Lab Report

Course Name: Problem Solving and Programming Experiment

Student Name: 蒋云翔

Student ID: 2022102330

College: International School

Major: CST

Office of Academic Affairs, Jinan University

2023/06/18

LAB REPORT 实验报告

Lab Title	Lab_01				Lab No.	1
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	2023/2/24		

Lab description/objectives:

Task1:

Modify example program 2.9.

Convert a Celsius temperature into its Fahrenheit equivalent.

Task2:

Calculate $10^2+11^2+...+20^2$ by using formula $1^2+2^2+3^2+....+n^2 = n(n+1)(2n+1)/6$

Source code:

Task 1：

```
#include <stdio.h>
int main()
{
    double celsius = 23.89;
    double fahrenheit;
    fahrenheit = 9.0 * celsius / 5.0 + 32;
    printf("The Fahrenheit equivalent of %5.2f degrees Celsius\n",
        celsius);
    printf("   is %5.2f degrees Fahrenheit\n", fahrenheit);

    return 0;
}
```

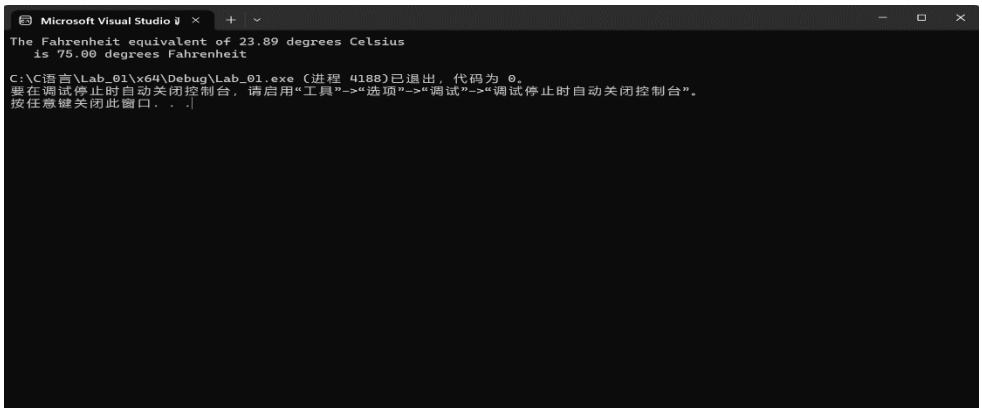
Task 2:

```
#include<stdio.h>

int calculate(int num)
{
    return (num * (num + 1) * (2 * num + 1)) / 6;
}
int main()
{
    int one_to_nine, one_to_twenty, ten_to_twenty;
    one_to_nine = calculate(9);
    one_to_twenty = calculate(20);
    ten_to_twenty = one_to_twenty - one_to_nine;
    printf("10^2 + 11^2 + …… + 20^2  equals %d", ten_to_twenty);
    return 0;
```

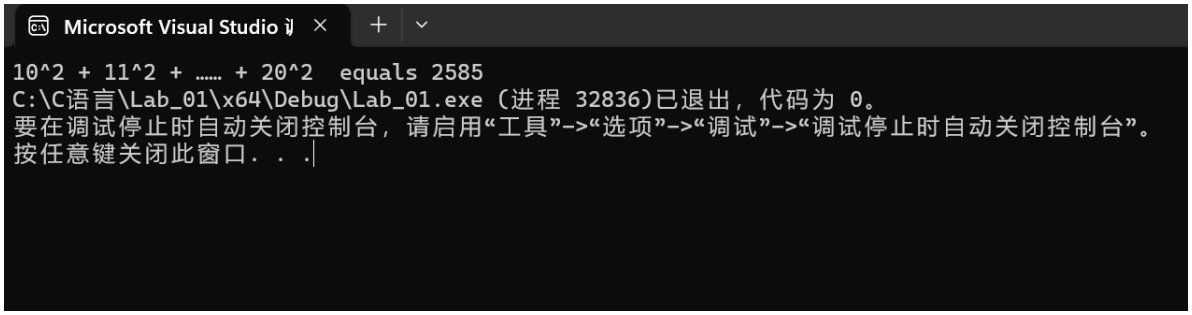
```
}
```

Program outputs:

Task 1:



Task 2:



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

It's easy to finish the task but it may take some times for me to simplify the code and make it clean and more visual for people.

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

None

Lab Title	Interactive Input and Selections				Lab No.	02
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	2023/3/3		

Lab description/objectives:

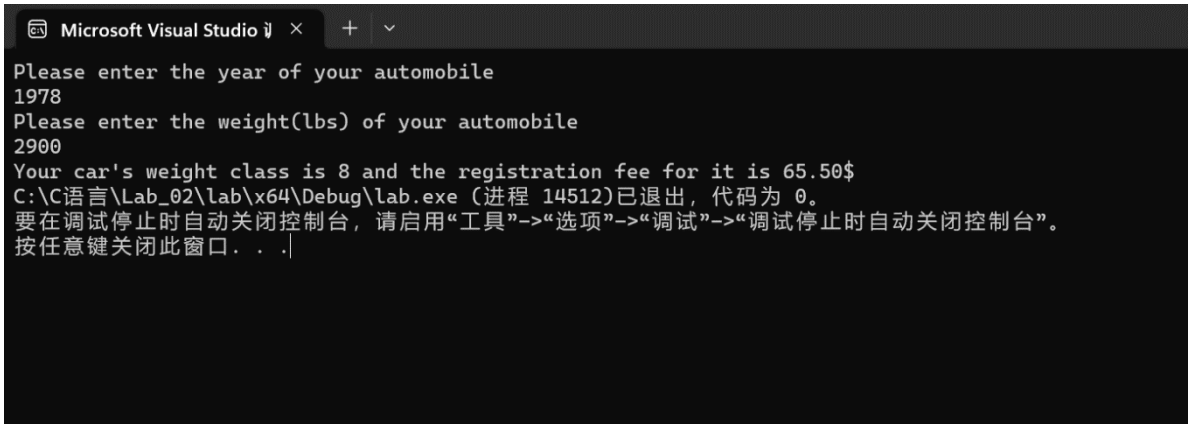
Learn Interactive Input and Selections

Source code:

```
#include <stdio.h>

int main()
{
    int year = 0;
    float weight = 0;
    printf("Please enter the year of your automobile\n");
    scanf("%d", &year);
    printf("Please enter the weight(lbs) of your automobile\n");
    scanf("%f", &weight);
    if (year <= 1970 && weight <= 2700)
        printf("Your car's weight class is 1 and the registration fee for it is 16.50$");
    else if (year <= 1970 && 2700 < weight && weight <= 3800)
        printf("Your car's weight class is 2 and the registration fee for it is 25.50$");
    else if (year <= 1970 && weight > 3800)
        printf("Your car's weight class is 3 and the registration fee for it is 46.50$");
    else if (1971 <= year <= 1979 && weight <= 2700)
        printf("Your car's weight class is 4 and the registration fee for it is 27.00$");
    else if (1971 <= year <= 1979 && 2700 < weight && weight <= 3800)
        printf("Your car's weight class is 5 and the registration fee for it is 30.50$");
    else if (1971 <= year <= 1979 && weight > 3800)
        printf("Your car's weight class is 6 and the registration fee for it is 52.50$");
    else if (year >= 1980 && weight <= 3500)
        printf("Your car's weight class is 7 and the registration fee for it is 35.50$");
    else if(year >= 1980 && weight > 3500)
        printf("Your car's weight class is 8 and the registration fee for it is 65.50$");
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

How to simplify the code and how to make it clear and beautiful.

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	**Funtions**			Lab No.	**03**
Stud. Name	Jiang Yunxiang	Major	**CST**	Class	
Student ID	**2022102330**	Date	2023/3/10		

Lab description/objectives:

Using this information write and test two functions

Source code:

```
#include <stdio.h>
int det2(int a11, int a12, int a21, int a22)
{
    return a11 * a22 - a21 * a12;
}

int det3(int a11, int a12, int a13, int a21, int a22, int a23, int a31, int a32, int a33)
{
    int result = 0;
    result = a11 * det2(a22, a23, a32, a33) - a21 * det2(a12, a13, a32, a33) + a31 *
det2(a12, a13, a22, a23);
    return result;
}

int main()
{
    printf("%d\n", det2(1, 2, 3, 4));
    printf("%d", det3(1, 2, 3, 4, 5, 6, 7, 8, 9));

    return 0;
}}
```

Program outputs:

（Paste the screenshot of program output here）

Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

How to simplify the code?

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	**Pointers**			Lab No.	**4**
Stud. Name	Jiang Yunxiang	Major	CST	Class	
Student ID	2022102330	Date	3/17		

Lab description/objectives:

Using pointers

Source code:

（Paste the source code here）

```
#include<stdio.h>

void date(int data, int* month, int* day, int* year)
{
    *day = data % 100;
    *month = data % 10000 / 100;
    *year = data / 10000;
}

int main()
{
    int month, day, year;
    date(20220411, &month, &day, &year);
    printf("%d\n%d\n%d\n",year,month,day);
```

```
    return 0;

}
```

Program outputs:

（Paste the screenshot of program output here）

Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

How to get the corresponding number of the data.

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	String				Lab No.	5
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	3/24		

Lab description/objectives:

Write a program that asks the user to input a string, the program will count and output the maximum number of consecutive repeating characters in the string.

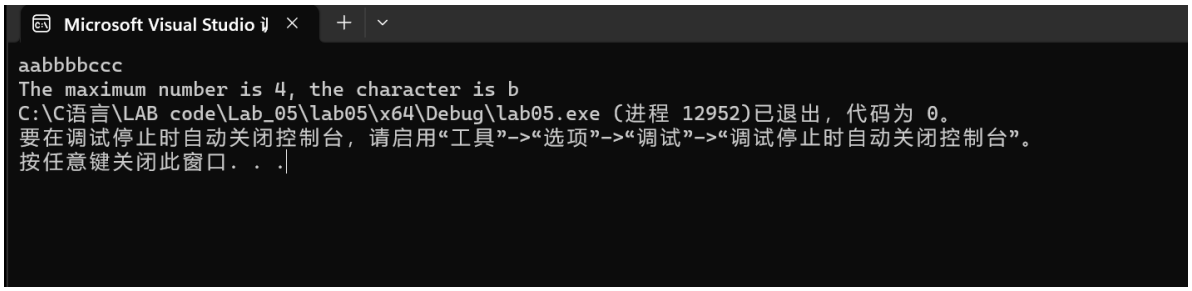
Source code:

（Paste the source code here）

```
#include<stdio.h>
#include<string.h>
#define N 20
int main()
{
    int maxnum=0, maximum = 0, num = 1;
    char str[N];
    gets(str);
    for (int i = 0; i < strlen(str); i++)
    {
        if (str[i] == str[i + 1])
        {
            num += 1;
        }
        else num = 1;
        if (num > maxnum)
        {
            maximum = str[i];
            maxnum = num;
        }
    }
    printf("The maximum number is %d, the character is %c", maxnum, maximum);
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

To simplify the source code

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Data Files			Lab No.	6
Stud. Name	Jiang Yunxiang	Major	CST	Class	
Student ID	2022102330	Date	3/31		

Lab description/objectives:

Write a C program to read data from a text file containing multiple lines of text (created with Notepad, assuming that the maximum length of each line is 100 characters), and write the even-numbered lines of text into a file, and write the odd-numbered lines of text into another file. (Hints: use fgets() and fputs())

Source code:

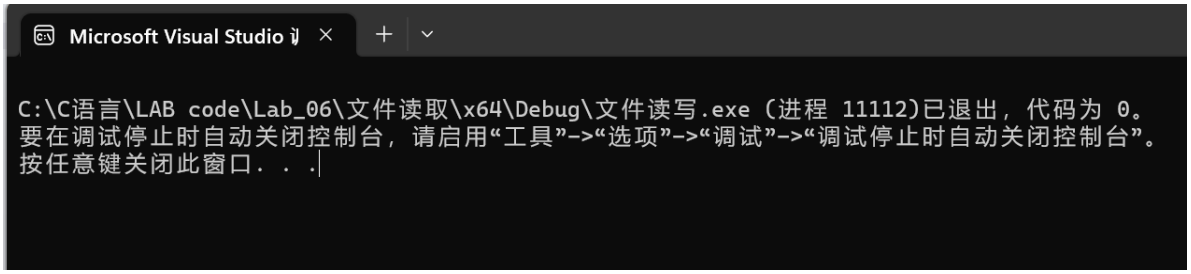
（Paste the source code here）

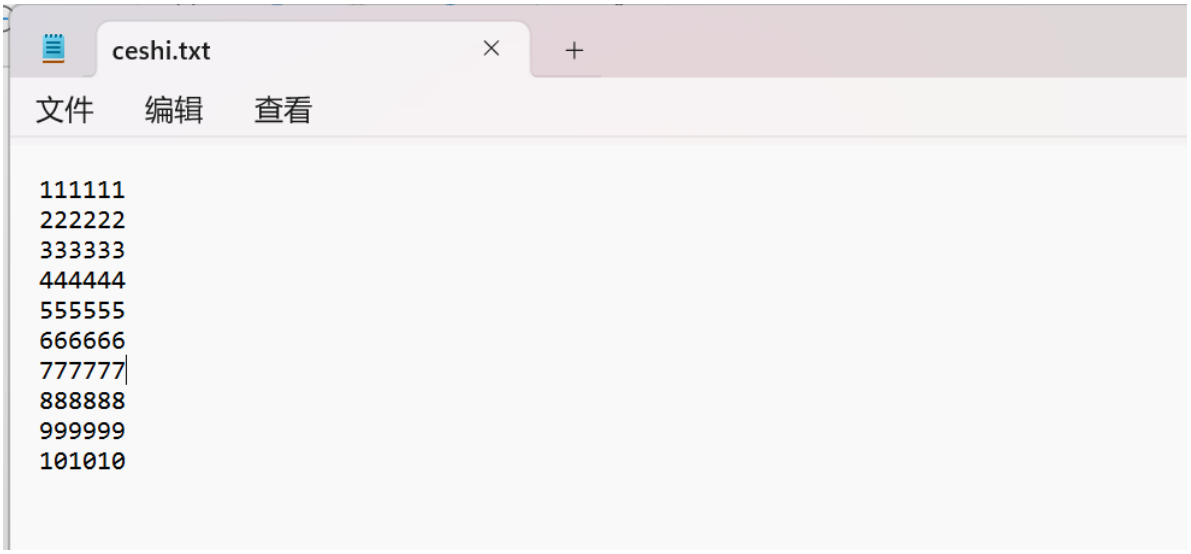
```
#include <stdio.h>
#include <stdlib.h>
#define Max 1010
int main()
{
    char Odd[Max];
    char Even[Max];
    FILE* file1, * file2, * file3;
    file1 = fopen("ceshi.txt", "r");
    file2 = fopen("oddfile.txt", "w");
    file3 = fopen("evenfile.txt", "w");
    while (fgets(Odd, 10, file1) != NULL) {
        fgets(Even, 10, file1);
        fputs(Even, file3);
```

```
        fputs(Odd, file2);
    }

    fclose(file1);
    fclose(file2);
    fclose(file3);
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）





Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Array of Pointers				Number	07
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	4/7		

Lab description/objectives:

Apr 7, 2023, is Friday. The user enters a date (1-30) in April, and your program will output the corresponding day of the week. Use an array of pointers to store the names of the days of the week.

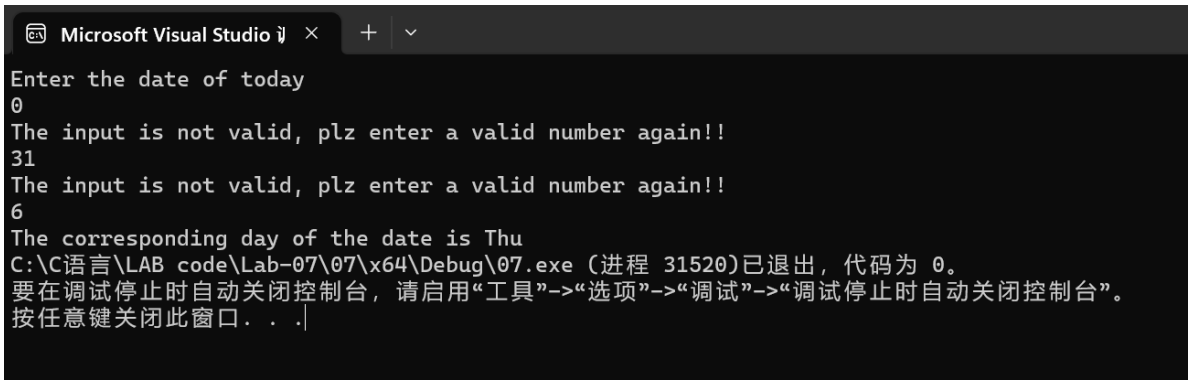
Source code:

（Paste the source code here）

```c
#include <stdio.h>

int main()
{
    char* arr[] = { "Fri", "Sat",  "Sun",  "Mon",  "Tue",  "Wed",  "Thu", };
    int date;
    printf("Enter the date of today\n");
    scanf("%d", &date);
    while (1) {
        int corresponding_day = date % 7;
        if (date >= 1 && date < 31) {
            printf("The corresponding day of the date is %s", *(arr + corresponding_day));
            break;
        }
        else {
            printf("The input is not valid, plz enter a valid number again!!\n");
            scanf("%d", &date);
        }
    }

    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Doubly Linked List				Lab No.	08
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	4/14		

Lab description/objectives:

a) A doubly linked list is a list in which each structure contains a pointer to both the following and previous structures in the list. Define an appropriate template for a doubly linked list of names and telephone numbers.

b) Using the template defined in a), modify Program 13.3 pgm13-3.c to list the names and phone numbers in reverse order.

Source code:

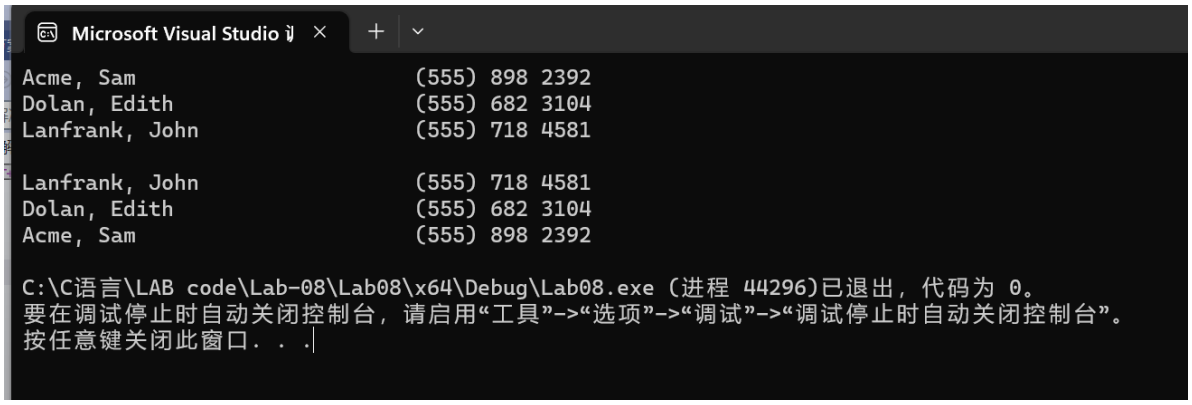
（Paste the source code here）

```
#include <stdio.h>
#define MAXNAME 30
#define MAXPHONE 15

struct TeleType {
    char name[MAXNAME];
    char phoneNum[MAXPHONE];
    struct TeleType* next;
    struct TeleType* prior;
};

void display(struct TeleType* contents)
{
    while (contents != NULL) {
```

```
        printf("%-30s %-20s\n", contents->name, contents->phoneNum);
        contents = contents->next;
    }
}

void reverse_display(struct TeleType* contents)
{
    while (contents != NULL) {
        printf("%-30s %-20s\n", contents->name, contents->phoneNum);
        contents = contents->prior;
    }
}

int main()
{
    struct TeleType t1 = { "Acme, Sam", "(555) 898 2392" };
    struct TeleType t2 = { "Dolan, Edith", "(555) 682 3104" };
    struct TeleType t3 = { "Lanfrank, John", "(555) 718 4581" };
    struct TeleType* first;
    struct TeleType* last;

    first = &t1; //定义第一个指针
    t1.next = &t2;
    t2.next = &t3;
    t3.next = NULL;

    last = &t3; //定义第二个指针
    t3.prior = &t2;
    t2.prior = &t1;
    t1.prior = NULL;

    display(first);
    printf("\n");
    reverse_display(last);

    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	**Dynamic Linked List**				Lab No.	**09**
Stud. Name	**Jiang Yunxiang**	Major	**CST**		Class	
Student ID	**2022102330**		Date	**4/21**		

Lab description/objectives:

Add two functions printFor(), printRev() to Program 13.7 pgm13-7.c that print elements of the linked list in forward, and reverse order using recursion.

Source code:

（Paste the source code here）

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXCHARS 30
#define DEBUG 0

/* here is the declaration of a linked list structure */
struct NameRec {
    char name[MAXCHARS];
    struct NameRec* nextAddr;
};

/* here is the definition of the first structure pointer */
struct NameRec* firstRec;

int main()
{
    void printFor();
    void printRev();
    void readInsert(); /* function prototypes */
    void display();

    firstRec = NULL; /* initialize list pointer */
    readInsert();
    printf("The print elements of the linked list in forward:\n");
    printFor(firstRec);
    printf("The print elements of the linked list in reverse:\n");
    printRev(firstRec);

    return 0;
}

/* get a name and insert it into the linked list */
void readInsert()
{
    char name[MAXCHARS];
    void insert(char*);

    printf("\nEnter as many names as you wish, one per line");
    printf("\nTo stop entering names, enter a single x\n");
    while (1) {
```

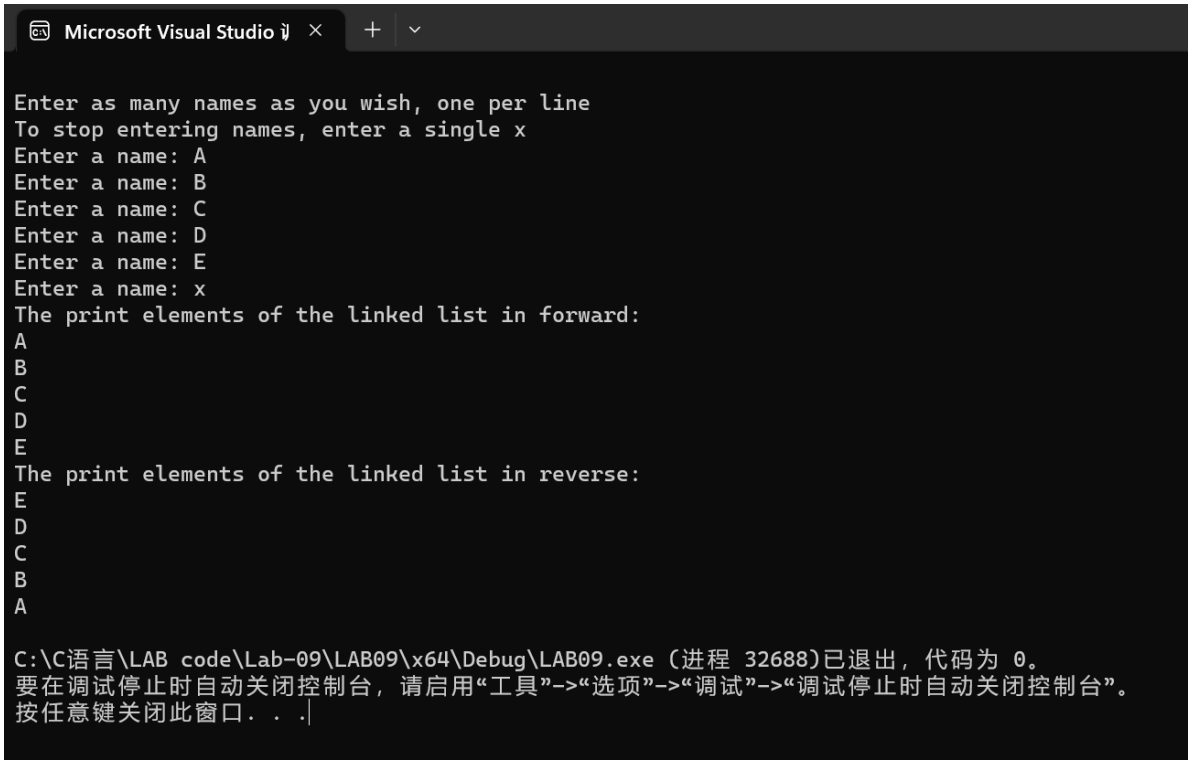
```
        printf("Enter a name: ");
        gets(name);
        if (strcmp(name, "x") == 0)
            break;
        insert(name);
    }
}

void insert(char* name)
{
    struct NameRec* linearLocate(char*); /* function prototype */
    struct NameRec* newaddr, * here; /* pointers to structure */
    /* of type NameRec */

    newaddr = (struct NameRec*)malloc(sizeof(struct NameRec));
    if (newaddr == (struct NameRec*)NULL) /* check the address */
    {
        printf("\nCould not allocate the requested space\n");
        exit(1);
    }

    /* locate where the new structure should be placed and */
    /* update all pointer members */
    if (firstRec == NULL) /* no list currently exists */
    {
        newaddr->nextAddr = NULL;
        firstRec = newaddr;
    }
    else if (strcmp(name, firstRec->name) < 0) /* a new first structure */
    {
        newaddr->nextAddr = firstRec;
        firstRec = newaddr;
    }
    else /* structure is not the first structure of the list */
    {
        here = linearLocate(name);
        newaddr->nextAddr = here->nextAddr;
        here->nextAddr = newaddr;
    }

    strcpy(newaddr->name, name); /* store the name */
}

/* This function locates the address of where a new structure
```

```
   should be inserted within an existing list.
   It receives the address of a name and returns the address of a
   structure of type NameRec
*/
struct NameRec* linearLocate(char* name)
{
    struct NameRec* one, * two;
    one = firstRec;
    two = one->nextAddr;

    if (two == NULL)
        return (one); /* new structure goes after the existing single structure */
    while (1) {
        if (strcmp(name, two->name) < 0) /* if it is located within the list */
            break;
        else if (two->nextAddr == NULL) /* it goes after the last structure */
        {
            one = two;
            break;
        }
        else /* more structures to search against */
        {
            one = two;
            two = one->nextAddr;
        }
    } /* the break takes us here */

    return (one);
}

/* display names from the linked list */
void display()
{
    struct NameRec* contents;

    contents = firstRec;
    printf("\nThe names currently in the list, in alphabetical");
    printf("\norder, are:\n");
    while (contents != NULL) /* display till end of list */
    {
        printf("%s\n", contents->name);
        contents = contents->nextAddr;
    }
}
```

```c
void printFor(struct NameRec* p)
{
    if (p == NULL) {
        return;
    }
    printf("%s\n", p->name);
    printFor(p->nextAddr);
}

void printRev(struct NameRec* p)
{
    if (p == NULL) {
        return;
    }
    printRev(p->nextAddr);
    printf("%s\n", p->name);
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Common-line Argument			Lab No.	10
Stud. Name	Jiang Yunxiang	Major	CST	Class	
Student ID	2022102330	Date	4/28		

Lab description/objectives:

*Write a calculator program **calculator.c** that accepts an arithmetic expression as command-line arguments, e.g. 22 + 33. The program should perform the corresponding calculation, and display the result: 22 + 33 = 55. There are 5 types of operations: 44 + 55 for addition; 44 - 55 for subtraction; 44 x 55 for multiplication; 44 / 22 for division; 44 % 33 for modulus.*

The program should at least check if the correct number of arguments are passed. If not, it displays a usage message and exits.

Source code:

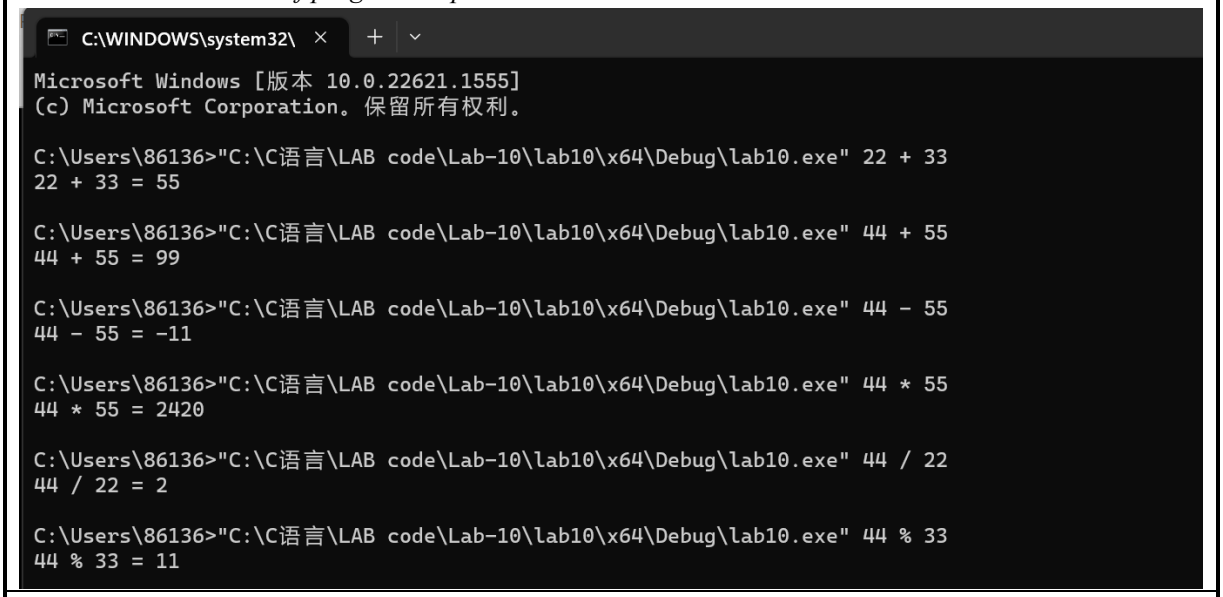
（Paste the source code here）

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

int main(int argc, char* argv[]) {
    if (argc != 4) { // check if the correct number of arguments are passed
        printf("Usage: calculator operand1 operator operand2\n");
        exit(1);
    }

    float num1 = atof(argv[1]); // convert first operand to a float
    float num2 = atof(argv[3]); // convert second operand to a float
    char* op = argv[2]; // get the operator
```

```
    float result;

    if (strcmp(op, "+") == 0) { // addition
        result = num1 + num2;
    }
    else if (strcmp(op, "-") == 0) { // subtraction
        result = num1 - num2;
    }
    else if (strcmp(op, "x") == 0 || strcmp(op, "*") == 0) { // multiplication
        result = num1 * num2;
    }
    else if (strcmp(op, "/") == 0) { // division
        if (num2 == 0) {
            printf("Error: division by zero\n");
            exit(1);
        }
        result = num1 / num2;
    }
    else if (strcmp(op, "%") == 0) { // modulus
        if (num2 == 0) {
            printf("Error: division by zero\n");
            exit(1);
        }
        result = (int)num1 % (int)num2;
    }
    else { // invalid operator
        printf("Invalid operator: %s\n", op);
        exit(1);
    }

    printf("%g %s %g = %g\n", num1, op, num2, result); // display the result

    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Bit-masking			Lab No.	11
Stud. Name	Jiang Yunxiang	Major	CST	Class	
Student ID	2022102330	Date	5/5		

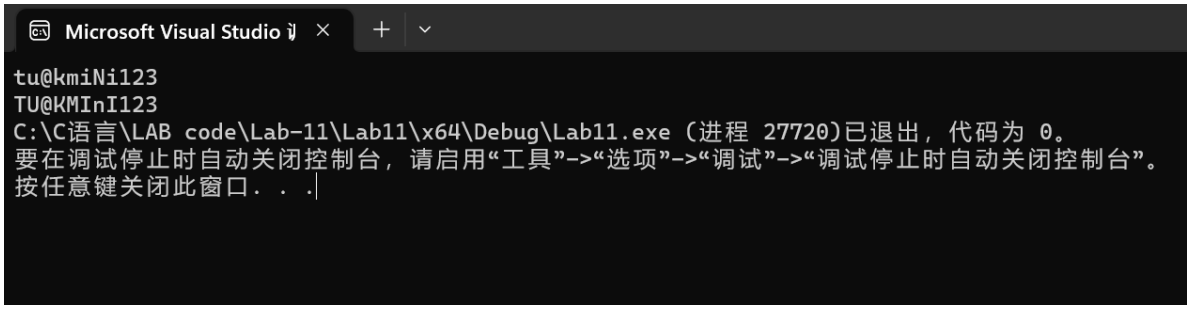
Lab description/objectives:

Investigate the difference between the binary representation of alphabets' uppercase and their lowercase, for instance, 'a' is 01100001, 'A' is 01000001, 'z' is 01111010, 'Z' is 01011010. Write functions, upperToLower(), lowerToUpper(), that convert between lowercase and uppercase by using bitwise operations.

```
#include <stdio.h>

void upperToLower(char *a)
{
int mask1 = 0b00100000;
*a = *a | mask1;
}

void lowerToUpper(char *b)
{
int mask2 = 0b11011111;
*b = mask2 & *b;
}

void toggle(char str[])
{
for (int i = 0; str[i] != '\0'; i++)
{
    if (str[i] >= 'a' && str[i] <= 'z')
    {
        lowerToUpper(&str[i]);
    }
    else if (str[i] >= 'A' && str[i] <= 'Z')
    {
        upperToLower(&str[i]);
    }
}
}

int main()
{
    char str[105];
    gets(str);
    toggle(str);
    printf("%s", str);
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

How to Call the two functions effectively

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Function overloading			Lab No.	12
Stud. Name	Jiang Yunxiang	Major	CST	Class	
Student ID	2022102330	Date	5/12		

Lab description/objectives:

Using function overloading, write a function area( ) to calculate the area of a circle, a rectangular, and a trapezoid with one, two and three arguments. Write another function max( ) to find the larger or longer one from two ints or two doubles or two strings.

Source code:

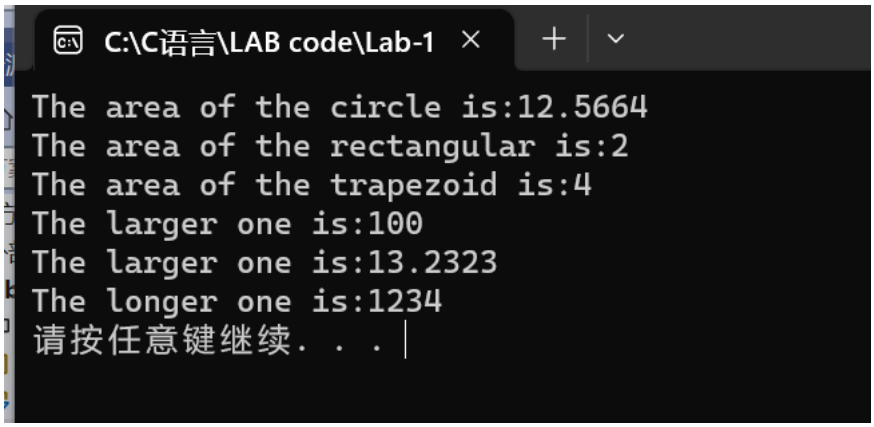
（Paste the source code here）

```
#include<iostream>
using namespace std;
#define pai acos(-1)
void area(float a)
{
```

```
    cout << "The area of the circle is:" << pai * a * a << endl;
}

void area(float a, float b)
{
    cout << "The area of the rectangular is:" << a*b << endl;
}

void area(float top, float bottom, float h)
{
    cout << "The area of the trapezoid is:" << (top+bottom)*h /2 << endl;
}

void max(int a, int b)
{
    if (a > b) cout << "The larger one is:" << a << endl;
    else cout << "The larger one is:" << b << endl;
}

void max(double a, double b)
{
    if (a > b) cout << "The larger one is:" << a << endl;
    else cout << "The larger one is:" << b << endl;
}

void max(string a, string b)
{
    if (a.length() > b.length()) cout << "The longer one is:" << a << endl;
    else cout << "The longer one is:" << b << endl;
}

int main()
{
    area(2);
    area(1,2);
    area(2,2,2);
    max(1, 100);
    max(12.11,13.2323);
    max("ABC","1234");
    system("pause");
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	**Operator Overloading in C++**			Lab No.	**13**
Stud. Name	Jiang Yunxiang	Major	**CST**	Class	
Student ID	**2022102330**	Date	**5/19**		

Lab description/objectives:

Write a C++ program to overload the + operator, to add two Time objects.

1. Define the Time class which should have three attributes of int data type: hour, minute, second, with 0 as their default value.

2. Define a member function showTime() to show the Time object as 11:45:33

3. Overload the + operator, to get the summation of two Time objects with the correct forms: hour(0-23), minute(0-59), second(0-59). For example, t1 is 1:50:30 , t2 is 2:15:25, then t1 + t2 will be 4:5:55

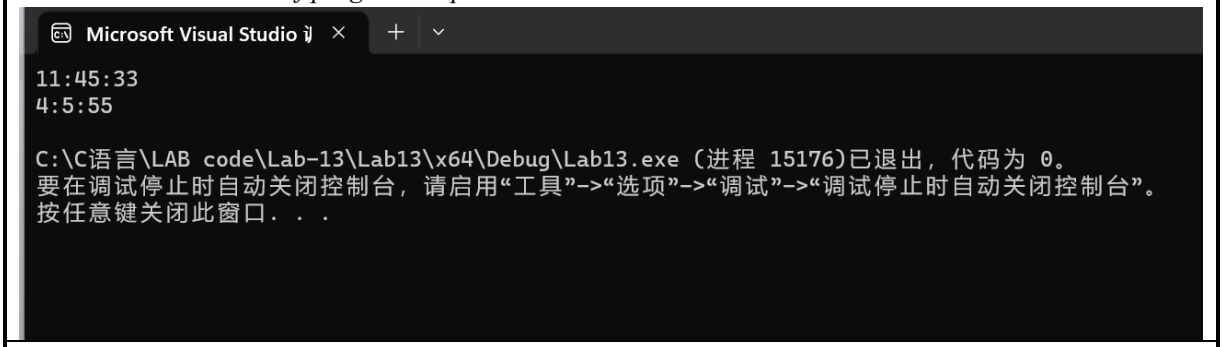
Source code:

（Paste the source code here）

```
#include<iostream>
using namespace std;

class Time {
public:
    int hour, minute, second;
    Time(int h = 0, int m = 0, int s = 0) :hour(h), minute(m), second(s) {}
    void showTime() {
        cout << hour << ":" << minute << ":" << second << endl;
    }
    Time operator+(Time& t2) {
        int h = hour + t2.hour;
        int m = minute + t2.minute;
        int s = second + t2.second;
        if (s >= 60) {
            s -= 60;
            m++;
        }
        if (m >= 60) {
            m -= 60;
            h++;
        }
        h = h % 24;
        return Time(h, m, s);
    }
};

int main()
{
    Time t0(11, 45, 33);
    Time t1(1, 50, 30);
    Time t2(2, 15, 25);
    Time t3 = t1 + t2;
    t0.showTime();
    t3.showTime();
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	C++ Templates				Lab No.	14
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	5/26		

Lab description/objectives:

Create the C++ Function Template named sumMultiPowers() so that it has three parameters sum, x, and n. The first two parameters will have the type represented by the function template type parameter T. n will always be int. The return type is void. All parameters are passed by value except for sum which is passed by reference. A Template Function created from sumMultiPowers() will compute

sum = 1 + x + 2x^2 + 3x^3 + ... + nx^n

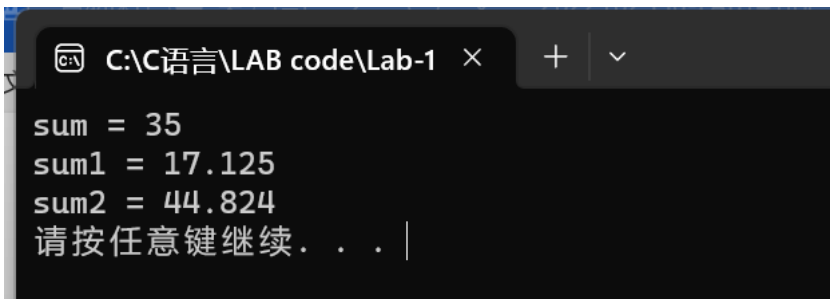
Source code:

（Paste the source code here）

```
#include<iostream>
using namespace std;
template <typename t>

void sumMutiPowers(t &sum, t x, int n)
{
    sum = 1;
    for (int i = 0; i <= n; i++)
    {
        sum = sum + i * pow(x, i);
    }
}
int main()
{
    int sum;
    sumMutiPowers(sum, 2, 3);
    cout << "sum = " << sum << endl;
    float sum1;
    sumMutiPowers(sum1,(float)1.5, 3);
    cout << "sum1 = " << sum1 << endl;
    double sum2;
    sumMutiPowers(sum2, 2.2, 3);
    cout << "sum2 = " << sum2 << endl;

    system("pause");
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（*What were the most difficult parts of your program to implement?*）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	C++ Files				Lab No.	15
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	6/2		

Lab description/objectives:

Write a C++ program that implements flipping the contents of each line of a string in a text file and outputs the flipped result to another text file. Both the input file name and output file name are specified by the user.

Source code:

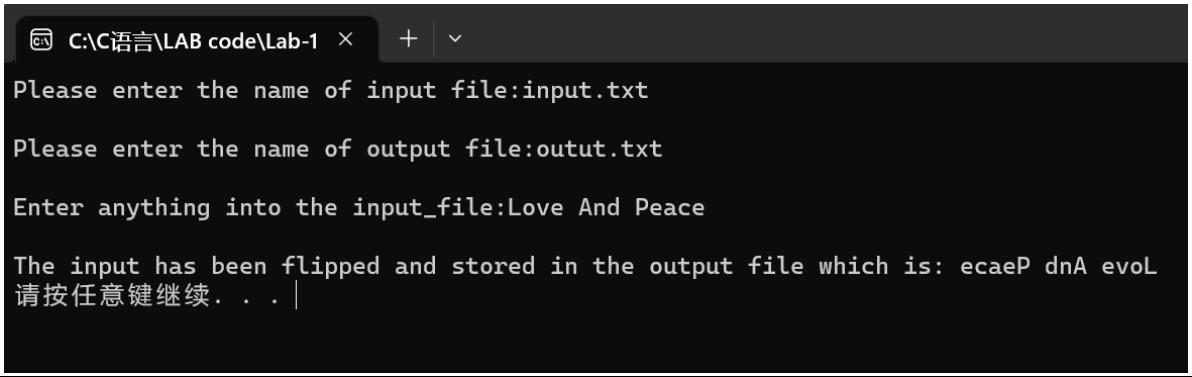
（*Paste the source code here*）

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;

int main()
{
    string input_file, output_file;
    cout << "Please enter the name of input file:";
    cin >> input_file;
    cout << endl;
    cout << "Please enter the name of output file:";
    cin >> output_file;
    cout << endl;
```

```
    ofstream input(input_file);
    ofstream output(output_file);
    cout << "Enter anything into the input_file:";
    string content;
    getline(cin, content);
    getline(cin, content);
    cout << endl;
    input << content;
    input.close();
    ifstream input1(input_file);
    string getcontent;
    while (getline(input1, getcontent))
    {
        reverse(getcontent.begin(), getcontent.end());
        output << getcontent;
    }
    cout << "The input has been flipped and stored in the output file which is: ";
    output.close();
    ifstream output1(output_file);
    string getcontent1;
    while (getline(output1, getcontent1))
    {
        cout << getcontent1<<endl;
    }

    input1.close();
    output1.close();
    system("pause");
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）



Discussion:

1. **Most difficult parts**

（*What were the most difficult parts of your program to implement?*）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope

LAB REPORT 实验报告

Lab Title	Final Lab				Lab No.	16
Stud. Name	Jiang Yunxiang	Major	CST		Class	
Student ID	2022102330		Date	6/9		

Lab description/objectives:

A palindrome string is a string that reads the same backward as forward. For example, "racecar" is a palindrome string because it reads the same way from left to right and from right to left. Another example is "level".

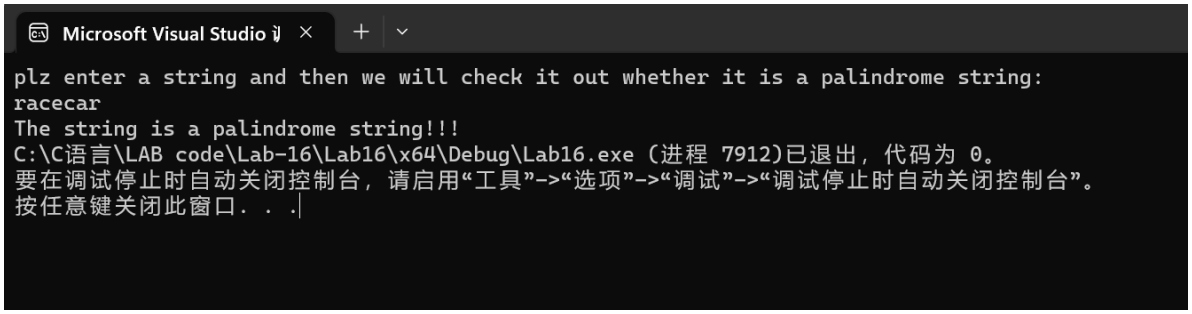
Write a C program to check whether a given string is a palindrome or not.

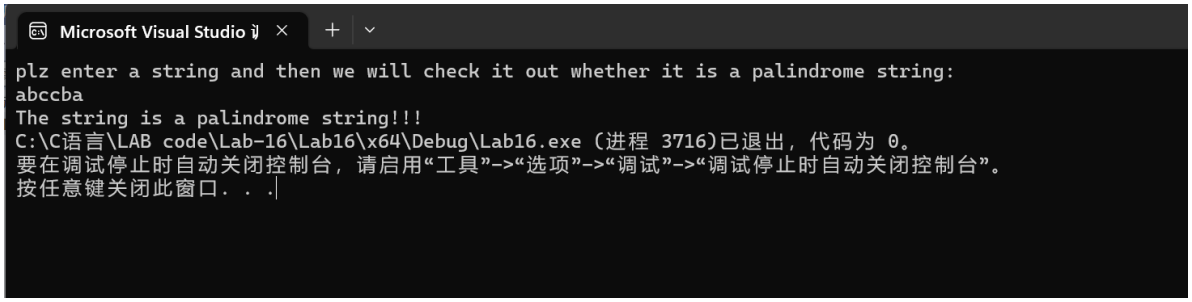
```
#include<stdio.h>
#define MAX 1005
#include<string.h>
#include<stdlib.h>
int main()
{
    printf("plz enter a string and then we will check it out whether it is a palindrome string:\n");
    char str[MAX] = "";
    scanf("%s", str);
    int j = strlen(str)-1;
    int k = 0;
    do
    {
        if (str[k] != str[j])
        {
            printf("The string is not a palindrome string!");
```

```
            exit(0);
        }
        else if (k > j)
        {
            printf("The string is a palindrome string!!!");
            exit(0);
        }
        else
        {
            k++;
            j--;
        }
    } while (str[k] == str[j]);
    return 0;
}
```

Program outputs:

（Paste the screenshot of program output here）





Discussion:

1. **Most difficult parts**

（What were the most difficult parts of your program to implement?）

Nope

2. **Bugs and/or Errors**

(List all the program bugs/errors you encountered and how you corrected them.)

Nope