



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

Структурно програмирање

Аудиториски вежби 5

Содржина

1. Контролни структури за циклуси	1
1.1. Задача 1а	1
1.2. Задача 1б	1
1.3. Задача 2	2
1.4. Задача 3	3
1.5. Задача 4	5
1.6. Задача 5	5
1.7. Задача 6	6
1.8. Задача 7	6
1.9. Задача 8	6
1.10. Задача 9	7
1.11. Задача 10	8
2. Контролна структура за повеќекратно гранење (switch)	9
2.1. Задача 1	9
2.2. Задача 2	10
2.3. Задача 3	10
2.4. Пример за реализација на (текст) мени	11
3. Задачи за вежбање	12
3.1. Задача за вежбање 1	12
3.2. Задача за вежбање 2	13
3.3. Задача за вежбање 3	14
4. Изворен код од примери и задачи	16

1. Контролни структури за циклуси

1.1. Задача 1а

Да се напише програма за пресметување на сумата на сите парни двоцифрени броеви. Добиената сума се печати на екран.

Решение:

```
#include <stdio.h>
int main () {
    int i = 10, sum = 0;
    while (i <= 98) {
        sum = sum + i;
        i+=2;
    }
    printf("%d\n", sum);
    return 0;
}
```

1.2. Задача 1б

Да се напише програма за пресметување на сумата на сите непарни двоцифрени броеви. Програмата ја печати сумата на екран во следниот формат:

```
11 + 13 + 15 + 17 + ... + 97 + 99 = 2475
```



Програмата да се реши без користење на наредбата `i f`

Решение 1:

```
#include <stdio.h>
int main () {
    int i = 11, sum = 0;
    printf("%d", i);
    sum = i;
    i=i+2;
    while (i <= 99){
        printf(" + %d", i);
        sum = sum + i;
        i+=2;
    }
    printf(" = %d\n", sum);
    return 0;
}
```

Решение 2:

```

#include <stdio.h>
int main () {
    int i = 11, sum = 0;
    while (i <= 97) {
        printf("%d + ", i);
        sum = sum + i;
        i+=2;
    }
    printf(" %d", i);
    sum = sum + i;
    printf(" = %d\n", sum);
    return 0;
}

```

1.3. Задача 2

Да се напише програма за пресметување на $y = x^n$ за даден природен број n , $n \geq 1$ и реален број x .

Решение 1:

```

#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    while (counter < n) {
        y *= x;
        counter++;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}

```

Решение 2:

```

#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    do {
        y *= x;
        counter++;
    } while (counter < n);
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}

```

Решение 3:

```
#include <stdio.h>
int main () {
    int counter, n;
    float x, y;
    printf("x: ");
    scanf("%f", &x);
    printf("n: ");
    scanf("%d", &n);
    for(counter = 1, y = x; counter < n; counter++) {
        y *= x;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

Решение 4:

```
#include <stdio.h>
int main () {
    int counter = 0, n;
    float x, y = 1;
    printf("x: "); scanf("%f", &x);
    printf("n: "); scanf("%d", &n);
    for( ; counter < n; counter++) {
        y *= x;
    }
    printf("%f^%d = %f\n", x, n, y);
    return 0;
}
```

1.4. Задача 3

Да се напише програма што од n броеви (внесени од тастатура) ќе го определи бројот на броеви што се деливи со 3, при делењето со 3 имаат остаток 1, односно 2.



Задачата да се реши со `while`, `do...while` и `for`

Решение со while:

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /* brojaci */
    scanf("%d", &n); /* koklu broevi kje bidat vneseni */
    while (i < n) {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
        i++;
    }
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

Решение со do... while:

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /* brojaci */
    scanf("%d", &n); /* koklu broevi kje bidat vneseni */
    do {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
        i++;
    } while (i < n);
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

Решение со for:

```
#include <stdio.h>
int main () {
    int n = 1, i = 0, number, div, r1, r2;
    div = r1 = r2 = 0; /* brojaci */
    scanf("%d", &n); /* koklu broevi kje bidat vneseni */
    for (i = 0; i < n; ++i) {
        scanf("%d", &number);
        if (number % 3 == 0)
            div++;
        else if (number % 3 == 1)
            r1++;
        else r2++;
    }
    printf("%d\n", div);
    printf("%d\n", r1);
    printf("%d\n", r2);
    return 0;
}
```

1.5. Задача 4

Да се напише програма што на екран ќе ги испечати сите четири-цифрени броеви кај кои збирот на трите најмалку значајни цифри е еднаков со најзначајната цифра.



4031 (4=0+3+1), 5131 (5=1+3+1)

Решение:

```
#include <stdio.h>
int main() {
    int i, n, sum, first_digit, digit;
    i = 1000;
    while (i <= 9999) {
        first_digit = i / 1000;
        n = i % 1000;
        sum = 0;
        while (n > 0) {
            digit = n % 10;
            sum += digit;
            n /= 10;
        }
        if (sum == first_digit) printf("%d\t", i);
        i++;
    }
    return 0;
}
```

1.6. Задача 5

Да се напише програма што ќе ги испечати сите броеви од зададен опсег кои се читаат исто и одлево надесно и оддесно налево.

пример броеви

12321 5061605

Решение:

```
#include <stdio.h>
int main () {
    int i, from, to, temp, op, digit;
    scanf("%d %d", &from, &to);
    for (i = from; i <= to; i++) {
        temp = i;
        op = 0;
        while (temp > 0) {
            digit = temp % 10;
            op = op * 10 + digit;
            temp /= 10;
        }
        if (op == i) printf("%d\t", i);
    }
    return 0;
}
```

1.7. Задача 6

Да се напише програма што од непознат број на цели броеви што се внесуваат од тастатура ќе го определи бројот со максимална вредност. Програмата завршува ако се внесе невалидна репрезентација на број.

Решение:

```
#include <stdio.h>
int main() {
    int n, max;
    if (scanf("%d", &max)) {
        while (scanf("%d", &n)) {
            if (max < n) {
                max = n;
            }
        }
        printf("%d", max);
    } else {
        printf("Ne e vnesen broj.");
    }
    return 0;
}
```

1.8. Задача 7

Да се напише програма што од непознат број на цели броеви што се внесуваат од тастатура ќе го определи бројот со максимална вредност. Притоа, броевите поголеми од 100 не се земаат предвид т.е. се игнорираат. Програмата завршува ако се внесе невалидна репрезентација на број.

Решение:

```
#include <stdio.h>
int main() {
    int n, max;
    if (scanf("%d", &max)) {
        while (scanf("%d", &n)) {
            if (n > 100) continue;
            if (max < n) {
                max = n;
            }
        }
        printf("%d", max);
    } else {
        printf("Ne e vnesen broj.");
    }
    return 0;
}
```

1.9. Задача 8

Да се напише програма што од непознат број на цели броеви што се внесуваат

Структурно програмирање

од тастатура ќе ги определи двата броја со најголеми вредности. Програмата завршува ако се внесе невалидна репрезентација на број.

пример

Ако се внесат броевите 2 4 7 4 2 1 8 6 9 7 10 3 програмата ќе отпечати 10 и 9.

Решение:

```
#include <stdio.h>
int main() {
    int n, max1, max2, temp;
    if (scanf("%d%d", &max1, &max2) == 2) {
        if (max2 > max1) {
            temp = max1;
            max1 = max2;
            max2 = temp;
        }
        while (scanf("%d", &n)) {
            if (n > max1) {
                max2 = max1;
                max1 = n;
            } else if (n > max2) {
                max2 = n;
            }
        }
        printf("%d\n", max1);
        printf("%d\n", max2);
    } else {
        printf("Vnesete najmalku 2 broja");
    }
    return 0;
}
```

1.10. Задача 9

Да се напише програма што од N цели броеви внесени од тастатура ќе ја определи разликата од сумите на броевите на парни и непарни позиции (според редоследот на внесување). Ако оваа разлика е помала од 10 на екран се печати "Dvete sumi se slicni", а во спротивно "Dvete sumi mnogu se razlikuvaat".

пример

За броевите внесени од тастатура:

2 4 3 4 2 1 1 6 1 7

suma_neparni_pozicii = 9

suma_parni_pozicii = 22

На екран ќе се испечати:

Dvete sumi mnogu se razlikuvaat

Решение:

```
#include <stdio.h>
int main() {
    int difference, i, n = 0, number = 0;
    int sum_odd_positions = 0, sum_even_positions = 0;
    scanf("%d", &n);
    for (i = 1; i <= n; i++) {
        scanf("%d", &number);
        if (i % 2) {
            sum_odd_positions += number;
        } else {
            sum_even_positions += number;
        }
    }
    difference = sum_even_positions - sum_odd_positions;
    if (difference < 10 && difference > -10) { //if(abs(difference) < 10){
        printf("Dvete sumi se slicni");
    } else {
        printf("Dvete sumi mnogu se razlikuvaat");
    }
    return 0;
}
```

1.11. Задача 10

Да се напише програма што од непознат број на цели броеви кои се внесуваат од тастатура ќе ги определи позициите (редните броеви на внесување) на двата последователни броја што ја имаат најголемата сума. Програмата завршува ако едно по друго (последователно) се внесат два негативни цели броја.

Решение:

```
#include <stdio.h>
int main() {
    int pol_position, position, max_sum, sum, previous, next;
    scanf("%d%d", &previous, &next);
    pol_position = position = 2;
    max_sum = sum = previous + next;
    while (1) {
        if (previous < 0 && next < 0) {
            break;
        }
        sum = previous + next;
        if (sum > max_sum) {
            max_sum = sum;
            pol_position = position;
        }
        previous = next;
        scanf("%d", &next);
        position++;
    }
    if (position > 2)
        printf("Broevite se na pozicii %d i %d i nivnata suma e %d",
            pol_position - 1, pol_position, max_sum);
    return 0;
}
```

2. Контролна структура за повеќекратно гранење (switch)

Пример

```
#include <stdio.h>
int main() {
    char odgovor;
    printf("Dali SP e lesen predmet? (d/n): ");
    //odgovor=getchar();
    scanf("%c", &odgovor);

    switch (odgovor) {
        case 'D':
            printf("I jas mislam taka!\n");
            break;
        case 'N':
            printf("Navistina?\n");
            break;
        default:
            printf("Ova e DA ili NE?\n");
    }

    return 0;
}
```

2.1. Задача 1

Да се напише програма што за знак внесен од тастатура ќе одреди дали е самогласка, цифра или друго.

Решение:

```
#include <stdio.h>
int main()
{
    char znak;
    printf("Vnesi znak: ");
    znak=getchar(); /* scanf("%c", &znak); */

    switch (znak) {
        case 'a': case 'A':
        case 'e': case 'E':
        case 'i': case 'I':
        case 'o': case 'O':
        case 'u': case 'U':
            printf("Vnesena e samoglaskata: %c\n", znak);
            break;
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            printf("Vnesena e cifrata: %c\n", znak);
            break;
        default:
            printf("Vnesen e znakot: %c\n", znak);
    }

    return 0;
}
```

2.2. Задача 2

Да се напише програма што ќе претставува едноставен калкулатор. Во програмата се вчитуваат два броја и оператор во формат:

```
broj1 operator broj2
```

По извршената операција во зависност од операторот, се печати резултатот во формат:

```
broj1 operator broj2 = rezultat
```

Решение:

```
#include <stdio.h>
int main() {
    char op;
    float num1, num2, result = 0;
    printf("num1 operator num2\n");
    scanf("%f %c %f", &num1, &op, &num2);
    switch (op) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            if (num2 == 0) {
                printf("Delenje so 0\n");
            } else {
                result = num1 / num2;
            }
            break;
        default:
            printf("Nepoznat operator %c\n", op);
            break;
    }
    if (result) printf("%.2f %c %.2f = %f", num1, op, num2, result);
    return 0;
}
```

2.3. Задача 3

Да се напише програма што ќе го пресмета просекот на еден студент, како и бројот на испити на што студентот паднал. Бројот на испити кои студентот ги полагал не е познат.

Решение 1:

```

#include <stdio.h>
int main() {
    int ocena, suma = 0, brocenii = 0, padnal = 0;
    while (scanf("%d", &ocena) == 1) {
        while (ocena < 5 || ocena > 10) {
            printf("vnesi ocena od 5 do 10 : ");
            if (scanf("%d", &ocena) != 1) break;
        }
        /* presmetuvanje */
        if (ocena >= 6 && ocena <= 10) {
            suma += ocena;
            brocenii++;
        }
        else /*if (ocena == 5) */
            padnal++;
    }
    if (brocenii == 0)
        printf("Nema polozeni ispiti");
    else
        printf("Prosekot e %.2f\n", (float)suma / brocenii);
    if (padnal != 0)
        printf("Padnal na %d ispiti\n", padnal);
    else
        printf("Ne padnal na ispit\n");
    return 0;
}

```

Решение 2:

```

#include <stdio.h>
int main() {
    int ocena, suma = 0, brocenii = 0, padnal = 0;
    while (scanf("%d", &ocena) == 1) {
        switch (ocena) {
            case 6:
            case 7:
            case 8:
            case 9:
            case 10: suma += ocena;
                    brocenii++; break;
            case 5: padnal++; break;
            default: printf("vnesi ocena od 5 do 10 : "); break;
        }
    }

    if (brocenii == 0)
        printf("Nema polozeni ispiti");
    else
        printf("Prosekot e %.2f\n", (float)suma / brocenii);
    if (padnal != 0)
        printf("Padnal na %d ispiti\n", padnal);
    else
        printf("Ne padnal na ispit\n");
    return 0;
}

```

2.4. Пример за реализација на (текст) мени

Решение:

```

#include <stdio.h>
int main() {
    int vrednost, izbor;
    printf("Vnesi pocetna vrednost: ");
    scanf("%d", &vrednost);

    do {
        do {
            printf("Meni:\n");
            printf("1 - zgolemi\n");
            printf("2 - namali\n");
            printf("3 - dupliraj\n");
            printf("0 - kraj\n");
            printf("izbor : ");
            scanf("%d", &izbor);
        } while ((izbor < 0) || (izbor > 3));
        switch (izbor) {
            case 1: vrednost++; break;
            case 2: vrednost--; break;
            case 3: vrednost *= 2; break;
            case 0: printf("Kraj\n"); break;
            default: printf("Nevaliden izbor!\n"); break;
        }
        printf("Vrednost = %d\n", vrednost);
    }
    while (izbor != 0);
    return 0;
}

```

3. Задачи за вежбање

3.1. Задача за вежбање 1

Нека $x = \sin y$ се менува од 0.02 до 0.80 со чекор 0.01 . Да се напише програма што ќе го табелира y во форма степени, минути и секунди.

$$y = \arctan\left(\frac{x}{\sqrt{1-x^2}}\right)$$

Решение:

```

#include <stdio.h>
/* #define _USE_MATH_DEFINES */
#include <math.h>
int main() {
    float x, y;
    int step, min, sec;
    printf(" x\tstepeni\tminuti\tsekundi\n");
    for (x = 0.02; x <= 0.8; x += 0.01) {
        y = 180/M_PI * atan(x / sqrt(1 - x*x));
        step = y;
        min = (y - step) * 60;
        sec = ((y - step) * 60 - min) * 60 + 0.5;
        printf("%5.2f\t%3d\t %2d\t %2d\n", x, step, min, sec);
    }
    return 0;
}

```

3.2. Задача за вежбање 2

Да се напише програма за приближно пресметување на π според формулата за развој на *Gregory–Leibniz*

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n + 1} = \frac{\pi}{4}$$

- користејќи ги првите 100 члена.
- се додека апсолутната вредност на членот што се додава е поголема од 10^{-6}

Решение 1.1:

```
#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    double clen = 1, pi = 0;
    int i, znak = 1, imenitel = 1;
    for(i=0; i<100; i++) {
        pi += clen;
        imenitel += 2;
        znak = -znak;
        clen = (double)znak / imenitel;
    }
    pi*=4;
    printf("pi (priblizno) = %lf\n", pi);
    return 0;
}
```

Решение 1.2:

```
#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    double pi = 0, znak = 4, clen = znak;
    int imenitel = 1;
    do {
        pi += clen;
        imenitel += 2;
        znak = -znak;
        clen = znak / imenitel;
    } while (fabs(clen) > 1e-6);

    /* pi = clen;
    while (fabs(clen) > 1e-6)
        pi += clen = ((znak*=-1) / (imenitel+=2));*/

    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}
```

Решение 2:

```

#include <stdio.h>
#define _USE_MATH_DEFINES
#include <math.h>
int main() {
    int faktori = 1, faktorb = 1;
    double pi = 1, clen = 1;
    /* vo ciklusot se presmetuva pi/2 */
    do {
        faktori += 2;
        clen *= (double)faktorb / faktori;
        faktorb++;
        pi += clen;
    } while (clen > 1e-6);

    /* do
        pi += clen*=(double)faktorb++ / (faktori+=2);
        while (clen > 1e-6); */

    pi *= 2;
    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}

```

3.3. Задача за вежбање 3

Да се напише програма за приближно пресметување на π користејќи го изразот $4\pi = \arctan(1)$ и развојот на \arctan во ред.

Забелешка: Развојот на $\arctan(1)$ во ред со користење на двоен факториел е:

$$1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \frac{1 \cdot 2 \cdot 3 \cdot 4}{3 \cdot 5 \cdot 7 \cdot 9} + \dots = \frac{\pi}{2}$$

$$\sum_{n=0}^{\infty} \frac{n!}{(2n+1)!!} = \frac{\pi}{2}$$

Решение:

```
#include <stdio.h>
#include <math.h>
int main()
{
    int faktori = 1, faktorb = 1;
    double pi = 1, clen = 1;
    /* vo ciklusot se presmetuva pi/2 */
    do
    {
        faktori += 2;
        clen *= (double)faktorb / faktori;
        faktorb++;
        pi += clen;
    } while (clen > 1e-6);

    /* do
        pi += clen*=(double)faktorb++ / (faktori+=2);
    while (clen > 1e-6); */

    pi *= 2;
    printf("pi (priblizno) = %10.8lf \t pi = %10.8lf\n", pi, M_PI);

    return 0;
}
```

4. Изворен код од примери и задачи

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

Source code ZIP