2021-2022 Cprog(LAB)_Homework

Author: H3Art

LAB01: Data Types and Variables

Task1:

Modify example Program 2.9 in section 2.6 to convert a Celsius temperature into its Fahrenheit equivalent.

Example program code: Csource.rar

```c
/* convert a Celsius temperature to Fahrenheit */
#include <stdio.h>

int main() {
    double celsius = 25; /* declaration and initialization */
    double fahrenheit;
    fahrenheit = 9.0 / 5.0 * celsius + 32.0;
    printf("The Fahrenheit equivalent of %5.2f degrees Celsius\n",
celsius);
    printf("is %5.2f degrees\n", fahrenheit);
    return 0;
}
```

Task2:

By using the formula $1^2 + 2^2 + 3^2 + ... + n^2 = n(n + 1)(2n + 1)/6$, write a C program to calculate: $10^2 + 11^2 + ... + 20^2$.

```c
/* calculate 10^2+11^2+12^2+……+20^2 */
#include <stdio.h>

int main() {
    int out = 0;
    int n1 = 20;
    int n2 = 9;
    out = n1 * (n1 + 1) * (2 * n1 + 1) / 6 - n2 * (n2 + 1) * (2 * n2 + 1)
/ 6;
    printf("%d\n", out);
    return 0;
}
```

LAB02: Interactive Input and Formatted Output

When a particular rubber ball is dropped from a given height (in meters), its impact speed (in meters/second) when it hits the ground is given by the formula speed = sqrt(2 * g * height). The ball then rebounds to 2/3 the height from which it last fell. Using this information write, test, and run a C program that calculates and displays the impact speed of the first three bounces and the rebound height of each bounce.

Test your program using an initial height of 2.0 meters.Run the program twice and compare the results for dropping the ball on Earth (g = 9.81 meters per sec^2) and on the moon (g = 1.67 meters per sec^2).

The values of g and initial height should be input from keyboard. Display the results with 3 decimal places.

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

int main() {
    double g, height, speed;
    scanf("%lf%lf", &g, &height);
    for (int i = 0; i < 3; i++) {
        speed = sqrt(2 * g * height);
        height *= (2.0 / 3.0);
        printf("time %d:\nspeed=%.3lf\nheight=%.3lf\n\n", i + 1, speed,
height);
    }
    return 0;
}
```

LAB03: if-else-if statement

Based on an automobile's model year and weight, the state of New Jersey determines the car's weight class and registration fee using the following schedule:

Model Year Registration	Weight	Weight Class	Fee
1970 or earlier	less than 2,700 ,lbs	1	$16.50

Model Year Registration	Weight	Weight Class	Fee
1970 or earlier	2,700 to 3,800 lbs	2	$25.50
1970 or earlier	more than 3,800 lbs	3	$46.50
1971 to 1979	less than 2,700 lbs	4	$27.00
1971 to 1979	2,700 to 3,800 lbs	5	$30.50
1971 to 1979	more than 3,800 lbs	6	$52.50
1980 or later	less than 3,500 lbs	7	$35.50
1980 or later	3,500 or more lbs	8	$65.50

Using this information, write a C program that accepts the year and weight of an automobile and determines and displays the weight class and registration fee for the car.

```c
#include <stdio.h>

int main() {
    int year, weight;
    scanf("%d %d", &year, &weight);
    if (year <= 1970) {
        if (weight < 2700) {
            printf("%d $%.2f", 1, 16.50);
        } else if (weight <= 3800) {
            printf("%d $%.2f", 2, 25.50);
        } else {
            printf("%d $%.2f", 3, 46.50);
        }
    } else if (year > 1970 && year <= 1979) {
        if (weight < 2700) {
            printf("%d $%.2f", 4, 27.00);
        } else if (weight <= 3800) {
            printf("%d $%.2f", 5, 30.50);
        } else {
            printf("%d $%.2f", 6, 52.50);
        }
    } else {
        if (weight < 3500) {
            printf("%d $%.2f", 7, 35.50);
        } else {
            printf("%d $%.2f", 8, 65.50);
        }
    }
    return 0;
}
```

LAB04: while, switch, EOF

Write a program that converts an input temperature from degrees Celsius to degrees Fahrenheit and vice versa. The program will terminate when an EOF sentinel is entered.

1. Use "38C" for Celsius or "75F" for Fahrenheit

2. EOF: CTRL-D (unix-style systems, online compilers) or CTRL-Z (Windows)

```c
#include <stdio.h>

int main() {
    double degree = 0, celsius = 0, fahrenheit = 0;
    char symbol;
    printf(
        "Please type in the degree and the unit (C/c/F/f) or use \"EOF\"
to "
        "terminate the program:\n");
    while (scanf("%lf%c", &degree, &symbol) != EOF) {
        if (symbol == 'c' || symbol == 'f') {
            symbol -= 32;
        }
        switch (symbol) {
            case 'C':
                fahrenheit = 9.0 / 5.0 * degree + 32.0;
                printf(
                    "The Fahrenheit equivalent of %5.2f degrees Celsius is
"

                    "%5.2f degrees\n\n",
                    degree, fahrenheit);
                break;
            case 'F':
                celsius = 5.0 / 9.0 * (degree - 32.0);
                printf(
                    "The Celsius equivalent of %5.2f degrees Fahrenheit is
"

                    "%5.2f degrees\n\n",
                    degree, celsius);
                break;
        }
        printf(
            "Please type in the degree and the unit or use \"EOF\" to "
            "terminate the program:\n");
    }
    return 0;
}
```

LAB05: Functions

The determinant of the 2-by-2 matrix

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

is $a_{11}a_{22} - a_{21}a_{12}$

Similarly, the determinant of a 3-by-3 matrix

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

is $a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$ - $a_{21}\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix}$ + $a_{31}\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}$

Using this information write and test two functions, named det2() and det3().The det2()function should accept the four coefficients of a 2-by-2 matrix and return its determinant. The det3() function should accept the nine coefficients of a 3-by-3 matrix and return its determinant by calling det2() to calculate the required 2-by-2 determinants.

```c
// 这个代码写得有点超前了，但是也没关系（直接用了二维数组和指针）
#include <stdio.h>
#include <stdlib.h>

double det2(double**, int, int, int, int);
double det3(double**);

int main() {
    int level = 0;
    double result = 0;
    printf("please type in the level of determinant(2 / 3)\n");
    scanf("%d", &level);
    if (level == 2) {
        double** det = (double**)malloc(sizeof(double*) * level);
        for (int i = 0; i < level; i++) {
            det[i] = (double*)malloc(sizeof(double) * level);
            for (int j = 0; j < 2; j++) {
                scanf("%lf", &det[i][j]);
            }
        }
        result = det2(det, 0, 1, 0, 1);
        printf("%lf\n", result);
    } else if (level == 3) {
        double** det = (double**)malloc(sizeof(double*) * level);
        for (int i = 0; i < level; i++) {
            det[i] = (double*)malloc(sizeof(double) * level);
            for (int j = 0; j < 3; j++) {
                scanf("%lf", &det[i][j]);
            }
        }
        result = det3(det);
        printf("%lf\n", result);
    } else {
        printf("Wrong input, the program will end.\n");
    }
    return 0;
}

double det2(double** det, int row1, int row2, int col1, int col2) {
    double result =
        det[row1][col1] * det[row2][col2] - det[row1][col2] * det[row2]
[col1];
    return result;
}

double det3(double** det) {
    double result = 0;
    result += det[0][0] * det2(det, 1, 2, 1, 2);
    result -= det[1][0] * det2(det, 0, 2, 1, 2);
    result += det[2][0] * det2(det, 0, 1, 1, 2);
    return result;
}
```

LAB06: Pointers

Write a function named date( ) that accepts an integer of the form yyyymmdd, such as 20220412; determines the corresponding month, day, and year; and returns these three values to the calling function. For example, if date is called using the statement

date(20220411, &month, &day, &year)

the number 4 should be returned in month, the number 11 in day, and the number 2022 in year.

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

void date(int input, int* month, int* day, int* year) {
    int m = 0, d = 0, y = 0;
    for (int i = 0; i < 8; i++) {
        if (i < 2) {
            d += (input % 10) * pow(10, i);
        } else if (i < 4) {
            m += (input % 10) * pow(10, i - 2);
        } else {
            y += (input % 10) * pow(10, i - 4);
        }
        input /= 10;
    }
    *month = m;
    *day = d;
    *year = y;
    return;
}

int main() {
    int input;
    int month, day, year;
```

```c
    printf("Please input a date (Format:yyyymmdd).\n");
    scanf("%d", &input);
    date(input, &month, &day, &year);
    printf("%d %d %d\n", year, month, day);
    return 0;
}
```

LAB07: 2-D Arrays

Write a function void mincol(int a[N][N], int b[N]) to find the smallest element in each column of the array a[][] of size NxN, and store it in array b[] of size N.

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

int n;
void mincol(int** a, int* b) {
    for (int i = 0; i < n; i++) {
        b[i] = a[0][i];
        for (int j = 1; j < n; j++) {
            if (a[j][i] < b[i]) {
                b[i] = a[j][i];
            }
        }
    }
    return;
}

int main() {
    printf("Input a number n to create a 2-D n x n array.\n");
    scanf("%d", &n);
    int** a = (int**)malloc(sizeof(int*) * n);
    int* b = (int*)calloc(n, sizeof(int));
    printf("Input the elements of this n x n array.\n");
    for (int i = 0; i < n; i++) {
        a[i] = (int*)calloc(n, sizeof(int));
        for (int j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    mincol(a, b);
    for (int i = 0; i < n; i++) {
        printf("The minimum number of column %d is %d.\n", i, b[i]);
    }
    return 0;
}
```

LAB08: Strings

Write a C program to sort a string array in ascending order.

Test Data: "resource"

Expected Output: "ceeorrsu"

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

#define MAXLEN 50

int main() {
    char str[MAXLEN];
    scanf("%s", str);
    for (int i = 0; i < strlen(str) - 1; i++) {
        for (int j = 0; j < strlen(str) - 1 - i; j++) {
            if (str[j] > str[j + 1]) {
                char temp = str[j];
                str[j] = str[j + 1];
                str[j + 1] = temp;
            }
        }
    }
    printf("The string in ascending order is \"%s\".\n", str);
    return 0;
}
```

LAB09: Random File Access(fseek(), getc())

Write a C program that will read and display every second character in a file named test.dat.

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

int main() {
    FILE* fp;
    if ((fp = fopen("test.dat", "r")) == NULL) {
        printf("The file is not open correctly.\n");
        printf("Please check that the file exists\n");
        exit(1);
    } else {
        int i = 1;
        printf("Every second character in a file shows as follow:\n");
        while (1) {
            fseek(fp, i, SEEK_SET);
            char character = getc(fp);
            if (character == EOF) {
                break;
            } else {
                putc(character, stdout);
            }
            i += 2;
        }
        printf("\n");
    }
    fclose(fp);
    return 0;
}
```

LAB10: Array of Pointers

June 1, 2021, is Tuesday. The user enters a date (1-30) in June, and your program will output the corresponding day of the week. Please use an array of pointers to store the names of the days of the week.

```
#include <stdio.h>

int main() {
    const char* dayofweek[7] = {"Monday", "Tuesday",  "Wednesday",
"Thursday",
                                "Friday", "Saturday", "Sunday"};
    int date = 0;
    scanf("%2d", &date);
    if (date < 1 || date > 30) {
        printf("The date is out of correct range.\n");
    } else {
        int cnt = date % 7;
        printf("This corresponding day of 2021/06/%d is %s.\n", date,
               *(dayofweek + cnt));
    }
    return 0;
}
```

LAB11: Unions

Each student's information includes name, student ID, gender(male:'M', female:'F'). In addition, for boys, you need to record their eyesight (normal: 'Y', abnormal: 'N'); for girls, you need to record their height and weight. Complete the following program that reads the information of two students from keyboard into a structure array stud[2](a boy and a girl), and displays it on the screen.

```
struct Student{
    char name[20];
    int studId;
    char gender;
    union Body {
        char eye;
        struct {
            int height;
            int weight;
        }f;
    }body;
}stud[2];

int main() {
    ...
    return 0;
}
```

```
#include <stdio.h>

struct Student {
    char name[20];
    long long studId;
    char gender;
    union Body {
        char eye;
        struct {
            int height;
            int weight;
        } f;
    } body;
} stud[2];

int main() {

    printf("Input format: Name Student ID Gender(M / F).\n");
    printf(
        "Then, if Gender is M(Male), you need to type in his eyesight(Y /
"
        "N),\n");
    printf("else you need to type in her height and weight.\n");

    for (int i = 0; i < 2; i++) {
        scanf("%s%10lld", stud[i].name, &stud[i].studId);

        do {
            stud[i].gender = getchar();
        } while (stud[i].gender != 'M' && stud[i].gender != 'F');

        if (stud[i].gender == 'M') {

            do {
                stud[i].body.eye = getchar();
            } while (stud[i].body.eye != 'Y' && stud[i].body.eye != 'N');

        } else {

            scanf("%d%d", &stud[i].body.f.height, &stud[i].body.f.weight);

        }
    }
    printf("\nThe students information is as follow:\n");
    for (int i = 0; i < 2; i++) {

        printf("Name:%22s\n", stud[i].name);
        printf("Student ID: %5c%010lld\n", ' ', stud[i].studId);

        if (stud[i].gender == 'M') {
            printf("Gender:%20s\n", "Male");

            if (stud[i].body.eye == 'Y') {
                printf("Eyesight: %17s\n", "Normal");
            } else {
                printf("Eyesight: %17s\n", "Abnormal");
            }

        } else {

            printf("Gender:%20s\n", "Female");
            printf("Height:%18dcm\n", stud[i].body.f.height);
            printf("Weight:%18dkg\n", stud[i].body.f.weight);

        }

        printf("\n");
    }
    return 0;
}
```

LAB12: Linked List

1: Watch the following video: Print elements of a linked list in forward and reverse order using recursion

2: Add two functions printForward(), printReverse() to Program 13.7 that print elements of the linked list in forward, and reverse order using recursion.

Example program code: Csource.rar

```
#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 readInsert(); /* function prototypes */
    void printForward(struct NameRec*);
    void printReverse(struct NameRec*);

    firstRec = NULL; /* initialize list pointer */
    readInsert();
    printf("\nThe names currently in the list, in alphabetical");
    printf("\norder, are:\n");
    printForward(firstRec);
    printf("\nThe above names displayed in reverse order, are:\n");
    printReverse(firstRec);
    printf("\n");

    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(Forward) */
void printForward(struct NameRec* node) {
    if (node == NULL) {
        return;
    }
    printf("%s\n", node->name);
    printForward(node->nextAddr);
}

/* display names from the linked list(Reverse) */
void printReverse(struct NameRec* node) {
    if (node == NULL) {
        return;
    }
    printReverse(node->nextAddr);
    printf("%s\n", node->name);
}
```

LAB13: Bit-masking

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**.

Based on the above functions, Write a C program to toggle all characters in a string, for example,

Input: tu@kmiNi123

Output: TU@KMInI123

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

char str[MAXLEN];

void upperToLower(char* letter) {
    *letter = (*letter) | ' ';
    return;
}

void lowerToUpper(char* letter) {
    *letter = (*letter) & '_';
    return;
}

int main(void) {
    printf("Please enter a string:\n");
    scanf("%s", 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]);
        }
    }
    printf("The result of uppercase and lowercase conversion is:\n");
    printf("%s\n", str);
    return 0;
}
```

LAB14: C++ function overloading

1）Using function overloading, write a function area( ) to calculate the area of a circle, rectangular, and a trapezoid with one, two and three arguments.

2）Write another function add( ) to add two ints or two doubles or two strings.

```
#include <iostream>
#include <string>
using namespace std;
#define PI 3.141592

double area(double radius) { return radius * radius * PI; }

double area(double length, double width) { return length * width; }

double area(double upperbase, double bottom, double height) {
    return height * (upperbase + bottom) / 2;
}

template <typename Type_Add>
Type_Add add(Type_Add a, Type_Add b) {
    return a + b;
}

int main(void) {
    ios::sync_with_stdio(false);

    // Calculation of circle area
    double r;
    cout << "Please enter the radius of a circle:" << endl;
    cin >> r;
    cout << "The area of a circle with radius " << r;
    cout << " is " << area(r) << "."
         << "\n"
         << endl;

    // Calculation of rectangular area
    double l, w;
    cout << "Please enter the length and width of a rectangle:" << endl;
    cin >> l >> w;
    cout << "The area of a rectangle with length " << l;
    cout << " and width " << w;
    cout << " is " << area(l, w) << "."
         << "\n"
         << endl;

    // Calculation of a trapezoid area
    double u, b, h;
    cout << "Please enter the upperbase, bottom and height of a
trapezoid:"
         << endl;
    cin >> u >> b >> h;
    cout << "The area of a trapezoid with upper base " << u;
    cout << " bottom " << b << " and height " << h;
    cout << " is " << area(u, b, h) << "."
         << "\n"
         << endl;

    // Addition of 2 int-type numbers
    int num1, num2;
    cout << "Please enter 2 int-type numbers:" << endl;
    cin >> num1 >> num2;
    cout << num1 << " + " << num2 << " = " << add(num1, num2) << "\n" <<
endl;

    // Addition of 2 double-type numbers
    double num3, num4;
    cout << "Please enter 2 double-type numbers:" << endl;
    cin >> num3 >> num4;
    cout << num3 << " + " << num4 << " = " << add(num3, num4) << "\n" <<
endl;

    // Addition of 2 strings
    string str1, str2;
    cout << "Please enter 2 string(Seperated by SPACE or ENTER):" << endl;
    cin >> str1 >> str2;
    cout << str1 << " + " << str2 << " = " << add(str1, str2) << "\n" <<
endl;

    return 0;
}
```