

Napomene:

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

18. vježbe uz predavanja

1. U modulu `prime.c` napišite definicije funkcija `resetPrime()` i `getNextPrime()`. U datoteci `prime.h` napišite njihove deklaracije. U modulu `glavni.c` nalazi se funkcija `main` koja ilustrira što trebaju raditi funkcije `resetPrime` i `getNextPrime`.

```
#include <stdio.h>
#include "prime.h"

int main (void) {
    resetPrime();
    printf("Inicijaliziran je generator prostih brojeva.\n");

    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());

    resetPrime();
    printf("Ponovo je inicijaliziran generator prostih brojeva.\n");

    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());
    printf("%d\n", getNextPrime());

    return 0;
}
```

Rezultat izvršavanja prikazane funkcije `main`:

```
Inicijaliziran je generator prostih brojeva.
2
3
5
Ponovo je inicijaliziran generator prostih brojeva.
2
3
5
7
11
13
```

U svim sljedećim zadacima u kojima se traži definiranje funkcije, treba napisati odgovarajući glavni program (tj. funkciju `main`) u kojem ćete po potrebi definirati stvarne argumente, pozvati funkciju i ispisati rezultat.

2. Napišite funkciju `toAbsDim` koja vrijednosti elemenata cjelobrojnog jednodimenzijskog polja mijenja u njihove apsolutne vrijednosti. Funkciju, glavni program i prototipove smjestite u tri zasebne datoteke. **Testirati prevođenje na dva načina:**
 - tako da se oba modula prevedu i povežu samo jednim pozivom prevodioca
 - tako da se zasebno prevede svaki modul, a zatim se dobiveni objektni kôd poveže u izvršni kôd

3. Napišite funkciju koja u zadanom jednodimenzijskom realnom polju prebroji koliko članova je veće od 0.0, koliko članova je manje od 0.0 i koliko članova je jednako 0.0. Dobivene vrijednosti funkcija mora vratiti u pozivajući program. Funkciju, glavni program i prototipove smjestite u tri zasebne datoteke. **Testirati prevođenje na dva načina:**
 - tako da se oba modula prevedu i povežu samo jednim pozivom prevodioca
 - tako da se zasebno prevede svaki modul, a zatim se dobiveni objektni kôd poveže u izvršni kôd

4. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
int main (void) {
    int a = 2, x = 10;
    int *p = &a;
    x += *p * 3;
    printf ("%d %d\n", *p, x);
    return 0;
}
```

5. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
void f (int *p) {
    printf ("%d %d\n", *p, *p+1);
}
int main (void) {
    int polje[6] = {1, 2, 3, 4, 5, 6};
    int *pp;
    pp = &polje[0];
    f(pp++);
    f(pp);
    f(++pp);
    return 0;
}
```

6. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
void f (int *p) {
    static int i = 2;
    printf ("%d\n", *(p + ++i));
}

int main (void) {
    int polje[8] = {1, 2, 3, 4, 5, 6, 7, 8};
    f(&polje[0]);
    f(&polje[0]);
    f(&polje[0]);
    f(&polje[1]);
    return 0;
}
```

7. Što će se ispisati sljedećim programom:

```
#include <stdio.h>
void f (int *p) {
    int i = 3;
    printf ("%d\n", *(p + --i));
}

int main (void) {
    int polje[6] = {1, 2, 3, 4, 5, 6};
    f(&polje[0]);
    f(&polje[1]);
    f(&polje[2]);
    return 0;
}
```

Rješenja

Rješenje 1. zadatka

prime.h

```
void resetPrime(void);  
int getNextPrime(void);
```

prime.c

```
#include <math.h>  
#include "prime.h"  
  
static int lastPrime = 0;  
  
void resetPrime(void) {  
    lastPrime = 0;  
}  
  
int getNextPrime(void) {  
    int i, isPrime;  
    if (lastPrime == 0)  
        lastPrime = 2;  
    else  
        do {  
            ++lastPrime;  
            isPrime = 1;  
            for (i = 2; isPrime && i <= sqrt(lastPrime); ++i)  
                if (lastPrime % i == 0)  
                    isPrime = 0;  
        } while (isPrime == 0);  
    return lastPrime;  
}
```

Rješenje 2. zadatka

glavni.c

```
#include <stdio.h>
#include "toapsdim.h"

#define MAXDIM 100

int main (void) {
    int m;
    int polje[MAXDIM];
    int i;

    printf ("Upisite m manji ili jednak %d: ", MAXDIM);
    scanf("%d", &m);
    printf ("Upisite elemente polja:\n");
    for (i = 0; i < m; ++i)
        scanf("%d", &polje[i]);

    printf("\nSlijedi ispis ucitanog niza\n\n");
    for (i = 0; i < m; ++i)
        printf("%d ", polje[i]);

    toApsDim(&polje[0], m);

    printf("\nSlijedi ispis izmijenjenog polja\n\n");
    for (i = 0; i < m; ++i)
        printf("%d ", polje[i]);
    return 0;
}
```

toapsdim.c

```
#include "toapsdim.h"

void toApsDim(int *polje, int n) { /* ili (int polje[], int n) */
    int i;
    for (i = 0; i < n; ++i)
        if (*(polje + i) < 0)
            *(polje + i) = - *(polje + i);
}
```

toapsdim.h

```
void toApsDim(int *polje, int n);
```

Rješenje 3. zadatka

funkcije.h

```
void prebroji(float *polje, int n,  
             int *vecihOdNula, int *manjihOdNula, int *jednakihNula);
```

funkcije.c

```
#include "funkcije.h"  
void prebroji(float *polje, int n,  
             int *vecihOdNula, int *manjihOdNula, int *jednakihNula) {  
    int i;  
    *vecihOdNula = *manjihOdNula = *jednakihNula = 0;  
    for (i = 0; i < n; ++i)  
        if (*(polje + i) < 0.0)  
            (*manjihOdNula)++;  
        else if (*(polje + i) == 0.0)  
            (*jednakihNula)++;  
        else  
            (*vecihOdNula)++;  
    return;  
}
```

test.c

```
#include <stdio.h>  
#include "funkcije.h"  
  
#define MAXDIM 100  
  
int main (void) {  
    int m;  
    float polje[MAXDIM];  
    int i;  
    int vecih, manjih, jednakih;  
  
    printf ("Upisite m manji ili jednak %d: ", MAXDIM);  
    scanf("%d", &m);  
    printf ("Upisite elemente polja:\n");  
    for (i = 0; i < m; ++i)  
        scanf("%f", &polje[i]);  
  
    printf("\n\nSlijedi ispis ucitanog niza\n");  
    for (i = 0; i < m; ++i)  
        printf("%f\n", polje[i]);  
  
    prebroji(&polje[0], m, &vecih, &manjih, &jednakih);  
    printf("\nn>0 ima: %d      n<0 ima: %d      n==0 ima: %d\n",  
           vecih, manjih, jednakih);  
    return 0;  
}
```