**PONAVLJANJE 3.b CHAR i STRING**

1. Zadatak

DA ZADATAK SA INSERT MODIFICIRATE TAKO DA KORISNIK SAM ODREDI OD KOJE POZICJE ŽELI DA SE UMETNE TEKST.

1. Zadatak

OVAJ GOTOVI ZADATAK KOJI PRONALAZI SVA POJAVLJIVANJA NEKE TRAŽENE RIJEČI PROUČITE SA UPUTAMA KOJE SU TU I VIDITE DA LI VAM JE JASNO



#include <iostream>

#include <string>

using namespace std;

int main(){

 string tekst;

 string trazi;

 int duljina;

 int trenutno=0;

 int polozaj;

 cout<<"Unesite tekst : "<<endl;

 getline (cin, tekst);

 cout<<endl;

 cout<<"Sto trazim?"<<endl;

 getline (cin, trazi);

 cout<<endl;

 duljina = tekst.size();

 while(trenutno < (duljina +1))

 {

 polozaj = tekst.find(trazi, trenutno);

 if (polozaj!=string::npos)

 {

 cout<<"Nalazi se na : "<<polozaj<<".mjestu"<<endl;

 trenutno=polozaj +1; // pamti svaku nadjenu rijec (brojac)

}

else

 {

 trenutno=duljina+1;

}

}

cout<<endl;

 system("pause");

 return 0; }

**OBJAŠNJENJE NAREDBI I ZADATKA:**

Prvo ide deklaracija objekata (varijabli), zatim varijabli koje nam
trebaju **tipa int** jer u njima nisu spremljena slova nego vrijednosti **int.**
**int duljina** nam treba zato da prije **while petlje** odredimo dokle će
program ići.
Znači u varijablu duljina spremamo **tekst.size()** našeg teksta
koji unesemo. To radimo pomoću naredbe **size()** (obratite opet pažnju na **točku!**)

Na taj nacin smo u **duljinu** spremili brojčanu vrijednost teksta.

Sad unutar **while** petlje upotrebljavamo varijablu **trenutno** i ona nam sluzi kao brojač.

Isto tako, ona nam određuje dok je trenutno manja od vrijednosti duljina (brojčana dužina teksta) naš program vrši ispitivanja unutar petlje while.
Unutar petlje while smo stavili if uvjet pomoću kojeg kao i prije u
programima ispitujemo da li je riječ nađena ili ne, ali prije te if naredbe moramo definirati varijablu **položaj** koju ispitujemo u **if naredbi.**
Opet upotrebljavamo **find** naredbu ali s malom izmjenom---umjesto broja ubacujemo int tip **trenutno** jer moramo pretražiti cijeli tekst, to uvjetujemo **while petljom.**
Sad unutar petlje if ispisujemo nađenu riječ kao i prije ali dodajemo
brojač koji će pamtiti svaki otkriveni položaj, a slijedeće pretraživanje kreće od tog položaja +1, i ispisuje ga na kraju programa.
U **else** petlji za svaku nepronađenu riječ opet **brojač** ali koji nam govori da ispitujemo slijedeće mjesto u rečenici dok ne dođemo do kraja.

**ZADACI ZA SAMOSTALAN RAD**

1. Program koji unosi jedan znak a zatim ga ispisuje 10 puta u jednom redu i zatim pet puta jedan ispod drugog.
2. Provjeriti je li upisani znak broj, slovo ili specijalan znak. Ponavljati ispitivanje sve dok korisnik ne unese slovo ***k*** za kraj ispitivanja.
3. Program koji rezervira prostor od 200 znakova za unos - broji koliko smo znakova unijeli kada smo unosili slova preko tipkovnice, a koliko nam je mjesta ostalo. Računaju se i prazna mjesta.
4. Program koji provjerava koliko je dug tekst koji smo unijeli.
5. Program koji provjerava koliko je niz dug (maksimalno 20 znakova) i ispisuje niz od sredine niza do kraja.
6. Napisati program koji u našem tekstu kojeg unesemo pronalazi točno određenu riječ i govori nam na kojoj se poziciji nalazi.
7. Program koji pronalazi sva pojavljivanja naše upisane riječi, a zatim umjesto nje upise riječ BRISANO.
8. Napišite program koji sa tipkovnice učitava niz te ispisuje koliko niz ima samoglasnika.
9. Napisati zadatak koji provjerava jesu li dva unesena teksta (riječi) jednake duljine i ispisuje odgovarajuću poruku ukoliko jesu, tj. nisu.
10. Napisati program kojim će korisnik unijeti jednu rečenicu, a računalo će zatim na zaslonu prikazati tu istu rečenicu ali će između prve i druge riječi ispisati „PISEMO“.

Npr: Unesi tekst: Danas ispit.

 *Rezultat:* Danas pisemo ispit.

1. Napisati program koji će pretražiti string kojega unosi korisnik i reći na kojim se sve pozicijama nalazi riječ FANTA ukoliko je unesena. Ukoliko riječi nema, tražit će ponovno upis ispisati će se odgovarajuća poruka da riječ nije pronađena.