

10. Dodatne vježbe

1. Napišite program koji učitava dva znaka te ispisuje sve znakove ASCII tablice koji se nalaze između ta dva znaka. Npr., ako se učitaju znakovi d i k, program ispisuje defghijk.
2. Načinite program koji će ispisati sljedeću tablicu:

```
A. a b c d e f .F
B. b c d e f g .G
C. c d e f g h .H
D. d e f g h i .I
... itd.
S. s t u v w x .X
T. t u v w x y .Y
U. u v w x y z .Z
```

Očekuje se da zadatak riješite pomoću dvije ugniježdene petlje, a ne npr. ovako:

```
printf("A. a b c d e f .F\n");
printf("B. b c d e f g .G\n");
printf("C. c d e f g h .H\n");
printf("D. d e f g h i .I\n");
... itd.
```

3. Prepravite program iz prethodnog zadatka tako da se na mjestima gdje bi se ispisao "mali" samoglasnik, umjesto toga ispiše znak '?'.
4. Načinite program za izračunavanje "m povrh n".

$$m! / (n! \cdot (m - n)!)$$

Vrijednosti za m i n učitati s tipkovnice uz kontrolu jesu li te vrijednosti ispravno zadane (cijeli brojevi veći ili jednaki 0, m je veći ili jednak n).

5. Ispišite sve pitagorine trojke čiji su članovi veći od 0 i manji ili jednaki 100. Ispis treba izgledati ovako:

```
1. trojka: 3^2 + 4^2 = 5^2
2. trojka: 4^2 + 3^2 = 5^2
3. trojka: 5^2 + 12^2 = 13^2
4. trojka: 6^2 + 8^2 = 10^2
5. trojka: 7^2 + 24^2 = 25^2
... itd.
101. trojka: 80^2 + 60^2 = 100^2
102. trojka: 84^2 + 13^2 = 85^2
103. trojka: 84^2 + 35^2 = 91^2
104. trojka: 96^2 + 28^2 = 100^2
```

Uputa: zadatak možete riješiti tako da pomoću tri ugniježdene petlje testirate svaku kombinaciju 3 cijela broja: 1 1 1; 1 1 2; 1 1 3; ... 1 1 99; 1 1 100; 1 2 1; 1 2 2; ...; 1 2 100; 1 3 1; ... Ispišite samo one kombinacije 3 cijela broja koji zadovoljavaju "uvjet pitagorine trojke".

6. Korisnik uzastopno upisuje cijele brojeve u intervalu [1, 9]. Za svaki učitani broj ispisati broj riječima. Zanimariti brojeve koji su izvan dopuštenog intervala. Učitavanje prekinuti kada se upiše broj nula. **Naputak:** koristiti naredbu `switch`. Npr.

```
3
tri
7
sedam
77
zanemarujem
1
jedan
0
```

7. Korisnik uzastopno upisuje cijele brojeve u intervalu [1, 9]. Za svaki učitani broj ispisati od kojih je sve brojeva učitani broj manji. Zanimariti brojeve koji su izvan dopuštenog intervala. Učitavanje prekinuti kada se upiše broj nula. **Naputak:** koristiti naredbu `switch` s "propadanjem po labelama". Npr.

```
3
manji je od 4
manji je od 5
manji je od 6
manji je od 7
manji je od 8
manji je od 9
77
zanemarujem
9
8
manji je od 9
0
```

8. Napišite program koji će ispisati prvih 25 prostih brojeva većih od 0.
9. Napišite program koji će ispisati sve proste brojeve iz intervala [1000, 10000].
10. S tipkovnice učitavati cijele brojeve dok god se naizmjenice upisuje jedan pozitivan, jedan negativan, jedan pozitivan broj, itd. Ako se dva puta za redom upiše pozitivan broj ili se dva puta za redom upiše negativan broj ili se upiše broj 0, program ispisuje sumu svih do tada upisanih brojeva i prekida se njegovo izvršavanje. Prvi broj koji se upiše s tipkovnice može biti ili pozitivan ili negativan.
11. Napišite program za izračunavanje $n!$ koji za realizaciju "petlje" koristi `goto` naredbu. **Napomena:** ovaj zadatak služi za vježbu, a ne za demonstraciju načina na koji bi trebalo rješavati zadatke s programskim petljama!

Ne gledajte ova rješenja dok sami ne pokušate riješiti zadatke!

Rješenje 2. zadatka

```
#include <stdio.h>

int main() {
    char i, j;
    for (i = 'A'; i <= 'U'; i++) {
        printf("%c. ", i);
        for (j = i + 32; j < i + 32 + 6; j++) {
            printf("%c ", j);
        }
        printf(":%c\n", i + 5);
    }
}
```

Vanjska petlja mijenja vrijednost varijable i od 'A' do 'U'. Na početku svakog retka se ispisuje vrijednost varijable i (naravno, u formatu %c), a na kraju retka ispisuje se slovo koje se u ASCII tablici nalazi "5 mjesta dalje" od slova koje se ispisalo na početku retka.

Unutarnja petlja mijenja vrijednost varijable j od "male verzije" slova koje je ispisano na početku retka, do slova koje se u ASCII tablici nalazi "6 mjesta dalje" od početnog malog slova.

Rješenje 4. zadatka

```
#include <stdio.h>

int main() {
    int m, n, i;
    double brojnik, naziv1, naziv2, mpovrh;

    /* unos vrijednosti za m i n */
    printf ("Unesite m i n:");
    scanf ("%d %d", &m, &n);
    if (m < 0 || n < 0 || m < n)
        printf("brojevi su neispravno zadani\n");
    else {
        brojnik = 1;
        for (i = 1; i <= m; i++)
            brojnik *= i;

        naziv1 = 1;
        for (i = 1; i <= n; i++)
            naziv1 *= i;

        naziv2 = 1;
        for (i = 1; i <= m-n; i++)
            naziv2 *= i;

        mpovrh = brojnik/(naziv1*naziv2);
        printf("%d povrh %d iznosi = %g\n", m, n, mpovrh);
    }
    return 0;
}
```

Rješenje 5. zadatka

```
#include <stdio.h>

int main() {
    int i, j, k;
    int n = 0;
    for (i=1; i <= 100; i++)
        for (j=1; j <= 100; j++)
            for (k=1; k <= 100; k++)
                if (i*i + j*j == k*k) {
                    n++;
                    printf("%d. trojka: %d^2 + %d^2 = %d^2\n", n, i, j, k);
                }
    return 0;
}
```

Rješenje 6. zadatka

```
#include <stdio.h>

int main() {
    int broj;
    do {
        scanf("%d", &broj);
        if (broj != 0)
            switch (broj) {
                case 1: printf("jedan\n");
                        break;
                case 2: printf("dva\n");
                        break;
                case 3: printf("tri\n");
                        break;
                case 4: printf("cetiri\n");
                        break;
                case 5: printf("pet\n");
                        break;
                case 6: printf("sest\n");
                        break;
                case 7: printf("sedam\n");
                        break;
                case 8: printf("osam\n");
                        break;
                case 9: printf("devet\n");
                        break;
                default: printf("Zanemarujem\n");
                        break;
            }
    } while (broj != 0);
    return 0;
}
```

Rješenje 7. zadatka

```
#include <stdio.h>

int main() {
    int broj;
    do {
        scanf("%d", &broj);
        if (broj != 0)
            switch (broj) {
                case 1: printf("manji je od 2\n");
                case 2: printf("manji je od 3\n");
                case 3: printf("manji je od 4\n");
                case 4: printf("manji je od 5\n");
                case 5: printf("manji je od 6\n");
                case 6: printf("manji je od 7\n");
                case 7: printf("manji je od 8\n");
                case 8: printf("manji je od 9\n");
                case 9: break;
                default: printf("Zanemarujem\n");
                        break;
            }
    } while (broj != 0);
    return 0;
}
```

Rješenje 8. zadatka

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

int main () {
    int testBroj = 0, pronadjenoProstih = 0;
    int i, jestProst;
    while (pronadjenoProstih < 25) {
        testBroj++;
        /* hipoteza: testBroj jest prost */
        jestProst = 1; /* true */
        for ( i = 2; i <= pow(testBroj, 0.5); i++)
            if( testBroj % i == 0 ) {
                /* oborena je hipoteza da je testBroj prost */
                jestProst = 0; /* false */
                break;
            }
        if (jestProst)
            printf("%d. %d\n", ++pronadjenoProstih, testBroj);
    }
    return 0;
}
```

ili

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

int main () {
    int testBroj = 0, pronadjenoProstih = 0;
    int i;
    while (pronadjenoProstih < 25) {
        testBroj++;
        i = 2;
        while ( i <= pow(testBroj, 0.5) && testBroj % i != 0)
            i++;
        if( i > pow(testBroj, 0.5) )
            /* petlja je prosla kroz sve djeljitelje <= pow(testBroj, 0.5) */
            printf("%d. %d\n", ++pronadjenoProstih, testBroj);
    }
    return 0;
}
```