

## 18. Dodatne vježbe

U svim 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.

1. Što će se ispisati tijekom izvođenja sljedećeg programa:

modul `proto.h`

```
void fun1(void);
void fun2(void);
void fun3(void);
void fun4(void);
```

modul `glavni.c`

```
#include <stdio.h>
#include "proto.h"
extern int x;

int main() {
    int x = 30;
    x += 2;
    printf("%d\n", x);
    fun1();
    fun2();
    fun3();
    fun4();
    fun3();
    return 0;
}

void fun1(void) {
    x += 3;
    printf("%d\n", x);
}
```

modul `modulA.c`

```
#include <stdio.h>
#include "proto.h"
extern int x = 20;

void fun2(void) {
    x += 4;
    printf("%d\n", x);
}
```

modul `modulB.c`

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

void fun3(void) {
    static int x = 5;
    x += 5;
    printf("%d\n", x);
}

void fun4(void) {
    extern int x;
    x += 6;
    printf("%d\n", x);
}
```

2. Napišite funkciju `toApsDim` koja vrijednosti elemenata cjelobrojnog jednodimenzionalnog polja mijenja u njihove apsolutne vrijednosti. Funkciju, glavni program i prototipove smjestite u tri zasebna modula.
3. Napišite funkciju koja u zadanom nizu znakova (jnodimenzionalnom polju znakova terminiranom sa znakom `'\0'`) pronalazi sve samoglasnike i ispisuje ih na ekran. Npr. za zadani niz *Antigona*, ispisuje *Aioa*.
4. Napišite funkciju koja iz zadanog niza znakova **izbacuje** sve samoglasnike. Npr. ako se funkciji zada niz znakova *Antigona*, funkcija ga mora **promijeniti** u niz znakova *ntgn*.
5. Napišite funkciju koja u zadanom jednodimenzionalnom 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.
6. Što će se ispisati sljedećim programom:

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

7. Što će se ispisati sljedećim programom:

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

8. Što će se ispisati sljedećim programom:

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

int main () {
    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;
}
```

9. Što će se ispisati sljedećim programom:

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

**Rješenja svih zadataka provjeriti prevođenjem i testiranjem vlastitih programa!**

## Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

### Rješenje 1. zadatka

Ovdje je važno uočiti postojanje **definicija** triju **različitih** varijabli x.

- na početku modula modulA.c **definirana** je eksterna varijabla x
- na početku funkcije main **definirana** je automatska varijabla x
- na početku funkcije fun3 **definirana** je statička varijabla x

Nakon što se odredi trajnost i područje važenja svake pojedine varijable, zadatak je lako riješiti.

### Rješenje 2. zadatka

#### glavni.c

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

#define MAXDIM 100

int main () {
    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

```
#include <stdio.h>

void ispisSamoglasa (char niz[]);

int main () {
    char niz[] = "Antigona";
    ispisSamoglasa(niz);
    printf("\n");
    return 0;
}

void ispisSamoglasa (char niz[]) {          /* ili bolje (char *niz) */
    int i = 0;
    while (niz[i] != '\0') {
        if (niz[i] == 'a' || niz[i] == 'A' ||
            niz[i] == 'e' || niz[i] == 'E' ||
            niz[i] == 'i' || niz[i] == 'I' ||
            niz[i] == 'o' || niz[i] == 'O' ||
            niz[i] == 'u' || niz[i] == 'U')
            printf("%c", niz[i]);
        i++;
    }
}
```

### Rješenje 4. zadatka

```
#include <stdio.h>

void izbaciSamoglase (char *niz);

int main () {
    char niz[] = "Antigona";
    printf("%s\n", niz);
    izbaciSamoglase(niz);
    printf("%s\n", niz);
    return 0;
}

void izbaciSamoglase (char *niz) {          /* ili (char niz[]) */
    int i = 0, potroseno = 0;
    while (*(niz + i) != '\0') {
        if (*(niz + i) != 'a' && *(niz + i) != 'A' &&
            *(niz + i) != 'e' && *(niz + i) != 'E' &&
            *(niz + i) != 'i' && *(niz + i) != 'I' &&
            *(niz + i) != 'o' && *(niz + i) != 'O' &&
            *(niz + i) != 'u' && *(niz + i) != 'U')
            *(niz + potroseno++) = *(niz + i);
        i++;
    }
    *(niz + potroseno) = '\0'; /* VAZNO PITANJE: sto bi bio rezultat bez ove naredbe? */
}
```

Ovdje je važno uočiti da u funkciji **nije moguće** koristiti pomoćni niz jer funkcija "ne može znati" kolika je najveća dopuštena duljina ulaznog niza (a taj je podatak potreban pri definiciji pomoćnog niza). Tek kad bi zadatak glasio: napišite funkciju koja iz zadanog niza znakova izbacuje sve samoglasnike, pri čemu zadani niz **sigurno nije duži od 1000 znakova**, tada bi postojala mogućnost u funkciji koristiti pomoćni niz (tada bi se u funkciji mogao definirati pomoćni niz duljine 1001 znaka).

## Rješenje 5. zadatka

```
#include <stdio.h>

#define MAXDIM 100

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;
}

int main () {
    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 ("\n\n>0 ima: %d      n<0 ima: %d      n==0 ima: %d\n",
            vecih, manjih, jednakih);
    return 0;
}
```