

## Auditorne vježbe 4.

- 4.1 Prikažite *tačan izgled ekrana* na kraju izvršavanja ovog C++ programa, uz kratko obrazloženje zbog čega su rezultati onakvi kakvi jesu. Oprez: bitan je svaki razmak, kao i prelasci u nove redove. Radi jasnoće, razmake prikažite kao kvadratiće.

```
#include <iostream>
#include <iomanip>
#include <complex>

using namespace std;

void F1(int a, int b) {
    a = b + 3; b = a - 1;
    cout << a << b;
}

void F2(int &a, int &b) {
    a = b + 3; b = a - 1;
    cout << a << b;
}

int *F3(int a, int *b) {
    cout << (*b)++;
    return new int(a);
}

int main() {
    int x(4);
    complex<double> c(4), i(0, 1);
    cout << setw(5) << c * c << c * i << setw(9) << c * i - c << endl;
    F1(x, x);
    cout << x << endl;
    F2(x, x);
    cout << x << endl;
    *F3(x, &x) = 7;
    cout << x << endl;
    return 0;
}
```

- 4.2 Napišite funkciju “*DaLiJePalindrom*” koja za nul-terminirani niz znakova koji joj je proslijeđen kao parametar ispituje da li predstavlja palindrom ili nije, i kao rezultat vraća odgovarajuću logičku vrijednost “**true**” ili “**false**” (vodite računa da je parametar klasični nul-terminirani niz znakova, a ne dinamički string, odnosno objekat tipa “*string*”). Pri tome, funkciju treba realizirati isključivo korištenjem pokazivačke aritmetike. Drugim riječima, nije dozvoljeno indeksiranje, niti njegovo trivijalno simuliranje (tj. upotreba konstrukcija poput “*\*(p+i)*” umjesto “*p[i]*”). Također nije dozvoljena upotreba nikakvih pomoćnih struktura podataka, poput dopunskih nizova ili vektora. Napišite i kratki testni program (“*main*” funkciju) u kojem ćete testirati napisanu funkciju.

**Napomena:** Pod palindromima smatramo riječi ili rečenice koje se isto čitaju sa obe strane (npr. “*kapak*”). Prilikom ispitivanja treba ignorirati eventualne razmake, interpunkcijske znake i razliku između velikih i malih slova, tako da rečenica “*Ana voli*

*Milovana*” treba da bude prepoznata kao palindrom, iako bukvalno pročitana sa suprotnog kraja glasi *“anavoliM ilovAna”*.

- 4.3 Napišite funkciju *“NadjiPodstring”* koja obavlja potpuno istu funkciju kao i funkcija *“strstr”* iz biblioteke *“cstring”*. Ova funkcija prima kao parametre pokazivače na prve elemente dva niza znakova (koji predstavljaju dva stringa) i vraća kao rezultat pokazivač na prvo mjesto unutar prvog stringa na kojem se drugi string nalazi kao njegov podstring, odnosno vraća nul-pokazivač ukoliko se drugi string ne nalazi unutar prvog stringa kao njegov podstring. Funkciju treba realizirati isključivo korištenjem pokazivačke aritmetike. Napisanu funkciju testirati u testnom programu koji je principijelno isti kao primjer iz skripte u kojem je demonstrirana funkcija *“strstr”*.
- 4.4 Napišite generičku funkciju *“KreirajNiz”* koja prima dva parametra. Prvi parametar predstavlja broj elemenata nekog niza, a drugi parametar je vrijednost kojom treba popuniti elemente niza. Funkcija treba da dinamički kreira niz sa zadanim brojem elemenata, da ga popuni zadanom vrijednošću, i da vrati kao rezultat pokazivač na prvi element novokreiranog niza. U slučaju da kreiranje ne uspije, funkcija treba da baci izuzetak koji se sastoji od teksta *“Kreiranje nije uspjelo”*. Zatim, za tako napisanu funkciju odgovorite koji će biti tip elemenata kreiranog niza ako se izvrši svaki od sljedećih poziva:

a) <code>KreirajNiz(10, 3);</code>	b) <code>KreirajNiz&lt;float&gt;(15, 5);</code>
c) <code>KreirajNiz(14, 2.);</code>	d) <code>KreirajNiz(8, 'a');</code>
e) <code>KreirajNiz&lt;int&gt;(27, 'b');</code>	f) <code>KreirajNiz(12, "c");</code>
g) <code>KreirajNiz&lt;string&gt;(44, "d");</code>	

- 4.5 Napišite funkciju sa jednim cjelobrojnim parametrom  $n$  koja dinamički alocira niz od  $n$  cijelih brojeva, popunjava ga sa prvih  $n$  prostih brojeva, i vraća pokazivač na prvi element tako alociranog niza. Ukoliko alokacija ne uspije ili ukoliko je  $n$  negativan ili nula, funkcija treba baciti izuzetak. Napisanu funkciju demonstrirajte u testnom programu koji ilustrira kako biste mogli pozvati ovu funkciju i osloboditi alociranu memoriju kada ona više nije potrebna. Predvidite i hvatanje eventualno bačenih izuzetaka.