**Zadaca 3 - dodatak**

GL HF

**Zadatak 1.**

Kreirati program koji implementira funkciju svicarski\_noz koja zavisno od tipa i broja argumenata vrsi razlicite stvari. Implementaciju zakljuciti na osnovu primjera upotrebe:

| Upotreba (kod) | Izlaz |
| --- | --- |
| std::vector<int> niz{1,2,3};  svicarski\_noz(2);  svicarski\_noz(3);  svicarski\_noz(4);  svicarski\_noz(“pozdrav”,”svima”);  svicarski\_noz(3.14);  svicarski\_noz(niz);  svicarski\_noz();  std::cout << svicarski\_noz(“pozdrav”);  std::cout << svicarski\_noz(svicarski\_noz(“pozdrav”)); | 4  8  16  pozdrav svima  3.14 6.28 9.42  1 2 3  11:04:10  vardzop  pozdrav |

**Zadatak 2.**

Kreirati program koji racuna najduzu distancu izmedju dvije tacke medju N tacaka. Korisnik unosi tacke dok ne zaustavi unos (CTRL + D). Nakon toga potrebno je ispisati najduzu distancu i dvije tacke koje cine istu.

| Upotreba | Izlaz |
| --- | --- |
| 2 3  1 2  4 8 | (1, 2) (4, 8) = 6.7 |

Tacku opisati uz pomoc strukture:

struct tacka{

int x;

int y;

};

**Zadatak 3.**

Analizirajte sljedeci kod:

#include <iostream>

void f1(int a){

if(a>10)

throw std::string("CIJELI BROJ VECI OD 10");

}

void f2(double c)

{

try{

f1(11);

throw std::string("IZNIMKA 2");

}

catch(int d)

{

std::cout << "CIJELI BROJ 2" << std::endl;

}

}

void f1(double b){

try

{

f1(5);

f2(b);

}

catch(...)

{

std::cout << "IZNIMKA 1" << std::endl;

if(b>2.3)

throw 2.5;

if(b<2.3)

throw 2;

throw 1;

}

}

int main()

{

double broj = 2.3;

try{

f1(broj);

}

catch(std::string iznimka){

std::cout << iznimka << std::endl;

}

catch(int broj){

std::cout << broj << std::endl;

if(broj > 1)

throw true;

}

catch(...){

std::cout << "NE ZNAM" << std::endl;

}

return 0;

}

Sta je ispis ukoliko je varijabla broj jednaka 2.2, 2.3 i 2.4? Detaljno objasniti za svaki slucaj.

**Zadatak 4.**

Kreirati program koji kreira vektor stringova sa rijecima po vasem izboru. Potrebno je kreirati sljedece funkcije:

* ispisi - kao parametre prima vektor i ostream objekat te ispisuje sadrzaj vektora uz pomoc ostream objekta
* dodaj - kao parametar prima rijec i vektor u koji ce dodati rijec na kraj
* kopiraj - kao parametre prima dva vektora, te kopira sadrzaj prvog u drugi vektor, brisajuci vec postojece elemente drugog
* izvrsi - kao parametre prima vektor i lambdu, te poziva lambdu nad svakim clanom vektora
* obrisi - kao parametar prima vektor, te brise citav sadrzaj vektora
* obrisiRijec - kao parametre prima vektor i rijec. Brise proslijedjenu rijec iz proslijedjenog vektora
* sortiraj - kao parametre prima vektor i lambdu kao kriterij za sortiranje, te sortira vektor na osnovu proslijedjene lambda. Lambda treba da bude default argumentovana tako da ukoliko je korisnik ne proslijedi bude sortirano leksikografski
* dodajVektor - kao parametar prima dva vektora, te sadrzaj prvog dodaje na kraj drugog
* obrisiIsteRijeci - kao parametre prima dva vektora, te iz drugog vektora brise sve rijeci koje se pojavljuju u prvom vektoru kao i u drugom vektoru
* obrisiRazliciteRijeci - kao parametre prima dva vektora, te iz drugog vektora brise sve rijeci koje se ne pojavljuju u prvom vektoru a pojavljuju u drugom vektoru
* izvrsiNPuta - kao parametre prima vektor, lambdu i cijeli broj. Proslijedjenu lambdu poziva nad svakim clanom niza onoliko puta koliko iznosi proslijedjeni cijeli broj.

Deklaracije funkcija odvojiti u header fajl pod nazivom funkcije.hpp. Implementirati funkcije u cpp fajl naziva funkcije.cpp. Testirati funkcionalnost unutar main.cpp. Povratne tipove funkcija zakljuciti na osnovu upotrebe u sljedecem primjeru:

std::vector<string> baza1 {“emir”, “mirza”, “meliha”, “jasmin”, “dzemo”};

std::vector<string> baza2;

kopiraj(baza1, baza2);

dodaj(baza1, “dino”);

dodaj(baza2, “samir”);

obrisiIsteRijeci(baza1, baza2);

obrisiRazliciteRijeci(baza2, baza1);

dodaj(baza2, “emina”);

dodaj(baza1, “muharem”);

dodajVektor(baza1, baza2);

sortiraj(baza2);

izvrsiNPuta(baza2,[](string& rijec){rijec[0]++;},3);

dodajVektor(baza2, baza2);

ispisi(sortiraj(dodaj(baza2,”amer”)), cout);

Ispis na ekran:

amer hmina hmina puharem puharem vamir vamir