 git:(main) X ./q5
sihT elif lliw eb desrever
→ Ex5 git:(main) X
```

Files



Write down macro definitions for the following

1. 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

```
→ Ex5 git:(main) X gcc q6a.c -o q6a
→ Ex5 git:(main) X ./q6a
The arithmetic mean of 10 and 20 is 15
```

2. 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

```
➜  Ex5 git:(main) ✗ gcc q6b.c -o q6b
➜  Ex5 git:(main) ✗ ./q6b
The absolute value is: 12%
➜  Ex5 git:(main) ✗
```

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

```
➜  Ex5 git:(main) ✗ gcc q6c.c -o q6c
➜  Ex5 git:(main) ✗ ./q6c
The lowercase alphabet is: a%
➜  Ex5 git:(main) ✗
```

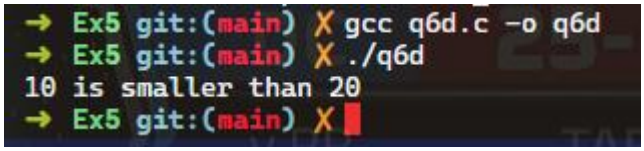
4. To obtain the bigger of two numbers.

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

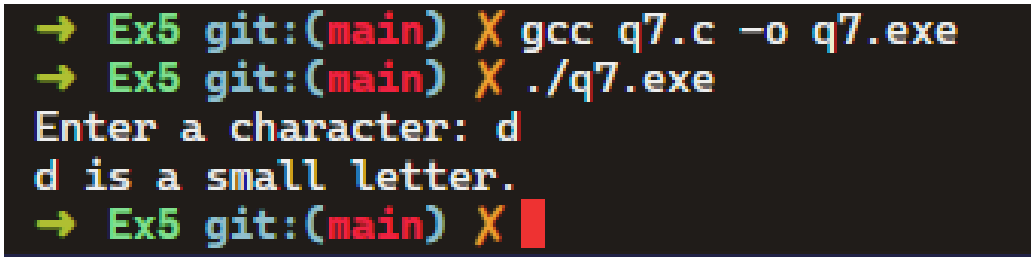


Write down macro definitions for the following. 1. To test whether a character entered is a small case letter or not. 2. To test whether a character entered is an upper case letter or not. 3. To test whether a character is an alphabet or not. Make use of the macros you defined in (1) and (2) above.

Code

```c
#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



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

```c
#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

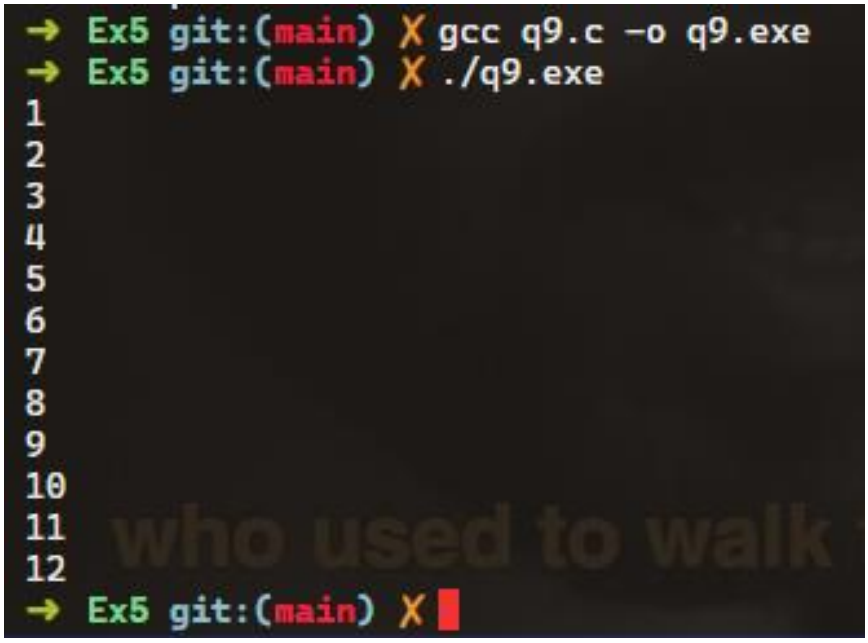
```
➜ Ex5 git:(main) X gcc q8.c -o q8.exe
➜ Ex5 git:(main) X ./q8.exe
Enter a number: 12
12 is not a prime number.
➜ Ex5 git:(main) X ./q8.exe
Enter a number: 7
7 is a prime number.
➜ Ex5 git:(main) X
```

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

Code

```c
#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



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

```c
#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: ");
```

```c
        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

```
➜ Ex5 git:(main) X gcc q10.c -o q10.exe
➜ Ex5 git:(main) X ./q10.exe
Enter name: sankalp
Enter regno: 21
Enter cgpa: 9.5
Enter age: 20
Enter name: samarth
Enter regno: 22
Enter cgpa: 9.6
Enter age: 20
Enter name: arjun
Enter regno: 23
Enter cgpa: 9.7
Enter age: 19

Name: sankalp
Regno: 21
Cgpa: 9.500000
Age: 20
Name: samarth
Regno: 22
Cgpa: 9.600000
Age: 20
Name: arjun
Regno: 23
Cgpa: 9.700000
Age: 19
➜ Ex5 git:(main) X
```