



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

Структурно програмирање

Пример задачи од 1 колоквиум

Содржина

1. Задачи	1
1.1. Задача 1	1
1.2. Задача 2	2
1.3. Задача 3	3
1.4. Задача 4	4
1.5. Задача 5	5
1.6. Задача 6	6
2. Изворен код од примери и задачи	8

1. Задачи

1.1. Задача 1

Од тастатура се внесува цел број n , а потоа и n тројки природни броеви (должини на три отсечки). За секоја тројка должини треба да се испечати порака Може ако со нив може да се конструира триаголник, во спротивно се печати Не може. На крај да се испечати редниот број на тројката броеви со која се конструира триаголникот со најголем периметар (се земаат предвид само тројките со кои може да се конструира триаголник), како и вредноста на најголемиот периметар. Ако постојат повеќе такви тројки броеви, да се испечати редниот број на последната. Се смета дека првата внесена тројка е со реден број еден, втората со реден број два итн.

Пример:

```
Влез:      Излез:
5
1 2 3      Ne moze
3 4 5      Moze
5 12 13    Moze
50 53 1    Ne moze
16 12 20    Moze
Najgolem perimetar: 48, reden broj 5
```

Решение p_kol1_1.c

```
#include <stdio.h>
int main () {
    int i, n, maxL = 0, maxIndeks = 0;
    int a, b, c, l;
    scanf ("%d", &n);
    for (i = 0; i < n; i++) {
        scanf ("%d %d %d", &a, &b, &c);
        if (a + b > c && a + c > b && b + c > a) {
            printf("Moze\n");
            l = a + b + c;
            if (l >= maxL) {
                maxL = l;
                maxIndeks = i;
            }
        }
        else
            printf("Ne moze\n");
    }
    printf ("Najgolem perimetar: %d, reden broj %d\n", maxL, maxIndeks + 1);
    return 0;
}
```

1.2. Задача 2

Да се напише програма во која од тастатура се внесуваат непознат број мали букви сè додека не се внесе знак што не е мала буква. За секој пар последователни букви, во нов ред треба да се испечати секвенца од буквите во абецеден редослед што се наоѓаат меѓу нив (без нив) проследено со вкупниот број на вакви букви. Секогаш печатењето се почнува од буквата што се наоѓа понапред во азбуката кон таа што е поназад. Ако парот букви е составен од две соседни букви во азбуката, не се печати ништо. На крај да се испечати вкупниот број на испечатени секвенци од букви.

Пример:

```
Пример:
Влез:
bcjfmбX

Излез:
defghi 6
ghi 3
ghijkl 6
cdefghijkl 10
4
```

Решение p_kol1_2.c

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    char c, prev, startL, endL;
    int first = 1, n = 1, m = 0;
    while (1) {
        scanf ("%c", &c);
        if (c < 'a' || c > 'z')
            break;
        if (first) {
            first = 0;
        } else {
            if (abs(prev - c) != 1) {
                if (prev < c) {
                    startL = prev + 1;
                    endL = c - 1;
                }
                else {
                    startL = c + 1;
                    endL = prev - 1;
                }
                for (n = 0; startL <= endL; startL++, n++)
                    printf ("%c", startL);
                printf (" %d\n", n);
                m++;
            }
        }
        prev = c;
    }
    printf ("Vkupno: %d\n", m );
    return 0;
}
```

1.3. Задача 3

Empires & Allies е стратегиска игра во повеќе нивоа со можност за играње од страна на повеќе играчи. Секој играч има своја територија, како и одбранбени и напаѓачки единици. Може да влезе во битка (напад на туѓа територија или одбрана на сопствената) со други играчи што може да заврши со победа или пораз за што се добиваат поени. Да се напише програма во која од тастатура се внесува бројот на играчи на играта Empires & Allies. За секој играч потоа се внесуваат податоци за ниво (цел број), број на победи и број на порази.

Резултатот на играчот се пресметува на следниот начин:

- ако се работи за прво ниво, секоја победа носи 13 поени, а секој пораз носи -1 поен
- ако се работи за второ ниво, секоја победа носи 13 поени, а секој пораз носи -3 поени

За секој играч треба да се испечати Dobar igrac ако неговиот број на поени е позитивен или Los igrac во сите други случаи. Потоа треба да се најде и испечати најдобриот резултат (најголемиот вкупен број на поени), како и редниот број на играчот со најдобар резултат (првиот играч е со реден број еден, вториот со реден број два итн.). Ако постојат повеќе играчи со ист најдобар резултат, да се испечати редниот број на првиот таков играч.

Пример

```
Пример:
Влез:      Излез:
4
1 3 7      Dobar igrac
2 1 8      Los igrac
2 6 4      Dobar igrac
1 5 5      Dobar igrac
Najdobar Igrac: br. 3, 66 poeni
```

Решение p_kol1_3.c

```

#include <stdio.h>
int main () {
    int i, n, maxPoeni, maxIndeks, f = 1;
    int pobedi, porazi, poeni, nivo;
    scanf ("%d", &n);
    for (i = 0; i < n; i++) {
        scanf ("%d %d %d", &nivo, &pobedi, &porazi);
        if (nivo == 1)
            poeni = pobedi * 13 - porazi;
        if (nivo == 2)
            poeni = pobedi * 13 - porazi * 3;
        if (poeni > 0) printf("Dobar igrac\n");
        else printf("Los igrac\n");
        if (f) {
            f = 0;
            maxPoeni = poeni;
            maxIndeks = i;
        }
        else if (poeni > maxPoeni) {
            maxPoeni = poeni ;
            maxIndeks = i;
        }
    }
    printf ("Najdobar igrac: br. %d, %d poeni\n", maxIndeks + 1, maxPoeni);
    return 0;
}

```

1.4. Задача 4

Од тастатура се внесува цел број m , а потоа непознат број цели броеви. Да се испечатат должините на секвенците составени од најмалку два последователни броја за кои истовремено важи:

- следниот број е строго поголем од претходниот и
- секој од нив има точно m цифри.

Задачата да се реши без користење на низи.

Пример

Пример:

Влез:

3 45 456 567 784 67890 12 543 321 462 2 23 34 567 765 898 975

Излез:

Dolzini: 3 2 4

```

#include <stdio.h>
int main () {
    int c, prev;
    int first = 1, tmp, noDigits, prevNoDigits;
    int n = 1, m;
    scanf("%d", &m);
    while (scanf ("%d", &c)) {
        tmp = c;
        noDigits = 0;
        while (tmp) {
            noDigits++;
            tmp /= 10;
        }
        if (first) {
            first = 0;
        } else {
            if (c > prev && noDigits == m && prevNoDigits == m) {
                n++;
            } else {
                if (n >= 2 ) {
                    printf("%d ", n);
                }
                n = 1;
            }
        }
        prev = c;
        prevNoDigits = noDigits;
    }
    if (n >= 2 ) {
        printf("%d", n);
    }
    return 0;
}

```

1.5. Задача 5

Од стандарден влез се чита еден број кој претставува датум во формат DDMMYYYY (DD-ден, MM-месец, YYYY-година) кој го означува денешниот датум. Потоа се вчитува цел број N, по кој се вчитуваат N датуми на раѓање во дадениот формат. За секој од прочитаните N датуми на раѓање треба да се отпечати "DA" ако на денешниот ден (вчитан на почетокот) има повеќе или точно 18 години, а во спортивно "NE".

```

#include <stdio.h>
int main() {
    int date;
    scanf("%d", &date);
    int n;
    scanf("%d", &n);
    int i;
    int day = date / 100000;
    int month = (date / 10000) % 100;
    int year = date % 10000;
    for (i = 0; i < n; ++i) {
        int d;
        scanf("%d", &d);
        int dd = d / 100000;
        int mm = (d / 10000) % 100;
        int yy = d % 10000;
        int years = year - yy;
        if (years < 18) {
            printf("NE\n");
        } else if (years == 18) {
            if (mm < month) {
                printf("DA\n");
            } else if (mm == month) {
                if (dd <= day) {
                    printf("DA\n");
                } else {
                    printf("NE\n");
                }
            } else {
                printf("NE\n");
            }
        } else if (years > 18) {
            printf("DA\n");
        }
    }
    return 0;
}

```

1.6. Задача 6

Со помош на знаците +, , и % се формира квадратна слика со „должина“ m ($m > 2$) која претставува квадрат од знаците %, впишан во квадрат од знаците + и (види примери). За внесено m да се испечати на екран соодветната слика. Внимавајте – не печатете нити еден дополнителен знак, како празно место, непотребен знак за нов ред... Сликата завршува со знак за нов ред.



Бидејќи е дадено ограничување за m , во точно решение на задачата не треба да се користи низа/матрица.

Пример:

$m = 5$


```
+++++
+%%%+
+%%%+
+%%%+
+++++
```

Решение p_kol1_6.c

```
#include <stdio.h>

int main() {
    int m, n, i, j;
    scanf("%d", &m);
    n = m;
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++)
            if (i > 0 && i < m - 1 && j > 0 && j < n - 1)
                printf("%%");
            else if (j > 0 && j < n - 1)
                printf("*");
            else
                printf("+");
        printf("\n");
    }
    return 0;
}
```

2. Изворен код од примери и задачи

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

Source code ZIP