Универзитет „Св. Кирил и Методиј“ - Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Структурно програмирање

Пример задачи од 2 колоквиум

Верзија 1.0, 19 Декември, 2016

Содржина

1. Задачи

1.1. Задача 1

За еден природен број `a` велиме дека е порамнување на друг природен број `b` ако и само ако цифрите еднакви на 9 во бројот `b` се заменети со цифрата 7 во бројот `a`. Пример. Бројот 734775 е порамнување на бројот 934795.

Од стандарден влез се внесуваат непознат број на цели броеви (не повеќе од 100), се додека не се внесе нешто што не може да се интерпретира како цел број. Ваша задача е да ги отпечатите најмалите 5 од порамнувањата на сите внесени броеви, по редослед од најмалиот кон најголемиот.

Доколку се внесат помалку од 5 броеви, тогаш печатите толку броеви колку што се соодветно внесени. Наоѓањето на порамнувањето на даден број треба да се реализира во посебна рекурзивна функција `poramnet(int a)`.

Пример.

За броевите: 9592, 69403, 100007, 6, 987, 6977, 33439, треба да се најдат нивните порамнувања (тоа се: 7572, 67403, 100007, 6, 787, 6777 и 33437, соодветно), и да се отпечатат најмалите 5 од нив по овој редослед: 6 787 6777 7572 33437.

ЗАБРАНЕТО е користење на глобални променливи.

Решение `p_kol2_1.c`

```c
#include<stdio.h>

int poramni(int a){
    if (a<=0)
        return 0;
    int retval = a % 10;
    if (retval == 9)
        retval = 7;
    return poramni(a/10) *10 + retval;
}
int max_index(int *a, int n){
    int max_i = 0;
    int i;
    for (i = 0; i < n; ++i){
        if (a[i]>a[max_i]){
            max_i = i;
        }
    }
    return max_i;
}
void sort_asc(int *a, int n){
    int i, j, tmp;
    for(i = 0; i < n - 1; i++){
        for(j = 0; j < n - i - 1; j++){
            if(a[j] > a[j+1]){
                tmp = a[j];
                a[j] = a[j+1];
                a[j+1] = tmp;
            }
        }
    }
}
void print_arr(int *a, int n){
    int i;
    for (i = 0; i < n; ++i){
        printf("%d ", a[i]);
    }
}
int main() {
    int x, k, n, max_i;
    int a[5];
    n=0;
    while (scanf("%d", &x)){
        k = poramni(x);
        if (n>=5){
            max_i = max_index(a, 5);
            if (a[max_i] > k){
                a[max_i] = k;
            }
        }
        else {
            a[n] = k;
            n++;
        }
    }
    sort_asc(a,n);
    print_arr(a, n);
    return 0;
}
```

1.2. Задача 2

За квадратна матрица A со димензии n x n, од стандарден влез се внесува бројот n (n >=2) и елементите на матрицата (реални броеви). Нека X е збирот од елементите под главната дијагонала во матрицата A. Нека Y е збирот од

елементите под споредната дијагонала во матрицата A. Да се креира нова матрица B на следниот начин:

- сите елементи од главната дијагонала во матрицата B треба да имаат вредност X
- сите елементи од споредната дијагонала во матрицата B треба да имаат вредност Y
- ако даден елемент припаѓа и на главната и на споредната дијагонала во матрицата B, тогаш неговата вредност е X+Y
- сите останати елементи во матрицата B имаат вредност 0

Новата матрица B да се испечати на стандарден излез.

Пример:

```
Матрица A
5         5.5          6          1.2          2.5
8         95.1         21.3       13           0.2
34        4.1          37.4       22           6
4.1       5.5          0.7        7            0
42        1.1          3.2        7.5          1.8
Матрица B
110.2          0          0          0          49.5
0              110.2      0          49.5       0
0              0          159.7      0          0
0              49.5       0          110.2      0
49.5           0          0          0          110.2
```

Решение `p_kol2_2.c`

```c
#include<stdio.h>

int main(){
    int i, j, n;
    float  broj, a[100][100], x = 0.0, y = 0.0;
    scanf("%d", &n);
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%f", &broj);
            if(i > j)
                x += broj;
            if(i + j >= n)
                y += broj;
        }
    }
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            if(i == j)
                a[i][j] = x;
            else if(i + j == n - 1)
                a[i][j] = y;
            else
                a[i][j] = 0;
        }

    }
    if(n % 2)
        a[n / 2][n / 2] = x + y;
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            printf("%.1f ", a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

1.3. Задача 3

Во дадена датотека `dat.txt` да се најде најдолгиот ред во кој има барем 2 цифри. На стандарден излез да се испечатат знаците од најдолгиот ред кои се наоѓаат помеѓу првата и последната цифра (заедно со тие 2 цифри) во истиот редослед. Доколку има повеќе такви редови се печати последниот. Се претпоставува дека ниту еден ред на датотеката не е подолг од 100 знаци.

Пример

```
Влез:
dat.txt:
aaa123aa222aa2aaa23aaaaa22
aaaaaaaaaaaa 23aaaa
123 aaa aaa aaa aaa 12345 aaa aaa 2a

Излез:
123 aaa aaa aaa aaa 12345 aaa aaa 2
```

Решение `p_kol2_3.c`

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

int main() {
    FILE *f = fopen("dat.txt", "r");
    int br = 0, l, flag = 0, start = 0, end = 0, m_start = 0, m_end = 0;
    int i = 0;
    char c;
    char word[100];
    char linija[100], max_linija[100];
    int max = 0;

    while(fgets(linija,100,f)!=NULL) {
        l = strlen(linija);
        flag=0;
            for(i=0;i<l;i++){
                if(isdigit(linija[i])){
                    if(!flag){
                        start=i;
                        flag = 1;
                    }
                    end=i;
                }
            }
        if(start!=end){
            if(l >= max){
                max = l;
                strcpy(max_linija,linija);
                m_start = start;
                m_end = end;
            }
        }
    }
    strncpy(word, max_linija+m_start, m_end-m_start+1);
    word[m_end-m_start+1]='\0';
    printf("%s\n",word);

    fclose(f);
    return 0;
}
```

1.4. Задача 4

Од стаднарден влез се читаат N низи од знаци (стрингови) не подолги од 80 знаци. На почетокот на програмата се читаат два цели броеви: N - бројот на низи од знаци кои ќе се читаат и X - поместување. Секоја од вчитаните низи од знаци треба да се трансформира на тој начин што секоја од малите и големите букви (a-z, A-Z) се заменува со истата буква поместена X места понапред во азбуката (az). Ако се надмине опсегот на буквите во азбуката, се продолжува циклично од почетокот на азбуката. Трансформираната низа да се отпечати на СИ. Трансформацијата да се имплементира со посебна рекурзивна функција.

Пример:

`Welcome` → трансформирано со поместување 5 → `Bjqhtrj`

Решение `p_kol2_4.c`

```c
#include <stdio.h>

void cipher(char *word, int x) {
    if(*word == 0) return;
    if(isalpha(*word)) {
        char first = 'a';
        if(isupper(*word))
            first = 'A';
        *word = first + (*word + x -first) % 26;
    }
    cipher(++word, x);
}

int main() {
    int n, x;
    scanf("%d %d\n", &n, &x);
    int i;
    char word[80];
    for(i = 0; i < n; ++i) {
        gets(word);
        cipher(word, x);
        printf("%s\n", word);
    }
    return 0;
}
```

1.5. Задача 5

Да се напише програма која вчитува матрица со димензии MxN (макс. 100x100). На почетокот се внесуваат димензиите на матрицата, а потоа и елементите на матрицата кои се само вредностите 1 и 0. Програмата треба да изброи и отпечати на СИ во колку од редиците и колоните има барем 3 последователни елементи со вредност 1.

Пример

```
1    1     1    0
1    0     1    1
1    0     0    1
1 ред + 1 колона = 2
```

Решение `p_kol2_5.c`

```c
#include <stdio.h>

int main() {
    int a[100][100];
    int rows, cols;
    int i, j;
    scanf("%d %d", &rows, &cols);
    for (i = 0; i < rows; ++i) {
        for (j = 0; j < cols; ++j) {
            scanf("%d", &a[i][j]);
        }
    }
    int count=0;
    int flag;
    int c;
    for (i = 0; i < rows; ++i) {
        c = 0;
        flag = 0;
        for (j = 0; j < cols; ++j) {
            if (a[i][j] == 1) {
                ++c;
            } else {
                if (c >= 3) {
                    flag = 1;
                    ++count;
                    break;
                }
                c = 0;
            }
        }
        if (!flag&&c >= 3) {
            ++count;
        }
    }
    for (i = 0; i < cols; ++i) {
        c = 0;
        flag = 0;
        for (j = 0; j < rows; ++j) {
            if (a[j][i] == 1) {
                ++c;
            } else {
                if (c >= 3) {
                    flag = 1;
                    ++count;
                    break;
                }
                c = 0;
            }
        }
        if (!flag&&c >= 3) {
            ++count;
        }
    }
    printf("%d\n", count);
    return 0;
}
```

1.6. Задача 6

Во дадена датотека `broevi.txt` се запишани повеќе редови со броеви така што секој ред започнува со еден цел број (N>=1) што означува колку броеви следуваат по него во тој ред. Да се напише програма која на СИ за секој ред ќе го испечати бројот со најголема најзначајна цифра. Читањето на броеви завршува

кога ќе се прочита 0.

Решение `p_kol2_6.c`

```c
#include <stdio.h>
#include <string.h>
#define MAX 100

int digit(int n) {
    while(n >= 10) {
        n /= 10;
    }
    return n;
}

int main(){
   FILE* f = fopen("broevi.txt", "r");
    int i;
    while(!feof(f)) {
        int n, x;
        fscanf(f, "%d", &n);
        if(n == 0) break;
        int max = 0;
        int maxN = 0;
        for(i = 0; i < n; ++i) {
            fscanf(f, "%d", &x);
            int y = digit(x);
            if(y > max) {
                max = y;
                maxN = x;
            }
        }
        printf("%d\n", maxN);
    }

    fclose(f);
    return 0;

}
```

2. Изворен код од примери и задачи

https://github.com/finki-mk/SP/

Source code ZIP