

1. Napisati funkciju `korijeni` koja za zadani nenegativni cijeli broj `n` (tipa `unsigned int`) vraća tri vrijednosti tipa `double`: $n^{\frac{1}{2}}$, $n^{\frac{1}{3}}$ i $n^{\frac{1}{4}}$. Napisati glavni program (funkciju `main`) tako da s tipkovnice učitava jedan cijeli broj, pozove funkciju i rezultate dobivene pozivom funkcije ispiše na zaslon sukladno prikazanom primjeru.

Primjer izvršavanja programa.

```
Upisite nenegativni cijeli broj > 12↵
Rezultati su:↵
3.46410162↵
2.28942849↵
1.64375183
```

2. Napisati funkciju `maks` koja kao parametre prima dva pokazivača na objekte tipa `int`, određuje veću od dvije vrijednosti u ta dva objekta i tu vrijednost (cijeli broj) vraća *preko imena* funkcije. Napisati glavni program (funkciju `main`) tako da s tipkovnice učitava dva cijela broja, pozove funkciju i rezultat dobiven pozivom funkcije ispiše na zaslon sukladno prikazanim primjerima.

Primjeri izvršavanja programa.

```
Upisite dva cijela broja > 12 15↵
Rezultat je 15
```

```
Upisite dva cijela broja > 12 12↵
Rezultat je 12
```

3. Napisati funkciju `poredaj` koja kao parametre prima tri pokazivača na objekte tipa `double` (pokazivače na prvi objekt, drugi objekt i treći objekt). Funkcija treba *vrijednosti* u ta tri objekta zamijeniti tako da vrijednost prvog objekta bude manja ili jednaka vrijednosti drugog objekta, a vrijednost drugog objekta manja ili jednaka vrijednosti trećeg objekta. Npr. ako je funkcija primila pokazivače na objekte s vrijednostima 5., 3., 7. tada njihove vrijednosti treba zamijeniti tako da objekti redom sadrže vrijednosti 3., 5., 7.

Napisati glavni program (funkciju `main`) koji će učitati vrijednosti za tri varijable tipa `double`, pozvati funkciju `poredaj` i ispisati nove vrijednosti tih varijabli.

Primjeri izvršavanja programa.

```
Upisite tri realna broja > 5. 3. 7.↵
3.000000 5.000000 7.000000
```

```
Upisite tri realna broja > 5. 2. 2.↵
2.000000 2.000000 5.000000
```

4. Što je neispravno u sljedećem programu?

```
#include <stdio.h>

int *vratiDvostuko(int n) {
    int rez;
    rez = 2 * n;

    return &rez;
}

int main(void) {
    int a = 10, *rez = NULL;
    rez = vratiDvostuko(a);
    printf("Rezultat je %d", *rez);

    return 0;
}
```

5. Što je neispravno u sljedećem programu?

```
#include <stdio.h>

int main(void) {
    int a = 10, *p1 = &a, b = 20, *p2, c;
    c = *p1 + *p2;
    printf("c = %d", c);

    return 0;
}
```

6. Što će se ispisati izvršavanjem sljedećeg programa? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

```
#include <stdio.h>

int main(void) {
    int polje[] = {2, 3, 5, 7, 11, 23, 29};
    int *p1 = &polje[0], *p2 = NULL;
    p2 = &polje[5];
    printf ("%d %d\n", *p1, *p2);
    printf ("%d %d\n", *(p2 - 1), *p2 - 1);
    printf ("%d %d\n", p2 - p1, *p2 - *p1);

    return 0;
}
```

7. Što će se ispisati izvršavanjem sljedećeg programa? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

```
#include <stdio.h>

int main(void) {
    int polje[4][3] = {{2, 11, 23},
                      {29, 31, 37},
                      {47, 51, 59},
                      {61, 67, 71}
    };

    int *p1 = NULL, *p2 = NULL, *p3 = NULL;
    p1 = &polje[0][1];
    p2 = p1 + 5;
    p3 = &polje[3][1];
    printf ("%d %d\n", *p1, *p2);
    printf ("%d %d\n", *(p1 + 3), *p1 + 3);
    printf ("%d %d\n", p3 - p2, *p3 - *p2);

    return 0;
}
```

8. Što će se ispisati izvršavanjem sljedećeg programa? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

```
#include <stdio.h>

int main(void) {
    int polje[2][3] = {{2, 11, 23}, {47, 51, 59}};
    int *p = &polje[0][1], i = 3;
    p += i;
    * (--p) = --i;
    printf ("%d %d %d %d\n", *p, polje[1][0], polje[1][1], polje[1][2]);

    return 0;
}
```

9. Što će se ispisati izvršavanjem sljedećeg programa? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

```
#include <stdio.h>

int main(void) {
    int polje[4][3] = {{2, 11, 23},
                       {29, 31, 37},
                       {47, 51, 59},
                       {61, 67, 71}
                      };

    int *p = polje[0];
    int i1 = (*p)++;
    int i2 = *p++;
    printf ("%d %d\n", i1, i2);

    p = &polje[0][0];
    i1 = ++*p;
    i2 = *++p;
    printf ("%d %d\n", i1, i2);

    p = &polje[1][1];
    i1 = ++*--p;
    p = &polje[2][2];
    i2 = ++*p--;
    printf ("%d %d\n", i1, i2);

    return 0;
}
```

Rješenja:

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

void korijeni(unsigned int n, double *x2, double *x3, double *x5) {
    *x2 = pow(n, 1. / 2.);
    *x3 = pow(n, 1. / 3.);
    *x5 = pow(n, 1. / 5.);
    return;
}

int main(void) {
    unsigned int n;
    double a, b, c;

    printf("Upisite nenegativni cijeli broj > ");
    scanf("%u", &n);

    korijeni(n, &a, &b, &c);

    printf("Rezultati su:\n%.8lf\n%.8lf\n%.8lf", a, b, c);

    return 0;
}
```

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

int maks(int *p1, int *p2) {
    int rez;
    if (*p1 > *p2) {
        rez = *p1;
    } else {
        rez = *p2;
    }

    return rez;
}

int main(void) {
    int m, n;

    printf("Upisite dva cijela broja > ");
    scanf("%d %d", &m, &n);

    printf("Rezultat je %d", maks(&m, &n));

    return 0;
}
```

3. `#include <stdio.h>`

```
void zamijeni(double *p1, double *p2) {
    double pom;
    pom = *p1;
    *p1 = *p2;
    *p2 = pom;

    return;
}

void poredaj(double *p1, double *p2, double *p3) {
    if (*p1 > *p2) {
        zamijeni(p1, p2);
    }
    if (*p1 > *p3) {
        zamijeni(p1, p3);
    }
    if (*p2 > *p3) {
        zamijeni(p2, p3);
    }

    return;
}

int main(void) {
    double x, y, z;

    printf("Upisite tri realna broja > ");
    scanf("%lf %lf %lf", &x, &y, &z);

    poredaj(&x, &y, &z);

    printf("%lf %lf %lf", x, y, z);

    return 0;
}
```

4. Funkcija vraća pokazivač na objekt (varijablu) koji je definiran u funkciji. Taj objekt više ne postoji kada funkcija završi (funkcija je vratila viseći pokazivač - *dangling pointer*).

5. Varijabla p2 nije inicijalizirana, stoga sadrži pokazivač koji pokazuje na neodređeno mjesto u memoriji. Rezultat izvršavanja programa je nepredvidiv.

6. -

7. -

8. -

9. -