

**PREDMET:** Programiranje II  
**Datum:** 10.5.2024.

**Rješenje semestralnog ispita**  
(GRUPA II)

1. Napišite funkciju  $F$  koja u vidu formalnog parametra prima vektor  $v$  sastavljen od  $n$  realnih brojeva  $a_1, a_2, \dots, a_n$ , a kao rezultat računa i vraća vrijednost sljedećeg izraza

$$\sqrt{a_1 + \sqrt{a_2 + \sqrt{\dots + a_n}}}$$

U slučaju da su elementi vektora  $v$  takvi da je potkorijena veličina manja od nule, funkcija treba baciti izuzetak. Obavezno predvidite i hvatanje eventualno bačenih izuzetaka iz funkcije. U glavnom programu (**main funkciji**) treba pisati nešto poput sljedećeg

```
int main() {
    int n; std::vector<double>v(n); for(int i(0);i<n;i++) std::cin>>v[i];
    std::cout<<F(v);
    return 0;
}
```

Rezultati na izlazu trebaju biti zaokruženi na 5 decimala. Ispod se nalaze primjeri testiranja tako da za date ulaze trebate dobiti željene izlaze.

ULAZ		IZLAZ
$n$	Elementi vektora $v$	Izlaz iz funkcije $F$
5	100 2 -38 4 54	Podkorijena velicina je negativna.
6	1 4 9 90 3 77	1.9709
10	9.1 8.3 7.5 5.2 6.1 2.2 1.1 9.9 1.5 2.1	3.5346

Izgled ispravnog testiranja kroz web aplikaciju za drugi test iz gornje tabele izgleda kao na slici ispod:

SEKCIJA: Unesite ulaz

6 1 4 9 90 3 77

Testiraj Pošalji Obriši

Rezultat:

1.9709

Rješenje.

```
#include <iostream>
#include <vector>
#include <cmath>
#include<iomanip>
double F(std::vector<double> v) {
    double suma(0);
    int i(v.size() - 1);
    while (i >= 0) {
        suma += v[i];
        if (suma < 0) throw std::domain_error("Podkorijena velicina je negativna.");
        suma = std::sqrt(suma);
    }
```

```

        i--;
    }
    return suma;
}
int main() {
    try {
        int n; std::cin >> n; std::vector<double>v(n); for (int i(0); i <
n; i++)std::cin >> v[i];
        std::cout<<std::setprecision(5)<<F(v);
    }
    catch (std::domain_error poruka) {
        std::cout << poruka.what();
    }
}

```

2. Napišite **funkciju**  $f$  sa tri parametra tako da prvi parametar bude neki cijeli broj  $n$ . Funkcija treba formirati dva nova broja koji se sastoje respektivno od prostih i složenih cifara polaznog broja, u istom redoslijedu u kojem se nalaze u polaznom broju  $n$ . Novoformirane brojeve smjestite redom u drugi  $m$  i treći  $k$  parametar funkcije  $f$ . Na primjer, ukoliko se kao prvi parametar zada broj 123456789, u drugi i treći parametar treba da se redom smjeste brojevi 2357 i 14689. Znak broja trebete ignorisati, odnosno isti efekat se dobija ukoliko se kao prvi parametar zada broj -123456789. Glavnu funkciju „main“ napišite tako da se sa tastature može unijeti cijeli broj  $n$ , te pozvati funkcija  $f$ . Ispod se nalaze ulazi, kao i odgovarajući izlazi koje trebete dobiti.

ULAZ	IZLAZ	
$n$	$m$	$k$
123456789	2357	14689
-1234567	2357	146
24681012	22	468101
13355779	335577	19
22446642	222	44664

Dakle, sa tastature se treba samo unijeti vrijednost broja  $n$ . Neposredno nakon poziva funkcije  $f$ , ispisati na ekranu sadržaj drugog i trećeg parametara funkcije  $f$ .

*Rješenje.*

```

#include <iostream>
#include <cmath>
bool jel_prost(int n){
    if(n <= 1)return false;
    for (int i(2); i*i<=n;++i) if(n%i==0)return false;
    return true;
}
void f(long int n, long int& parni, long int& neparni) {
    int suma1(0), suma2(0);
    n = std::abs(n);
    while (n != 0) {
        if (jel_prost(n % 10)) {
            suma1 += n % 10;
            suma1 *= 10;
        }
        else {
            suma2 += n % 10;
            suma2 *= 10;
        }
        n /= 10;
    }
    suma1 = suma1 / 10;
    suma2 = suma2 / 10;
    int sumax(0), sumay(0);
    while (suma1 != 0) {
        sumax += suma1 % 10;

```

```

        sumax *= 10;
        suma1 /= 10;
    }
    while (suma2 != 0) {
        sumay += suma2 % 10;
        sumay *= 10;
        suma2 /= 10;
    }
    parni = sumax / 10;
    neparni = sumay / 10;
}
int main() {
    long int n, x, y;
    std::cin >> n;
    f(n, x, y);
    std::cout<<x<<" "<<y;
    return 0;
}

```

3. Napišite generičku funkciju  $f$ , koja kao parametar prima dva objekta tipa `std::set`, tj. dva skupa  $A$  i  $B$ , čiji su elementi istog tipa, a koja kao rezultat vraća novi skup  $C$  istog tipa kao i skupovi  $A$  i  $B$ , koji predstavlja simetričnu razliku skupova  $A$  i  $B$ . **Simetrična razlika** se definira kao skup koji sadrži one i samo one elemente koji se nalaze ili u jednom, ili u drugom skupu, ali ne u oba skupa istovremeno. Na primjer, ukoliko se skupovi  $A = \{4, 0, 12, 6, 10, -4\}$  i  $B = \{4, 11, 6, -4, -5\}$  proslijede funkciji  $f$ , ona kao rezultat treba vratiti skup  $C = \{-5, 0, 10, 11, 12\}$ . Glavnu funkciju „main“ napišite tako da se sa tastature mogu unijeti prirodni projevi  $n_a$  i  $n_b$ , te  $n_a$  elemenata skupa  $A$  i  $n_b$  elemenata skupa  $B$ . Ispod se nalaze **ulazi**, kao i odgovarajući **izlazi** koje trebate dobiti.

ULAZ				IZLAZ
$n_a$	Elementi skupa $A$	$n_b$	Elementi skupa $B$	Izlaz iz funkcije $f$
6	4 0 12 6 10 -4	6	4 11 6 -4 -5 1000	-5 0 10 11 12 1000
4	1 2 3 4	6	-10 2 3 90 5 1	-10 4 5 90
5	1.1 3.1 2.3 -4.9 9.9	7	1.2 2.3 -3.1 -4.1 6.7 1.1 5.2	-4.9 -4.1 -3.1 1.2 3.1 5.2 6.7 9.9

Dakle, sa tastature se prvo unosi broj elemenata  $n_a$  skupa  $A$ , pa zatim se unose elementi skupa  $A$ , isto važi i za skup  $B$ . Također, sadržaj skupa  $C = f(A, B)$  se treba ispisati na ekranu i on predstavlja izlaz funkcije  $f$ .

Rješenje.

```

#include <iostream>
#include <set>
using namespace std;
template<typename tip>
bool nalazi_se(const set<tip>& s, tip a) {
    for (auto it(s.begin()); it != s.end(); it++) if (*it == a) return true;
    return false;
}
template <typename tip>
set<tip>f(const set<tip>& s1, const set<tip>& s2) {
    set<tip> razlika;
    for (auto it(s1.begin()); it != s1.end(); it++) if (!nalazi_se(s2, *it)) razlika.insert(*it);
    for (auto it(s2.begin()); it != s2.end(); it++) if (!nalazi_se(s1, *it)) razlika.insert(*it);
    return razlika;
}
int main() {
    int na(0); std::cin >> na; set<double>A;
    for (int i(0); i < na; i++) {
        double x; std::cin >> x; A.insert(x);
    }
}

```

```
int nb(0); std::cin>>nb; set<double>B;  
for(int i(0);i<nb;i++) {  
    double x; std::cin >> x; B.insert(x);  
}  
set<double> C=f(A, B);  
for (auto it(C.begin()); it != C.end(); it++) cout << *it << " ";  
return 0;  
}
```